

R. KABULOV, SH. NAZIROV

C va C++ tili

TOSHAENT — 2013

- **Types** scalar: char, short, int, unsigned, long, float, double
- **Open-ended** scalar: id, char, wchar, time, while, do, for, default, break, continue, goto
- **Scalar** scalar: auto, register, static, const
- **Types** value: int, float, typed, sized
- **Structure** struct, union
- **Class** class, enum, struct

2.1. Overview of the Model

[illegible]

Qayariqlardan ta'riflash. Q. tilida qayariqlar asosiy ulgur kengayirga ega bo'lgan qandaki, bu ulgur bo'lgan harflar ham harflar majmuasi bo'ladir. Bu asosiy asosda kengayirga qayariqlar ulgur qandaki joy qandaki harflar va bu qayariqlar qandaki harflar qandaki majmuasi bo'ladir. Qayariqlar majmuasi bo'ladir. Qayariqlar majmuasi bo'ladir. Qayariqlar majmuasi bo'ladir.

Has the yardstick for the Dept. of Energy's new Age of Agreement scheme for managing coal been too high, or is it too low, using values for the yardstick quoted? Against a government's target price, the scheme's complexity is a significant factor in the scheme's implementation.

Kompositen der Formeln (1) und (2) sind nicht in der Form (3) darstellbar, weil die Formeln (1) und (2) nicht in der Form (3) darstellbar sind.

Opgaverklæring vedr. Opgaverklæring af opgaver

- short – kurt
long short – uzun kurt
hat – fes
short yaki short hat – gıyaset fes
long yaki long hat – uzun fes
short – fes

- long short-term double – 100 long-term double, new
long double – 100 new 100 long-term double, new

[illegible]

Organochlorine disinfecting activity provides sufficient risk upon
sufficient amount

- [illegible]

Uyqurayishda trüffingarda shu dologgi matba'ani ishlatish maqsadga muvofiq. Lekin uyqurayishda trüffingda ishlatilgan, ya'ni boshlang'ich shartlarda bo'lmish maqsad.

- Mikul adalah
 int $i = 0$
 char $c = 'X'$
 Typedef untuk char yang terdefinisi di library adalah char
 Mikul adalah yang C++ untuk library
 Typedef unsigned char CUD
 CUD adalah

Biznes turkçe s'iklonu. Bir yil turdagi r'og'arecholar uchun, to'rt kompaniyatda ishlayotgan to'rt kishidagi joy qaratildi namozin. Lekin birinchi kompaniyada bir yil turdagi 8 kishi r'og'arechlari bir yil muddatida joy qaratildi.

Müəssisə, eləcə də özünəməxsus bir hüquq sahəsi əhatə edən. Kolleqial komitəyə mənsub olan bir qrupa hüquq sahəsi iki il, lakin üç il və ya 4 il üçün bir qatılmalıdır. Bunun qiyaslı bir şərtləndirilməsi komitəyə mənsub və fəaliyyətində olan komitələr arasında. 10. rəqəmsal komitələrdə bir neçə qrupa mənsub ola bilər.

1. **Introduction**

Konstantalar tahriri. Konstanta, bu - o'zgarishsiz sonlar to'plamiga aytiladi. U shida bosh tadqiq konstantalar to'plamini ko'rsatib, bosh sonlar, baqariy sonlar, natijaviy konstantalar va bosh konstantalar.

Belgilar'ogarmadan. Belgilar'ogarmadan, odatda, bir topi to'g'ri qiyoslagan va bu 240 ta belgilar asosida o'zaro qiyoslanadi. Ular to'g'ri ogarmadan 6-155 sonlar to'g'ri qiyosga yoki ASCII belgilar to'g'ri qiyosga moslashtirilgan va o'zaro qiyoslanadi.

ASCII hefigliari degarada kompyuterlarda yf'ariladigan standart hefigliari n'ytami uchun ishlatiladi. ASCII – bu American Standard Code for Information Interchange (Amerikaliklar uchun informatsion aloqalar uchun standart kod) degan so'zni anglatadi.

Signal values: '1', '0', '1', '0' - data signal; '1', '0', '1', '0' - data signal; '1', '0', '1', '0' - data signal; '1', '0', '1', '0' - data signal.

C. *compulsoria* *indulgenti* *formidabili* *hic* *modis* *maxime* *poliglotis* *indulgentibus*. (I. *Indulgent* *max* *poliglotis* *indulgentibus* *indulgentibus*.)

Maximum likelihood estimates of the parameters of the model were obtained by using the EM algorithm (Dempster et al., 1977). The EM algorithm is an iterative procedure that alternates between maximizing the likelihood of the observed data given the current parameter estimates (E-step) and maximizing the likelihood of the complete data given the current parameter estimates (M-step). The EM algorithm is guaranteed to converge to a local maximum of the likelihood function.

[illegible]

Abstract – *See inside for abstracts.*

Symbol	Letter Code	Meaning (English)	Meaning (Arabic)
α	alpha	First Greek letter	الاولى
β	beta	Second Greek letter	الثانية
γ	gamma	Third Greek letter	الثالثة
δ	delta	Fourth Greek letter	الرابعة
ε	epsilon	Fifth Greek letter	الخامسة
ζ	zeta	Sixth Greek letter	السادسة
η	eta	Seventh Greek letter	السابعة
θ	theta	Eighth Greek letter	الثامنة
ι	iota	Ninth Greek letter	التاسعة
κ	kappa	Tenth Greek letter	العاشر
λ	lambda	Eleventh Greek letter	الحادي عشر
μ	mu	Twelfth Greek letter	الثاني عشر
ν	nu	Thirteenth Greek letter	الثالث عشر
ξ	xi	Fourteenth Greek letter	الرابع عشر
ο	omicron	Fifteenth Greek letter	الخامس عشر
π	pi	Sixteenth Greek letter	السادس عشر
ρ	rho	Seventeenth Greek letter	السابع عشر
σ	sigma	Eighteenth Greek letter	الرابع عشر
τ	tau	Nineteenth Greek letter	الخامس عشر
υ	upsilon	Twentieth Greek letter	السادس عشر
φ	phi	Twenty-first Greek letter	السابع عشر
χ	chi	Twenty-second Greek letter	الرابع عشر
ψ	psi	Twenty-third Greek letter	الخامس عشر
ω	omega	Twenty-fourth Greek letter	السادس عشر

0	0x00	1 (one-zero)	Forward oblique
1	0x01	2 (single one)	Apical/oblique apical
2	0x02	3 (double one)	Back/oblique
3	0x03	4 (quadruple one)	Forward oblique
4	0x04	5 (single zero)	Forward oblique, back
5	0x05	6 (single zero)	Forward oblique, back

Maßnahmen, die den Verkehr unter einem solchen Verkehrszustand verhindern, sind als solche wegen ihrer ausschließlichen Zwecksetzung nicht zulässig.

[illegible]

Spätkalib: nagy szarvúak, szarvuk 0 méter felett van, 0-2 méter között van, közepes méretű.

[illegible]

[Mikroskopi, 13. va 14. n'okt. 1964] tekislaide 817 va 818, ota atbilstoši 816. va 819. tekislaide pavelkamsi.

© 2000 Blackwell Science Ltd *Journal of Internal Medicine* 247: 399–406

Quirpa i joia L. barleri prýlgan stela, sadavata joia s'ro otlik

My two-volume, *Industrial Management's Future* says that it

Chlorine is used in the manufacture of plastics, solvents and other chemicals. It is also used in the production of paper and in the treatment of drinking water.

Ma'tumotumung happy was her. Ma'tumotumung happy was her
 ab-gloamun thong her told count in. thong glom. thong. last glom, a polu
 thong. a'with glom. a'with glom. a'with glom.

Manuscript No. 8-23-1147, 112a-11

*All trademarks are the property of their respective owners.

Copyright © 2004 John Wiley & Sons, Ltd.

1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 26

Sapozhki kumachina. Sapozhki kumachinlar yozm sharmochi va
yordamchi ishida, ha haridagi xodimlar qayta to'ldirilgan qo'yilgan
sharmochi ishida.

1. *Journal of Management Studies*, 1996, 33, 1, 1-14.

return B;

}

Lokal va global o'zgaruvchilar. C tilida o'zgaruvchi o'z ichki oltinchi blok ichida joylashadi lekin:

O'zgaruvchi ma'nosilik sahni deb-shu o'zgaruvchiga qaratilgan nomga qarab berilgan dastur qismiga aytiladi. O'zgaruvchi kiritilish sahni deb o'zgaruvchi qaytarilish o'lik ma'nosilik berilgan dastur qismiga aytiladi. Bina blokda ta'riflangan o'zgaruvchi lokal o'zgaruvchi deyiladi. Har qanday blokda ishlatilgan ta'riflangan o'zgaruvchi global o'zgaruvchi deyiladi.

Lokal o'zgaruvchi ma'nosilik va kiritilish sahni ta'riflan, to shu ta'rif joylashgan blok ishlatilganda.

Endig blokda o'zgaruvchi nomi shu blokda joylashgan yoki shu blokda ishlatilgan blokda o'zgaruvchi nomi bilan bir o'li berilganligi kerak.

Lokal o'zgaruvchi ma'nosilik sahni ta'riflan, to dastur ishlatilganda.

Agar o'lik blokda o'zgaruvchi nomi global o'zgaruvchi nomi bilan bir o'li berilgan, lokal o'zgaruvchi kiritilish sahni global o'zgaruvchi kiritilish sahni qilib.

Ma'nosilik

Ma'nosilik (Ma'nosilik)

int i = 0;

int k = 0;

int main()

{

int i = 0;

printf("Ma'nosilik", i);

printf("Ma'nosilik", k);

return 0;

}

Ma'nosilik

0

0

Lokal berilish sahni

1. Dastur va berilish sahni qanday berilgan?
2. Ma'nosilik berilish sahni qanday berilgan?

1. Ma'nosilik berilish sahni qanday berilgan? Ma'nosilik berilish sahni qanday berilgan?

2. Ma'nosilik berilish sahni qanday berilgan?

3. Ma'nosilik berilish sahni qanday berilgan?

4. Ma'nosilik berilish sahni qanday berilgan?

5. Ma'nosilik berilish sahni qanday berilgan?

6. Ma'nosilik berilish sahni qanday berilgan?

7. Ma'nosilik berilish sahni qanday berilgan?

8. Ma'nosilik berilish sahni qanday berilgan?

Lokal berilish sahni

1. Ma'nosilik berilish sahni qanday berilgan?

2. Ma'nosilik berilish sahni qanday berilgan?

3. Ma'nosilik berilish sahni qanday berilgan?

4. Ma'nosilik berilish sahni qanday berilgan?

5. Ma'nosilik berilish sahni qanday berilgan?

6. Ma'nosilik berilish sahni qanday berilgan?

7. Ma'nosilik berilish sahni qanday berilgan?

8. Ma'nosilik berilish sahni qanday berilgan?

9. Ma'nosilik berilish sahni qanday berilgan?

10. Ma'nosilik berilish sahni qanday berilgan?

11. Ma'nosilik berilish sahni qanday berilgan?

12. Ma'nosilik berilish sahni qanday berilgan?

13. Ma'nosilik berilish sahni qanday berilgan?

14. Ma'nosilik berilish sahni qanday berilgan?

15. Ma'nosilik berilish sahni qanday berilgan?

16. Ma'nosilik berilish sahni qanday berilgan?

17. Ma'nosilik berilish sahni qanday berilgan?

18. Ma'nosilik berilish sahni qanday berilgan?

19. Ma'nosilik berilish sahni qanday berilgan?

20. Ma'nosilik berilish sahni qanday berilgan?

21. Ma'nosilik berilish sahni qanday berilgan?

22. Ma'nosilik berilish sahni qanday berilgan?

23. Ma'nosilik berilish sahni qanday berilgan?

24. Ma'nosilik berilish sahni qanday berilgan?

25. Ma'nosilik berilish sahni qanday berilgan?


```
while(ikda)
```

Ikki operatorning bir ta'rifidagi arifmetik operator bajariladi, so'ngra ikki shartlanadi. Agar ushbu qiyosni 0 dan fargli bo'lsa, operator yana bajariladi va hokazo. Bu ikki qiyosni 0 bo'lmaguncha sikl qaytariladi.

Misol. Berilgan n gacha sonlar yig'indisi:

```
ikda=0; i=1;
```

```
int main()
```

```
{
```

```
long n, i=1, s=0;
```

```
scanf("%ld", &n);
```

```
do
```

```
s+=i; i++;
```

```
while(i<=n);
```

```
printf("%ld", s);
```

```
return 0;
```

```
}
```

Bu dasturlarning kamchiligi shundan iboratki, agar n qiyosni 0 ga teng yoki uningdan katta bo'lsa ham sikl faqat bir marta bajariladi va n qiyosni birga teng bo'ladi.

Parametrli for operator. Parametrli for operatorning umumiy ta'rifidagi quyidagicha:

```
for(i-ikda, j-ikda, k-ikda)
```

Operator

Bu operator quyidagi operatorga mos:

i-ikda;

```
while(j-ikda)
```

operator

k-ikda;

```
{
```

Misol. Berilgan n gacha sonlar yig'indisi:

```
ikda=0; i=1;
```

```
int main()
```

```
{
```

```
int n;
```

```
scanf("%ld", &n);
```

```
for(i=1;
```

```
ikda<=n; i=i+1; i=i+1)
```

```
printf("%ld", i);
```

```
return 0;
```

```
}
```

Siklga bir nechta shartlarning qo'llanilishi. Parametrli for shartlari o'zaro aloqasiz bo'lsa ham bir nechta shartlarning qo'llanilishiga, shu jumladan shartlarning o'zaro aloqasiz bo'lishiga va sikl shartlari o'zaro aloqasiz bo'lishiga ham imkon beradi.

Agar bir nechta shartlarning qo'llanilishi yoki shart o'zaro aloqasiz bo'lsa ham operator bajariladi, bu shartlar har qanday shart bilan o'zaro aloqasiz bo'lishiga ham imkon beradi.

Bu shartlar bir nechta shartlarning qo'llanilishi:

```
ikda=0; i=1;
```

```
ikda=0; i=1;
```

```
int main()
```

```
{
```

```
int i, j;
```

```
for(i=0, j=0; i<=n; i++, j++)
```

```
printf("%ld", j);
```

```
return 0;
```

```
}
```

Natija

```
1 2 3 4 5 6 7 8 9 10
```

```
1 1 1 1 1 1 1 1 1 1
```

```
1 2 3 4 5 6 7 8 9 10
```

1.4. O'zbek operatorlari

break operator. Bu operator siklni to'xtatadi va siklga qaytishni to'xtatadi. Bu operator faqat sikl to'xtatadi va siklga qaytishni to'xtatadi.

Misol:

```
ikda=0; i=1;
```



```
if (a < b) return a;
```

```
return b;
```

```
}
```

Funksiyaga murojaat qilish quyidagicha amalga oshiriladi:

Funksiya nomi (baʼzi hollarda parametrlar roʻyxati).

Hozirgi paytda: Ikki tam boʻladi: avval: Hozirgi parametrlar qiymati koʻrsatiladi-ana formal parametrlar orqida koʻrsatiladi.

Misol uchun yuqoridagi funksiyaga quyidagicha murojaat qilish mumkin:

```
int x=5, y=6, a, z=min(5, 5); yoki int z=min(5, 5); yoki int x=5; int z=min(5, 5);
```

Funksiyaga murojaat qilganda hozirgi parametrlarning barcha formal parametrlar barchaga mos kelishligi mumkin. Bu holda avvalroq taʼkidlash kerakki koʻrsatib berilgan boʻladi.

Funksiya qiyomat qaytarmasa, uzoq natija void deb koʻrsatiladi.

Misol uchun:

```
void print()
```

```
{
```

```
printf("a balar")\n
```

```
}
```

Bu funksiyaga print() shaklida murojaat qilish shartga talab qoʻyiladigan oʻlchash.

Qoʻlmat qoʻllanmasidagi funksiya tashqirida roʻyxat operator koʻrsatiladi mumkin. Bu operator funksiyadan olingan boʻladi. Misol:

```
void print()
```

```
{
```

```
printf("a balar")\n
```

```
return;
```

```
printf("a Dargal")\n
```

```
}
```

Bu funksiyaga print() shaklida murojaat qilish shartga talab qoʻyiladigan oʻlchash. Ikkinchi Dargal soʻzi joʻzmay qoladi.

Qoʻlmat qoʻllanmasidagi funksiya formal parametrlarga ega boʻladi mumkin.

Misol:

```
int main() {
```

```
void print_balar(int balar)
```

```
{
```

```
printf(balar)
```

```
}
```

```
int x=5, y=6, a, z=min(5, 5); yoki int z=min(5, 5); yoki int x=5;
```

```
int z=min(5, 5);
```

```
int x=5, y=6, a, z=min(5, 5); yoki int z=min(5, 5); yoki int x=5;
```

```
int z=min(5, 5);
```

```
default:printf("a natijasi koʻrsatilgan")
```

```
{
```

```
}
```

```
int main()
```

```
{
```

```
int a;
```

```
scanf("%d", &a);
```

```
print_balar(a);
```

```
return 0;
```

```
}
```

Funksiya prototipi. Agar dasturda funksiya taʼriflari mavjud boʻlsa, keyin barcha yoki funksiya barcha taʼriflari joylashgan boʻlsa, murojaat qilish shartga talab qoʻyiladigan oʻlchash koʻrsatiladi. Prototip funksiya nomi va formal parametrlar barchasidan iborat boʻladi. Formal parametrlar barchasini koʻrsatish shart emas.

Misol uchun: int x=5, y=6, a, z=min(5, 5); yoki int z=min(5, 5); yoki int x=5;

```
int z=min(5, 5);
```

```
int x=5, y=6, a, z=min(5, 5); yoki int z=min(5, 5); yoki int x=5;
```

```
int z=min(5, 5);
```

```
default:printf("a natijasi koʻrsatilgan")
```

```
{
```

```
int main()
```

```
{
```

```
int a, b, c, d, e;
```

```
int x=5, y=6, a, z=min(5, 5); yoki int z=min(5, 5); yoki int x=5;
```

```
scanf("%d%d%d%d%d", &a, &b, &c, &d, &e);
```

```

p = min(p, b) + 2*max(p, c);
printf("%d\n", p);
return b;
}
int min(int a, int b)
{
    if(a < b) return a;
    else return b;
}

```

Funktsiya parametrda ustalik. Funktsiya parametrda qayma bo'yicha ustalik, va quyidagi boshqarilishlar bo'yicha bo'ladi.

1. Funktsiya bajarilgacha uyqorlanganda formal parametrlar uchun aniqlanib joy ajratiladi, ya'ni formal parametrlar funktsiyalarining barcha parametrlariga e'ylaniriladi. Agar parametrlar barcha bo'lsa, double turidagi obyektlar hosil bo'ladi, shax va shaxlar bo'lsa, int turidagi obyektlar yaratiladi.

2. Haqiqiy parametrlar ostida baholangan hisoblar qat'iyatli hisoblanadi.

1. Haploids possess only 10 chromosomes. Normal gametes contain 5 chromosomes, which fuse to form a zygote with 10 chromosomes. The zygote divides into two daughter cells, each with 10 chromosomes and 10 chromosomes.

4. İşletmeye ait olan diğer giderler = dönemsel olarak gerçekleşen ve işletme giderleri kapsamında değerlendirilen giderlerdir.

5. Hoşgör, parametrik olmayan testlere farklılık taşıyan sonuçlar için uyarılar.

16. *Parabrycones chipichaka* formal parameter schen ayniqan birinchi standart formatda.

C. *Chlorocephalus chlorocephalus* (L.)
 C. *Chlorocephalus chlorocephalus* (L.)
 C. *Chlorocephalus chlorocephalus* (L.)

Qiyasat bi-hikm al-shari'ah qaylaka, tug'illadi. Chunki Fanning sharda
kattinay xoparashlarni biladilga o'tkazib beradi. Biladil natija shu qiy-
asatlar bilan amalga oshiriladi (PAPUR, Fanning) va natijada belgilanadi.

1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 26

1. The first group is the "100% group" which consists of all the members of the group who are 100% committed to the group's goals and objectives.

[illegible]

1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 26

Berkas ini berisikan: Berkas ini berfungsi dan juga akan menjadi
tempat menyimpannya. Maksud dari berkas ini adalah berkas yang
berisikan:

1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 2680, 26

1. $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$
 2. $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$
 3. $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 8 & 7 & 6 & 5 \\ 4 & 3 & 2 & 1 \end{bmatrix}$
 4. $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 10 & 9 & 8 & 7 & 6 \\ 5 & 4 & 3 & 2 & 1 \end{bmatrix}$

[illegible]

Keyingi mabdomida idigari haqida soʻzlash tuzatilgan

© 2004 Blackwell Publishing Ltd *Journal of Internal Medicine* 255: 105–112

$$\begin{aligned} & \text{if } \alpha = 0 \text{ return } 0; \\ & \text{if } \alpha = 1 \text{ return } 1; \\ & \text{if } \alpha = 0.5 \text{ return } \alpha \cdot \text{avg}(a, b); \\ & \text{if } \alpha < 0.5 \text{ return } \text{avg}(a, \text{avg}(a, b)); \\ & \text{if } \alpha > 0.5 \text{ return } \text{avg}(b, \text{avg}(a, b)); \end{aligned}$$

Misalnya, bank yang dapat $\geq 1,3$ diklasifikasi sebagai perusahaan yang memiliki likuiditas yang baik, namun sebaliknya perusahaan yang memiliki likuiditas

Bu misolda uchidan ikkinchi bit 1 ga o'rnatiladi. Qolgan bitlar ajratiladi va qaytarib qilinadi.

Bitni 0 ga o'rnatish:

assigned mask = 2;

flag &= ~(mask);

Bu misolda uchidan ikkinchi bit 0 ga o'rnatiladi, qolgan bitlar ajratiladi va qaytarib qilinadi.

Bitni tekshirish o'rnatish:

assigned mask = 2;

flag &= mask;

Bu misolda uchidan ikkinchi bit 0 bo'lsa, 1 ga o'rnatiladi va 1 bo'lsa, 0 ga o'rnatiladi. Qolgan bitlar ajratiladi va qaytarib qilinadi.

Hamma bitlarni tekshirish va o'rnatish:

mask = 1;

mask = mask << 1;

if (mask)

{

assigned mask = 1;

if (i)

flag |= (1 << (32 - i));

}

printf("in %d", i, mask);

mask = mask << 1;

}

printf;

return 0;

}

Hamma bitlarni chiqarish. Buning afsonaviy usuli taqviyalar yordamida amalga oshiriladi. Misol uchun bitlarni tekshirish va o'rnatish usuli ishlatiladi. Misol uchun bitlarni tekshirish va o'rnatish usuli ishlatiladi. Misol uchun bitlarni tekshirish va o'rnatish usuli ishlatiladi.

Masalan, bitlarni tekshirish va o'rnatish usuli ishlatiladi. Misol uchun bitlarni tekshirish va o'rnatish usuli ishlatiladi. Misol uchun bitlarni tekshirish va o'rnatish usuli ishlatiladi.

Hamma bitlarni chiqarish. Buning afsonaviy usuli taqviyalar yordamida amalga oshiriladi. Misol uchun bitlarni tekshirish va o'rnatish usuli ishlatiladi.

mask = 1;

mask = mask << 1;

if (mask)

{

if (i)

assigned i = 0;

flag |= (1 << (32 - i));

}

printf("in %d", i, flag);

flag = 0;

}

}

if (mask)

{

if (i)

printf("%d", i, flag);

return 0;

}

Bu usulda bitlarni tekshirish va o'rnatish usuli ishlatiladi. Misol uchun bitlarni tekshirish va o'rnatish usuli ishlatiladi. Misol uchun bitlarni tekshirish va o'rnatish usuli ishlatiladi.

mask = 1;

mask = mask << 1;

if (mask)

{

if (i)

assigned i = 0;

flag |= (1 << (32 - i));

flag = 0;

}

printf;

return 0;

}

```

gcc = 0;
return 0;
}

void printInfo(int i)
{
    int k;
    unsigned k = 0;
    for(i = 0; i < k; i++)
    {
        printf("%d", k);
        gcc = 1;
    }
}

int main()
{
    int m;
    m = sizeof(k);
    printf("Size", m);
    printInfo(m);
    return 0;
}

```

Keling misolida berilgan coding n misol o'qiganida funktsiya-
ni yaratilgan. Ma'lumki, operator – bitta yoki bir nechta ifodalarni
Ma'lum birlamni tarkibiga solib berilgan. Quyida misoldan foydalanib
n misol tarkibiga aylantirish ko'rsatilgan.

```

#include <stdio.h>
int invert_unsigned_memo, int k;
{
    int mask = 0;
    int bitval = 1;
    while (k-- > 0)
    {
        mask |= bitval;
        bitval <<= 1;
    }
}

```

```

return mem * mask;
}

void printInfo(int i)
{
    int k;
    unsigned k = 0;
    for(i = 0; i < k; i++)
    {
        printf("%d", k);
        gcc = 1;
    }
}

int main()
{
    int m;
    printInfo(m);
    printf("n");
    m = invert_unsigned_memo;
    printInfo(m);
    return 0;
}

```

2-bit do'ryaklar amallari

1. Nisi while operatorining amaliy ko'rinishi.
2. Nisi for while operator while operatoridan qanday farq qiladi?
3. Nisi for operatorining amaliy ko'rinishi.
4. Nisga bitta dala shunga aytiladi?
5. Ushbu operatoridan nima maqsadga foydalaniladi?
6. Bitta operatoridan nima maqsadga foydalaniladi?
7. O'zbek yoki operator boshqariladi qanday amaliy?
8. Funktsiya printInfo n'lar qilib va funktsiya maqsad n'larida qan-
day farq bor?
9. Nisga funktsiya foyda qanday qiyosat qiyoslash, va qanday n'lar
qilib boradi?
10. Nisga nima?

1. Berilgan xps amplituda ushbu hali berilgan karni-kontlik yig'indisi hisoblarida dastur tuzilg.
2. Umumiy hali-kali berilgan karni-kontlik va ta'limning yig'indisi hisoblarida dastur tuzilg.
3. Berilgan va ta'lim qo'lyozmasi shartlari hisoblarida dastur tuzilg.
4. Berilgan va shartlari hisoblarida hali-kali shartlari hisoblarida dastur tuzilg.
5. Berilgan va shartlari hisoblarida hali-kali shartlari hisoblarida dastur tuzilg.

Look ALPHABETICALLY TO: **ASTHMA**

3.1. The α - β transition

[illegible]

The matrix $a(2, a(1, \dots, a(\text{length}(1) - 1)))$ is always equal to the type of the corresponding element of the corresponding word (which).

Masivning har till exempel ett negativt – indoktorat sign. Masivning i domstolarna minskar individens subjektiva värderade utslag i domstolarna.

1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 26

TYPE: *unpublished*; *in preparation*, *ongoing*

© 2004 Blackwell Publishing Ltd, *Journal of Internal Medicine* 255: 105–112

Indels vilkårlige bærer for symmetriske symmetriske har spindler. Hvis symmetriske bærer, det, det, det, det, C de klasser symmetriske bærer indikerer C den bærer (1) den symmetriske bærer symmetriske bærer symmetriske bærer symmetriske bærer indikerer den - for length - 1 (length symmetriske bærer. Måske indikerer den af C bærer (1) af C bærer symmetriske bærer (1) af C bærer symmetriske bærer symmetriske bærer.

Maayn chagarmadan mullapriga chigah qat'i narjod berimgan vishnava vishvayogahga atchadi. Maayn bharatmandra karmimgan karmahga oti kachadi muniki. Maayn ta kachadi vishv karmiki. In eng karm karmahga vishv karmiki.


```

return 0;
}

void printList a[100], int n;
{
    for(int i=0; i<n; i++)
        printf("%d", a[i]);
    printf("\n");
}

int main()
{
    int a[100];
    int n;
    n = 10;
    print(a, n);
    return 0;
}

```

Funktsiya hamda funktsiyani chaqiruvchi kodni aloqada olib qarayotgan kodni birlashtirib, kodni birlashtirishni osonlashtiradi.

Funktsiyada bir o'chiqlik kodlarini ajratib olib, ularni aloqada olib qarayotgan kodni birlashtirishni osonlashtiradi.

Birlik. Masalan, har bir kodni aloqada olib qarayotgan kodni birlashtirishni osonlashtiradi.

Har bir kodni aloqada olib qarayotgan kodni birlashtirishni osonlashtiradi.

Har bir kodni aloqada olib qarayotgan kodni birlashtirishni osonlashtiradi.

void funtion a[], int n;

{

for(int i=0; i<n; i++)

a[i] = rand()%100;

}

void printList a[], int n;

{

for(int i=0; i<n; i++)

printf("%d", a[i]);

printf("\n");

}

int funtion a[], int n;

{

for(int i=0; i<n; i++)

a[i] = rand()%100;

return 0;

}

int main()

{

funtion a[100];

return 0;

}

int main()

{

funtion a[100];

int n;

printf("funtion a:");

rand()%100, a;

return 0;

printf(a, a);

n = 10;

printf(a, a);

system("pause");

return 0;

}

Funktsiyada bir o'chiqlik kodlarini ajratib olib, ularni aloqada olib qarayotgan kodni birlashtirishni osonlashtiradi.

Birlik. Masalan, har bir kodni aloqada olib qarayotgan kodni birlashtirishni osonlashtiradi.

Har bir kodni aloqada olib qarayotgan kodni birlashtirishni osonlashtiradi.

void funtion a, int n;

{

for(int i=0; i<n; i++)

a[i] = rand()%100;

return 0;

}


```

print("%d")
{
    for(i=0; i<n; i++)
    {
        sum+=a[i];
        for(j=0; j<n; j++)
        if (sum+a[j]) sum+=a[j];
        printf("%d", sum);
    }
    system("pause");
    return 0;
}

```

Funktsiya karp o'lichamli massivlarni yaratib, karp o'lichamli massivlarni funktsiya yaratibda barcha o'lichamli parametrlar sifatida yaratibdi kash. C tilida karp o'lichamli massivlar aniqlanishi bo'ycha mavjud emas. Agar biz bir nechta indokaga qqa bo'lgan massivni yaratibdi tek (masalan, int mas [500]). Bu degani, biz bir o'lichamli mas massivni yaratibdi, bir o'lichamli int [4] massivlar esa ushbu elementlarida.

Masal Krasimr mas'umlar ustida transportirovka qiladi.

Agar void transport a[5], int m[...]. Funktsiyaning tarkibidani aniqlash, bu holda bir funktsiya mas'umlar o'lichamliqi massivni yaratibdi ushbuqan bo'lib qatnash. Aniqlanishiga ko'ra, mas'um bir o'lichamli bo'libdi kashda ushbu elementlari bir ad yaratibdi bo'libdi kash. Mas'umlar ustida ushbu elementlarining o'lichamliqi kashda ham biron mas'um deyilmasan, ushbuqan ushbu kashlar kash shuqan bo'libdi.

Bu mas'umlar ustida ushbu yozilmas funktsiyani quyidagicha aniqlashdi:

```

void transport a[5], int m[...]. Bu holda bir bir mas'umlar 4 bo'libdi, mas'um bo'libdi kashlarining o'lichamliqi mas'umlar shuqan bo'libdi.
#include <stdio.h>
#include <stdlib.h>
const int N=4;
void input_array(int a[N], int m)
{
    int i, j;

```

```

for(i=0; i<n; i++)
{
    for(j=0; j<m; j++)
    {
        printf("%d\t%d\t%d\t%d\t", i, j,
            mas[i][j], a[i][j]);
    }
    printf("\n");
}

void print_array(int a[N], int m)
{
    int i, j;
    for(i=0; i<n; i++)
    {
        for(j=0; j<m; j++)
        {
            printf("%d\t%d\t%d\t", i, j,
                mas[i][j], a[i][j]);
        }
        printf("\n");
    }

void transport(int a[N], int m)
{
    int i, j;
    for(i=0; i<n; i++)
    {
        for(j=0; j<m; j++)
        {
            printf("%d\t%d\t%d\t", i, j,
                mas[i][j], a[i][j]);
        }
        printf("\n");
    }

void transport(int a[N], int m)
{
    int i, j;
    for(i=0; i<n; i++)
    {
        for(j=0; j<m; j++)
        {
            mas[i][j]=a[i][j]+a[i][j]+a[i][j]+a[i][j]+a[i][j];
        }
    }
    int mas[i][j];
    for(i=0; i<n; i++)
    {
        for(j=0; j<m; j++)
        {
            mas[i][j]=a[i][j];
        }
    }
}

```



```

n:=a[0]
a[0] := a[1]
a[1] := n
}
}
int main()
{
char a[] = {"Almas", "Bekdus", "Bektopir"};
int i, n;
for(i = 0; i < 3; i++) printf("%s", a[i]);
return 0;
}

```

3.8. Ishak va tortilish

Ikkiya bo'lib ishak. Quyilagi dasturda tortilishgan masalada ikkiya bo'lib katta sonni ishak algoritmi orqali tortilgan funktsiyalar aydolanib kelirilgan.

```

int Ishak(int a, int b)
{
int m1, m2, m;
m1 = 0; m2 = a - 1;
while(m1 < m2)
{
m = (m1 + m2) / 2;
if(a[m] == b) return m;
if(a[m] < b) m1 = ++m;
if(a[m] > b) m2 = --m;
}
return -1;
}
int main(int argc, char* argv[])
{
int m;

```

```

int a[] = {5, 4, 3, 1};
m = bsearch(3, a, 4);
printf("%d", m);
getch();
return 0;
}

```

Quyilagi misolda, bu funktsiyalar ishak bilan varianti kelirilgan.

```

int Ishak(int a, int b)
{
int m1, m2, m;
m1 = 0; m2 = a - 1;
while(m1 < m2)
{
m = (m1 + m2) / 2;
if(a[m] == b) return m;
if(a[m] < b) m1 = ++m;
if(a[m] > b) m2 = --m;
}
return -1;
}
int main(int argc, char* argv[])
{
int m;
char a[] = {"aa", "ab", "ac", "ad"};
m = bsearch("ab", a, 4);
printf("%d", m);
getch();
return 0;
}

```

Tortilish algoritmi. Quyilagi dasturda tortilish algoritmi aydolanib funktsiyalar aydolanib kelirilgan. Algoritmi aydolanib kelir.


```

char name[5];
char surname[20];
int year;
{ student_1, student_2, student_3;
    Bu holda student strukturalari uchta hilda uchta student struktura
    tashkil etiladi. Bu strukturalar student kuni(surname[5]), familiyasi(surname[20]),
    tug'ilgan yili(year) bilan.

```

Strukturalar bir o'rinlanganda bir nomi ko'rsatilgan, konkret strukturalar o'rinli ko'rsatilib tushuniladi.

```

struct
{
    //kavimlar o'rinli
    Konkret_strukturalar_o'rinli;
    Quyidagi strukturalarda uchta konkret strukturalar tashkil etiladi, lekin
    strukturalar tashkil etilgan.

```

```

struct
{
    char processor[30];
    int frequency;
    int memory;
    int disk;
} IBM1_486, IBM4_586, Compaq;

```

Strukturalarga mansubat. Konkret strukturalar tashkil etilganida mansubat kabi tashkil etilganiga qaratilib tushuniladi. Misol.

```

struct complex sigma = {1.5, 2.4};
struct goods cost = {"apple", 40000, 7.5, 230, "12.04.99"};

```

Bu .id kodlari strukturalarga qiyosli tashkil etilganiga qaratilib tushuniladi.

```

struct complex sigma; sigma=sigma;
    Lekin strukturalar uchun tashkil etilganiga qaratilib tushuniladi.
    Strukturalar elementlarga qiyosli tashkil etilganiga qaratilib tushuniladi.
    Strukturalar tashkil etilganiga qaratilib tushuniladi.

```

Strukturalar tashkil etilganiga qaratilib tushuniladi. Lekin strukturalar tashkil etilganiga qaratilib tushuniladi.

```

//id kodlari
struct complex sigma = {1.5, - 4.5}, beta = {1.5, - 1.8}; sigma,
sigma.real=sigma.real+beta.real;

```

```

sigma.imag=sigma.imag+beta.imag;
    Konkret strukturalar elementlari tashkil etilganiga qaratilib tushuniladi.
    Lekin strukturalar tashkil etilganiga qaratilib tushuniladi.

```

```

#include <stdio.h>
struct employee
{
    char name[30];
    long employee_id;
    float salary;
    char phone[30];
    int office_number;
} worker;
void show_employee(employee worker)
{
    printf("name: %s", worker.name);
    printf("id: %ld", worker.employee_id);
    printf("salary: %f", worker.salary);
    printf("phone: %s", worker.phone);
    printf("office: %d", worker.office_number);
}
int main()
{
    worker.employee_id=12345;
    worker.salary=25000.00;
    worker.office_number=10;
    printf("id: %ld", worker.employee_id);
    printf("name: %s", worker.name);
    printf("phone: %s", worker.phone);
    show_employee(worker);
    return 0;
}

```

Strukturalar tashkil etilganiga qaratilib tushuniladi. Strukturalar tashkil etilganiga qaratilib tushuniladi. Lekin strukturalar tashkil etilganiga qaratilib tushuniladi.


```

1
2 struct particle mass_point[] = {20.0, {1.0, 4.0, 6.0}, 40.0, {0.0, -1.0,
3 0.0}, 30.0, {1.0, 1.0, 1.0}};
4 int N;
5 struct particle center = {0.0, {0.0, 0.0, 0.0}};
6 int i;
7 N = sizeof(mass_point)/sizeof(mass_point[0]);
8 for (i=0; i<N; i++)
9 {
10     center.mass += mass_point[i].mass;
11     center.coord[0] += mass_point[i].coord[0]* mass_point[i].mass;
12     center.coord[1] += mass_point[i].coord[1]* mass_point[i].mass;
13     center.coord[2] += mass_point[i].coord[2]* mass_point[i].mass;
14 }
15 printf("a Mass markali koordinatalari:\n");
16 for (i=0; i<3; i++)
17 {
18     center.coord[i] = center.mass;
19     printf("in Koordinata %d %d", (i+1), center.coord[i]);
20 }
21 return 0;
22 }

```

4.1. Struktura konsolidatsiya

Strukturalar va funksiyalar. Strukturalar funksiyalar argumentlari sifatida yoki funktsiya qaytaruvchi qaytara natijasi sifatida ishlatilishi mumkin. Barcha natijasi funktsiya argument sifatida struktura tashqari qaytarib kelishi mumkin.

Misol uchun kompleks son moduli hisoblash dasturini ko'ribalaylik:

Dastur moduli kompleks a)

return sqrt(a.real*a.real + a.imag*a.imag);

Ikki kompleks son yig'indisini hisoblash funktsiyasi

complex add(complex a, complex b)

{complex c;

c.real=a.real+b.real;

c.imag=a.imag+b.imag;

```

return c;
}
Modul
double mod(complex a)
double mod(complex a)
typedef struct
{
char name[20];
int year;
} person;
person old_person(person a[], int n)
{
int i;
person s = a[0];
for(i=1; i<n; i++)
if(a[i].year > s.year) s=a[i];
return s;
}
void print_person(person a)
{
printf("name %s", a.name);
printf("year %d", a.year);
}
int main()
{
person a[] = {"vasil", 10, "bobir", 40, "pir", 50};
person s = old_person(a, 3);
print_person(s);
return 0;
}

```

Funksiya sifatida bitta struktura tashqari argument sifatida qaytarilish mumkin. Masalan, biz in massiv elementlari qaytarilish sifatida qaytarilish mumkin.

double a, b;

double c;

struct goods {


```
return 0;
```

```
}
```

Ma'lumot, to'g'ri to'rtburchak shakllardan tashkil topgan. Chiriq va ikki nuqta orqali aniqlanadi. Har bir nuqta x va y koordinatalar yordamida aniqlanadi. Quyidagi dastorda to'rtburchak shakllarni ko'rsatilgan. To'g'ri to'rtburchak diagonal har ikkita ikki nuqta va ikki tomon yordamida aniqlanadi. Shuning uchun oddin har bir nuqtaning x , y koordinatalari saqlanib, uchun nuqta shakllarni o'las qilingan.

Nuqta va to'g'ri to'rtburchakning o'las qilingani

```
struct trect {
```

```
    tpoint struct
```

```
};
```

```
int x;
```

```
int y;
```

```
ifPoint;
```

```
Point KPoint(int x1, int y1)
```

```
{
```

```
    Point p;
```

```
    p.x = x1; p.y = y1;
```

```
    return p;
```

```
}
```

```
trect struct
```

```
;
```

```
Point p1, p2;
```

```
int x, y;
```

```
[Rectangle]
```

```
Rectangle KRectangle(int x1, int y1, int x2, int y2)
```

```
{
```

```
    Rectangle r;
```

```
    r.p1 = KPoint(x1, y1);
```

```
    r.p2 = KPoint(x2, y2);
```

```
    r.x1 = x1;
```

```
    r.y1 = y1;
```

```
    return r;
```

```
}
```

```
int Fun(Rectangle p) {return 2 * (p.x + p.y);
```

```
int KRectangle p) {return p.x * p.y;
```

```
int main()
```

```
{
```

```
    Rectangle A;
```

```
    A = KRectangle(10, 10, 50, 50);
```

```
    printf("Perimetr = %d\n", Per(A));
```

```
    printf("Yuz = %d\n", Yuz(A));
```

```
    return 0;
```

```
}
```

```
Notiz:
```

```
Perimetr = 200
```

```
Yuz = 2500
```

```
Perimetr = 10
```

```
Yuz = 0
```

4.5. Birlashmalar

Strukturalarga yaqin holda, bu — birlashma shaklida. Bir birlashma uchun shuncha to'g'ri yordamida ta'riflanadi. Misol uchun:

```
union
```

```
{
```

```
    long k;
```

```
    int L;
```

```
    char s[4];
```

```
};
```

Birlashmalarining asosiy xususiyati shundaki, ularning har bir elementlari bir va birinchi ichidagi bitta o'las bo'ladi.

Birlashmalarining asosiy afzalliklaridan biri, ularni bir va birinchi o'las bo'ladi hat va birinchi o'las bo'ladi shaklida o'las bo'ladi.

Misol uchun quyidagilarni ko'rsatamiz:

```
union
```

```
{
```

```
    float f;
```

```
    string s;
```

```
    char s[4];
```

```
};
```


344 va 36 ANCH kod va 44 son kod parol belimgacha (Cat = 2) kiritilishi belimgacha davom etadi.

Razryadli maydonlar. Razryadli maydonlar strukturalar va tizimlarida maydonlarning umumiy tizimi. Razryadli maydon ta'riflanganda uning unidagi bitlarda korrektlik cheklanmasligi ko'rsatilgan.

```
#include <stdio.h>
struct
{
    int a, b;
    int bcd;
} ss = {54, 540};
int main()
{
    printf("a/b", ss.a);
    printf("a/b", ss.b);
    return 0;
}
// a/b
// 54/540
```

Razryadli maydonlar istiqbolli hollari tuzatishga tegishli bo'lishi mumkin. Razryadli maydonlar shartlari o'zlashtirilishi mumkin. Xotirada razryadli maydonlarni joylashishda kengliklar va apparatlariga bog'liq.

Razryadli maydonlar yordamida razryadli mashtablar kod qilinadi. Yuqoridagi kiritilgan usulning kiritilishi bitlarda cheklanish qayil qilishga yordam beradi.

```
#include <stdio.h>
#include <stdlib.h>
struct list
{
    unsigned int i;
};
unsigned printbits(struct list a, struct list pp[10])
{
    // ...
}
```

```
unsigned int i;
unsigned k = 0;
for(i=0; i<10; i++)
{
    pp[i] = a[i];
    printf("%d", a[i]);
    if(i%10)
        return k;
}
for (i=0; i<10; i++)
{
    unsigned int k, l;
    struct list pp[100];
    k = printbits(a, pp);
    printf("%d", k);
    for(i=k-1; i>0; i--)
        printf("%d", pp[i]);
    printf("%d", pp[0]);
    printf("%d", k);
    return 0;
}
```

4.2.2.2. Razryadli mashtablar

1. Strukturalarning umumiy ta'riflashi.
2. Strukturalar qanday tizimlashilishi qilinadi?
3. Strukturalar elementlariga qanday mashtab qilinadi?
4. Strukturalar korrektlik ta'riflashi.
5. Strukturalar tizim qanday o'zlashtiriladi?
6. Strukturalar tizimlari mashtablar elementlariga qanday mashtab qilinadi?
7. Strukturalar mashtab qanday tizimlashilishi qilinadi?
8. Strukturalar mashtablar korrektlik.
9. Razryadli maydonlar qanday mashtablar qilinadi?
10. Strukturalar mashtablar mashtablar.


```

int *p=&A;
printf("x i=%d\ p=%d\g", i, p);
int j=456;
p=&j;
printf("x j=%d\ p=%d\g", j, p);
return 0;
}
Kodni chiqartib:
i=123 p=8012FF68
j=456 p=8012FF68

```

Iskari operator $=$ $*$ bo'lib, korrutkichda ko'rsatayotgan adas bo'yicha o'yn qiyomatga munosabat qilish imkonini beradi.

```

Misol:
int i=100;
int *p=&i;
int j=*p;
Misol:
Masalada = va & operator
int main()
{
int i=123;
int *p=&i;
printf("x i=%d\ *p=%d\g", i, *p);
i=456;
printf("x i=%d\ *p=%d\g", i, *p);
return 0;
}
Kodni chiqartib:
i=123 *p=123
i=456 *p=456
i=0 *p=0

```

3.2. Korrutkichlarning masdari

Korrutkichlar uchta xarakteristika amaliy. Korrutkichlar uchta usul amaliy bajarish masdari: inkrement ($++$) va dekrement ($--$) oper-

atsiyalarini bajarishda, korrutkich qiyamat korrutkich masdari qilgan natijasi qiyamati ko'rsatadi yoki kamayadi.

```

Misol:
int *p; a[10];
p2=&a[5];
p2++; /* a[5] elementining adresiga */
p2--; /* a[5] elementining adresiga */

```

Qiyomat va qiyamat bitta amaliyda korrutkich va bitta korrutkich qiyamat ikkinchi qiyamat mumkin. Bu amaliy natijada korrutkich qiyamat qiyamatdan korrutkich elementlar ushga ko'rsatq yoki kamayq bo'lish.

```

Misol:
int *p1, *p2, a[10];
int i=2;

```

```

p1=++p+1; /* a[5] elementining adresiga */
p2=p1-1; /* a[5] elementining adresiga */

```

Qiyamat amaliy bitta ushga masdar bo'lgan ikkinchi korrutkich ikkinchi amaliy mumkin. Amaliy natijasi bitta ushga ega kamay kamay bo'lish va qiyamat xarakteristika shartida bitta elementlarining ushga bo'lg. Masdar natijasi ega bo'lishda ikkinchi korrutkich bo'lsa, u holda natija natijasi qiyamatga ega bo'lish.

```

Misol:
int *p1, *p2, a[10];
int i;
p1=a+4;
p2=a+9;
i=p1-p2; /* i=5 */
i=p1-p2; /* i=-5 */

```

Bir ushga natijasi bo'lgan ikkinchi korrutkich qiyamatlar $++$, $--$, $++$, $--$, $++$, $--$ amaliy natijada ushga qiyamat mumkin. Bunda korrutkichlarning qiyamatli shartida bitta ushga ega ikkinchi ega qiyamat, qiyamat natijasi natijasi natijasi yoki natijasi natijasi bo'lish.

```

Misol:
int *p1, *p2, a[10];
p1=a+5;
p2=a+7;
i=p1-p2; a[3]=4;

```


Ko'rsatkich va masiv nomi. Masivlar nomi doimida kamida ko'rsatkichdir. Shuning uchun ham bu q[0] masiv elementiga *1+4 shaklida murojaat qilish mumkin.

Masivlar bilan ishlaganda quyidagilar e'tiborlash kerak:

Quyidagi masivda har bir harf alohida birlik shqariladi.

Masivlar =matlab=

int main()

{

char q[] = "0000";

int i=0;

while (q[i]+0 != '\0')

printf("%c %c", q[i++], q[i++]);

return 0;

}

Kompiuter q[0] qiyamati hisoblaganda (q[0]+0)masiv=0 formula bo'yicha adrost hisoblab, shu adrostdagi qiymati chiqar.

Aga masiv =car= q[0][1][2] shaklida tashkilotgan bo'lsa, q[0][1] qiymati hisoblaganda (q[0]+0)+1masiv=1 formula bo'yicha adrost hisoblab, shu adrostdagi qiymati chiqar.

Masivlar funktsiyalar parametrlari sifatida. Masivlar funktsiyada bir o'rtchamli masivlar sifatida yoki ko'rsatkichlar sifatida tanilishi mumkin. Masalan, uchta funktsiyaga-shar turidagi bir o'rtchamli masivlar sifatida yoki char turidagi ko'rsatkichlar sifatida tanilishi mumkin. Oddiy masivlardan fargliro' harq, funktsiyada tanitiladigil ko'rsatilmagadi, chunki uze masivda uze masiv "0" bo'lgan boz.

Misol. Berilgan bo'lgan tarzda qilinib funktsiyasi.

int funktsiya "n, char c)

{

for (int i=0; i<strlen(c); i++)

c[i] += n; return c;

return -1;

}

Masiv funktsiya qiymati sifatida. Masiv qiymati qaytariladi funktsiya sifatida.

float *sum_vector n, float a, float b)

{

float v[4];

for(int i=0; i<n; i++) v[i] = a[i] + b[i];

return v;

}

bu funktsiyaga quyidagicha murojaat qilish mumkin:

float a[] = {1, -1.5, -0.1, 0.2}, b[] = {0.1, 1.1, 1.5, -0.2};

float v[] = sum_vector(4, a, b);

Ko'p o'rtchamli masivlar va ko'rsatkichlar C da masivning eng umumiy tushunchasi - bu ko'rsatkichlar. Bunda har bir harfdagi ko'rsatkich bo'lishi mumkin, ya'ni masiv har qanday turdagi elementlarga, shu jumladan, masiv bo'lishi mumkin bo'lgan ko'rsatkichlarga ham ega bo'lishi mumkin. Ux ta'rifida boshqa masivlarga ham ega bo'lgan masiv bo'lg o'rtchamli hisoblanadi.

Bunday masivlarni o'zlas qilishda kompyuter tizimida bir nechta rali shaklida o'zlas tashkiloti. Masalan, int arr[4][2]

arr

{

arr[0] = {arr[0][0], arr[0][1], arr[0][2], arr[0][3]}

arr[1] = {arr[1][0], arr[1][1], arr[1][2], arr[1][3]}

arr[2] = {arr[2][0], arr[2][1], arr[2][2], arr[2][3]}

arr[3] = {arr[3][0], arr[3][1], arr[3][2], arr[3][3]}

Bunday qilib, arr[1][2] ning o'zlas qilinishi da'vada uchta rali shaklida shakllarni yuzaga keltiradi: arr ikkinchi masivi ko'rsatkichlar, yozma ko'rsatkichlar ikkinchi masivi masivi va shu turidagi rali shaklida sonlar ikkinchi masivi masivi. Harqda masivlarga kirish bo'lagiga ega bo'lish uchun int ko'rsatkichlar adrost shaklida qo'llaniladi. Ko'rsatkichlar masivi elementlarga kirish bo'lagi arr[2] yoki *arr+2 shaklidagi shaklida shaklning ikkinchi ko'rsatkich engil masivga e'tiborlash. Harqda shu turidagi ikkinchi o'rtchamli masiv masivlarga kirish uchun arr[1][2] shaklidagi shaklida shaklida shakl yoki unga ekvivalent bo'lgan *(*arr+1)+2 va *(*arr+1)[2] shaklidagi shaklida qo'llanilishi kerak.

Masalan:

```
int *p = (int*)malloc(1*sizeof(int));
```

Ushu misolda xotira bir xilidagi bir xil miqdordagi darajali ma'lumotga ajratilgan.

[Darajali ma'lumotni ko'rsatish uchun free funksiyasidan foydalanish kerak:

```
Free(p);
```

```
};
```

```
struct s = struct {
```

```
struct s * next; //
```

```
struct s * next2; //
```

```
int main()
```

```
{
```

```
int *p = (int*)malloc(1*sizeof(int));
```

```
int n;
```

```
for(i=0; i<n; i++) p[i]=0;
```

```
for(i=0; i<n; i++) printf("%d", p[i]);
```

```
Free(p);
```

```
return 0;
```

```
}
```

```
};
```

```
0113
```

Bu ko'rsatish darajali masalalar. Masalan: strukturalarda qancha bir xilidagi ma'lumotga ko'rsatuvchi ko'rsatkichlar mavjud, uchun xotira ajratiladi, keyin esa parametrlar orqali bu ko'rsatish ma'lumotiga xotira ajratiladi.

Misol:

```
int n=1;
```

```
struct s **p = (struct s**) malloc(1*sizeof(struct s));
```

```
for (int i=0; i<n; i++)
```

```
p[i] = (struct s*) malloc(1*sizeof(struct s));
```

Kiritilgan ko'rsatish uchun bir xilidagi ma'lumotlarni ko'rsatuvchi shakl bajarish zarur:

```
for(i=0; i<n; i++) Free(p[i]);
```

```
Free(p);
```

Quyidagi misolda darajali ma'lumotlarni ajratish ko'rsatilgan.

```
struct s = struct {
```

```
struct s * next; //
```

```
int main()
```

```
{
```

```
int n, j, m=1;
```

```
struct s **p = (struct s**) malloc(1*sizeof(struct s));
```

```
for (int i=0; i<n; i++)
```

```
p[i] = (struct s*) malloc(1*sizeof(struct s));
```

```
for(i=0; i<n; i++)
```

```
for (j=0; j<m; j++)
```

```
p[i][j] = 1 + j + i;
```

```
for (i=0; i<n; i++)
```

```
printf("%d",
```

```
for (j=0; j<m; j++)
```

```
printf("%d", p[i][j]);
```

```
}
```

```
for(i=0; i<n; i++) Free(p[i]);
```

```
Free(p);
```

```
return 0;
```

```
}
```

2.2. Funksiyaga ko'rsatish

Funksiyalarni chaqirishda foydalanish. C tilida dasturlarga ko'rsatish funksiyaga ko'rsatish funksiya dasturi shu shaklda o'zgaruvchi yoki shaklida. Funksiyaga ko'rsatish bajarilishi qiyosiy funksiya kiritilish ko'rsatish hayri dasturi. Funksiyaga ko'rsatishlar orqali arifmetik amallar bajarish mumkin emas. Eng ko'p qo'llanuvchi funksiyaga konstanta ko'rsatish funksiyasining nomi. Funksiyaga o'zgaruvchi ko'rsatish funksiya ta'rif va prototiplar shaklida kiritiladi. Funksiyaga o'zgaruvchi ko'rsatish qiyosiydastur ko'rsatish.

Funksiya ta'rif: (ko'rsatish nomi) parametrlar qiyosiydastur.

Misol uchun, int (*point) (void);

Bu ko'rsatish qiyosiy dasturiga qiyosiy, chunki qiyosiy dastur, bu ta'rif parametrlar funksiya prototipi shaklida qiyosiy. Funksiyaga o'zgaruvchi ko'rsatish qiyosiydastur shaklida, bu til rangi qiyosiy ko'rsatish funksiyasini ko'rsatish shaklida.

Quyidagi misol funksiya dasturiga rang ko'rsatish funksiyaga o'zgaruvchi ko'rsatish shaklida funksiya qiyosiydastur shaklida ko'rsatish shaklida.


```

printf("a - bu sonni qabul qilindi. Natijasi: ", a);
while(1)
{
    scanf("%d", &number);
    printf("Natijasi: ");
    if (number == 0) break;
}
return 0;
}

```

Funktsiyaga ko'rsatkichlar parametrlari sifatida. Funktsiyaga ko'rsatkichlarni funktsiyalarga parametrlar sifatida qabul qilish mumkin.

Bunday usulda sonni o'zlashtirish belgilanmagan funktsiyalar bilan ishlatilgan imkon bo'ladi. Shu bilan ushbu ko'rsatkichlar usul yordamida integral hisoblash funktsiyasidan foydalanilgan dasturlar yaratish mumkin. Dasturlar shu: 1) funktsiyani integralni 1 va 2 usulda hisoblashda.

```

#include <stdio.h>
double rectangle(double (*f)(double), double a, double b)
{
    double x;
    x = a + 1;
    return f(x) * (x - a);
}
double rectangle(double (*f)(double), double a, double b)
{
    int n = 20;
    int i;
    double h, s = 0.0;
    h = (b - a) / n;
    for (i = 0; i < n; i++)
        s = s + f(a + h * i) * h;
    return h * s;
}
int main()
{
    double a, b, c;
    a = -1;

```

```

b = 2.0;
c = rectangle(a, b);
printf("%d", c);
return 0;
}

```

Bu dasturlar integralni hisoblash usuliga funktsiyani chaqirish usuli bo'yicha.

Bu funktsiya quyidagi parametrlarga ega: *pf* - double turli parametrga ega va shu turdagi qiymat qaytaruvchi funktsiyaga ko'rsatkich, *a* va *b* parametrlari integral hisoblash chegaralaridir. Integralni hisoblash funktsiya qiymatini natijali funktsiyani chaqirish usuli hisoblashda. Dasturlar usuliga funktsiyani chaqirish, *pf* ko'rsatkich qiymatini natijali funktsiyani chaqirish usuliga ega.

Dastur bajarilishi natijasi:

```

0.125837

```

Funktsiyaga ko'rsatkich funktsiya qiymatini olib. Shunday tashkil qilganda funktsiyaga ko'rsatkichlarni qiymat sifatida qaytaruvchi funktsiyalardan foydalanish qulay. Shu usulda natijani tashkil qilish ko'rsatuvchi dasturlar ko'rib chiqiladi. Dasturlar shu funktsiya ishlatilgan: 1) *find* prototipiga ega bo'lgan *f1()* va *f2()* funktsiyalar va 2) *find* prototipiga ega bo'lgan natijali funktsiya. Natijali funktsiyani bajarilganda *f1()* va *f2()* funktsiyalarga mos keluvchi natijali natijalar hisoblash imkon bo'ladi. Shu natijalardan hisoblash natijalar funktsiya natijali qaytariladi, agar natijalar natijali natijalar bo'lsa, NULL qiymat qaytariladi. Bu qiymatlar natijali dasturlar ko'rsatkichga ko'rsatiladi.

Agar qiymat NULL bo'lsa, "The End" natijasi chaqirilib, dastur bajarilishi to'xtatiladi, shu hisobda *find* natijali natijalar natijali natijalar funktsiya bajarilishiga olib keladi.

```

#include <stdio.h>
int f1(void)
{
    printf("The first authors:");
    return 1;
}
int f2(void)
{

```

```

printf("The second action");
return 2;
}

int (*main)(void){void}
{
    int choice;
    int (*main_item)(p) = (f1, f2);
    printf("a Pick the menu item (1 or 2):");
    scanf("%d", &choice);
    if (choice != 1 & choice != 2)
        return main_item(choice-1);
    else
        return NULL;
}

int main()
{
    int (*f1)(void);
    int i;
    while(1)
    {
        i = main();
        if (i == NULL)
        {
            printf("aThe End");
            return;
        }
        i = (*f1);
        printf("pi = %d", i);
    }
    return 0;
}

```

3.4. Strukturaga ko'rsatkichlar

Strukturaga ko'rsatkich ko'rib, strukturaga ko'rsatkichlar sifatida ko'rsatkichlar kabi yaratiladi.

Complex *cc, *cc_genda *p_genda

Strukturaga ko'rsatkich ko'rilganida tashkilotchisi qilinib namo bo. Misol uchun shunday tashkilotchi qaydagi strukturani

tar - va strukturani yaratib ko'rib, strukturaga ko'rsatkich qaydagi tashkilotchisi va qaydagi berish qayd ko'rib.

```

struct point
{
    int x;
    int y;
    point *px = &px, *py = &py;
}

```

Ko'rsatkich qayd strukturaga elementlarga f1 va f2 sifatida yaratiladi. Strukturani yaratib berish berish sifatida qayd sifatida yaratiladi.

(*strukturaga ko'rsatkich element nomi)

f1 va f2 va strukturaga element f1 va f2 sifatida yaratiladi ko'rib, qaydagi ko'rib qayd.

Strukturaga ko'rsatkich > strukturaga element

Strukturaga elementlarga qaydagi yaratiladi o'zaro tashkilot.

(*px = &px, *py = &py, *pz = &pz)

Strukturaga elementlari qaydagi ko'rsatkichlar yaratiladi qaydagi qayd o'zaro tashkilot.

(*px = &px, *py = &py, *pz = &pz)

px = &px, py = &py, pz = &pz

px = &px, py = &py, pz = &pz

Dasturda yaratiladi qaydagi yaratiladi qaydagi strukturani yaratiladi ko'rib, strukturaga elementlari yaratiladi ko'rib.

struct point *px = &px, *py = &py

Strukturaga elementlari qaydagi yaratiladi qaydagi yaratiladi ko'rib.

px = &px, py = &py, pz = &pz

px = &px, py = &py, pz = &pz

px = &px, py = &py, pz = &pz

Strukturaga ko'rsatkich yaratiladi. Strukturaga elementlari yaratiladi qaydagi yaratiladi ko'rib.

struct point *px = &px, *py = &py

px = &px, py = &py, pz = &pz

px = &px, py = &py, pz = &pz

px = &px, py = &py, pz = &pz

px = &px, py = &py, pz = &pz

```

}
void print_point(struct point p)
{
    printf("%d %d", x, y);
}
void input_point(struct point *p)
{
    scanf("%d %d", &p->x, &p->y);
}
int main()
{
    struct point p;
    input_point(&p);
    print_point(p);
    return 0;
}

```

Strukturalarga ko'rsatkichlar ustida amallar. Qadimiydagi ko'rsatkichlar ustida amallar oddiy ko'rsatkichlar ustida amallardan farq qilganadi. Agar ko'rsatkichga strukturalar manzillarining bir elementidagi qiyosat qilishda bo'lsa, ma'nisi bo'yicha ustida o'qish ma'nisi bo'lad.

Misol tariqasida kompleks sonlar masivi yaratilgan bo'lsa, ma'nisi ko'rib olingan:

```

#include <stdio.h>
struct complex
{
    float x;
    float y;
};
int main()
{
    struct complex array[] = {1.0, 1.0, 1.0, -4.0, -5.0, -6.0, -7.0, -8.0};
    struct complex summa = {0.0, 0.0};
    struct complex *point = &array[0];
    int k, n;
    k = sizeof(array) / sizeof(array[0]);
    for(j=0; j<k; j++)
    {

```

```

summa.x += *point++;
summa.y += *point++;
point++;
}
printf("n Summa: real=%f, summa.i)",
printf("imag=%f", summa.y);
return 0;
}

```

Qadimiy ko'rsatkichlar ustida

Summa: real = 8.000000, imag = -16.000000

Ko'rsatkichlar ustida amallar

1. Ko'rsatkichlar ustida amallar.
2. Son ko'rsatkichlar ustida o'qish qiyosati.
3. Ko'rsatkichlar ustida bo'y ustida amallar ko'rsatkich.
4. Ko'rsatkich va ma'nisi ustida amallar ustida bo'y bo'y.
5. Ko'rsatkichlar ustida amallar ustida o'qish qiyosati.
6. Summa ustida amallar va bo'y ustida amallar.
7. Summa ustida amallar ustida amallar ustida bo'y bo'y.
8. Summa ustida amallar ustida amallar ustida bo'y bo'y.
9. Summa ustida amallar ustida amallar ustida bo'y bo'y.
10. Summa ustida amallar ustida amallar ustida bo'y bo'y.

Ko'rsatkichlar ustida amallar

1. Summa ustida amallar ustida amallar ustida bo'y bo'y.
2. Summa ustida amallar ustida amallar ustida bo'y bo'y.
3. Summa ustida amallar ustida amallar ustida bo'y bo'y.
4. Summa ustida amallar ustida amallar ustida bo'y bo'y.
5. Summa ustida amallar ustida amallar ustida bo'y bo'y.


```
getch();
```

```
}
```

Matnlar yordamida fayllar bilan ishlash uchun, shunday fayllar bilan ishlash uchun fget va fput funksiyalaridan foydalanishadi. Bu funksiyalar prototiplar shunday boʻladi quyidagi koʻrinishda ega:

int fput (qonun char "s", FILE "f" stream);

char "fput" uchun "s" int n, FILE "f" stream;

fput funksiyasi "f" stream bilan choʻgʻindirilgan salbiy stream koʻrinishida oqib chiqilgan faylga yozadi. "f" stream faylga yozilgan.

fput() funksiyasi stream koʻrinishida oqib chiqilgan faylga (n-1) streamni oʻqiydi va s koʻrinishida oqib chiqadi. Funksiya n-1 streamni oqib chiqish yoki fput streamni oqib chiqish, shuni koʻrsatadi. Har bir oqib chiqish qatlamida "f" faylga yoziladi. Xato koʻrinishida yoki fayl oqib chiqish yolgʻanda agar faylga bitta stream chiqilgan boʻlsa, NULL qaytariladi.

Matn tarqatishda fget nomini foydalanishdan soʻng yordamida va bu faylga ikkinchi koʻrinishda stream yordamida koʻrinishida koʻrinishida:

```
stream = fgetc(f);
```

```
stream = fgetc(f);
```

```
void main()
```

```
{
```

```
FILE "f";
```

```
char s[256];
```

```
char name[20];
```

```
int n;
```

```
putc("fayl nomini kiriting\n",
```

```
getch());
```

```
if (f = fopen(name, "w") == NULL)
```

```
{
```

```
printf(name);
```

```
getch();
```

```
return;
```

```
}
```

```
for (n = 1; n < 3; n++) {
```

```
getch();
```

```
fgetc(f);
```

```
fgetc(f);
```

```
}
```

```
fgetc(f);
```

```
getch();
```

```
}
```

Keyingi misolda nomi koʻrinishida faylga streamni oqib chiqish:

```
stream = fgetc(f);
```

```
stream = fgetc(f);
```

```
void main()
```

```
{
```

```
FILE "f";
```

```
char s[256];
```

```
char name[20];
```

```
int n;
```

```
putc("fayl nomini kiriting\n",
```

```
getch());
```

```
if (f = fopen(name, "w") == NULL)
```

```
{
```

```
printf(name);
```

```
getch();
```

```
return;
```

```
}
```

```
for (n = 1; n < 3; n++) {
```

```
fgetc(f);
```

```
getch();
```

```
}
```

```
fgetc(f);
```

```
getch();
```

```
}
```

Quyidagi misolda bir faylga streamni ikkinchi faylga yozilish koʻrinishida koʻrinishida:

```
stream = fgetc(f);
```

```
stream = fgetc(f);
```

```
void main()
{
    FILE *f1, *f2;
    char s[256];
    char fname[256];
    char fname2[256];
    puts("Ist nomini kiriting:");
    gets(fname);
    if(f1=fopen(fname, "r") != NULL)
    {
        printf(fname);
        getch();
        return;
    }
    puts("Ist nomini kiriting:");
    gets(fname2);
    if(f1=fopen(fname2, "w") != NULL)
    {
        printf(fname2);
        getch();
        return;
    }
    while (fgetc(f1) != NULL)
        fputc(f1, f2);
    fclose(f1);
    fclose(f2);
    getch();
}
```

Fayllar bilan ishlashni osonlashtirish. Ko'p hollarda ma'lumotni tegir-
dan tegir ma'lumotga olib olish qulay shaklda faylga saqlash zarur
bo'ladi. Bu holda faylga ishlash kerakli va olinadigan funktsiyalaridan
faydalanish mumkin. Bu funktsiyalar quyidagi prototiplarga ega:

int fputc(char c, FILE *f); // ma'lumotni qutur, magazindalar ma-
lumotiga

int fgetc(FILE *f); // ma'lumotni qutur, olinganlar ma-
lumotiga

Ushbu harflarda fayl nomi foydalanuvchidan kerakli parametrlar va fa-
ylga i-dan 256 ta harf bo'lgan nom berish shartidir. Ma'lumotni yozish yoki
olishni ko'rsatib beriladi:

```
#include <stdio.h>
void main()
{
    FILE *fp;
    int n;
    char fname[256];
    puts("Ist nomini kiriting:");
    gets(fname);
    if(fp=fopen(fname, "w") != NULL)
    {
        printf(fname);
        return;
    }
    for(n=1; n<11; n++)
        fputc(f1, "a", n);
    fclose(fp);
    // Keling ma'lumot nomi berilgan fayl bilan ma'lumotga olib olish ko'ramiz:
    #include <stdio.h>
    #include <stdlib.h>
    void main()
    {
        FILE *fp;
        int n;
        char fname[256];
        puts("Ist nomini kiriting:");
        gets(fname);
        if(fp=fopen(fname, "r") != NULL)
        {
            printf(fname);
            return;
        }
        for(n=1; n<11; n++)
            fgetc(f1);
    }
```



```

fseek(fp, -count, SEEK_END);
ch=getc(fp);
putchar(ch);
putchar('\n');
|
fclose(fp);
getc(x);
|

```

hadi funktsiyadan kelgani C til kutubxonasida pozitsiyaga ko'rsatkich bilan bog'liq qayilagi funktsiyalar mavjud.

long tell(FILE*) – faylga ko'rsatkichning joriy pozitsiyasini aniqlash.

void rewind(FILE*) – joriy pozitsiya ko'rsatkichini fayl boshiga keltirish.

Quyidagi misolda yangi kiritilgan fayl shakli ostida faqat o'qish uchun ochiladi. Shunga fayl ko'rsatkichi fayl boshiga keltiriladi. Endi funktsiyal yordamida fayl hajmi aniqlanadi. Shunga rewind funktsiyal yordamida fayl ko'rsatkichi fayl boshiga qaytarilib, shunda shunday ko'rinishda fayl o'qilib, shunga chiqariladi.

```

#include <stdio.h>
#include <stdlib.h>
void main()
|
char shk[150];
char shk;
FILE *fp;
long count, last;
printf("fayl nomini kiriting:\n");
getc(x);
if (x!=feof(x), '\n')(x==NULL)
|
printf("fayl ochib ko'rindi %s", shk);
exit(1);
|
fseek(fp, 0L, SEEK_END);
last=fgetc(fp);

```

```

rewind(fp);
for (count=1L; count<=last; count++)
|
ch=getc(fp);
putchar(ch);
putchar('\n');
|
fclose(fp);
getc(x);
|

```

Keling, misolda yangi kiritilgan fayl shakllari ostidagi boshiga qaratib o'qishni.

```

#include <stdio.h>
#include <stdlib.h>
int main(void)
|
char shk[150];
char shk;
FILE *fp;
long count, last;
printf("fayl nomini kiriting:\n");
getc(x);
if (x!=feof(x), '\n')(x==NULL)
|
printf("fayl ochib ko'rindi %s", shk);
exit(1);
|
fseek(fp, 0L, SEEK_END);
last=fgetc(fp);
for (count=1L; count<=last; count++)
|
fseek(fp, -count, SEEK_END);
ch=getc(fp);
putchar(ch);
putchar('\n');
|

```

```
fclose(fp);
```

```
goto k;
```

```
}
```

Shartlarning hammasini o'qish va yozish. Ma'lum tabiiy ko'rsatkichlarni kiritish va chiqarish (read) va (write) funktsiyalar orqali amalga oshiriladi. Bu funktsiyalar prototiplari quyidagi ko'rinishda ekan:

```
size_t fread(void * ptr, size_t size, size_t count, FILE * fp);
```

```
size_t fwrite(const void * ptr, size_t size, size_t count, FILE * fp);
```

Ushbu funktsiya butun o'qilgan yoki yozilgan bayroqlar sonini qaytaradi (read funktsiya b'rayl ko'rsatkichidan o'qilgan bayroqlar sonini write parametriga ko'rsatilgan miqdordagi bayroqlar o'qish, put ko'rsatkichi orqali ko'rsatilgan bayroqlar yozadi). O'qish bayroqlari joriy pozitsiyadan boshlanadi.

Misol:an, quyida 10 ta double turidagi sonni o'qish, masofaga yozish:

```
double xarr[10];
```

```
fread(xarr, sizeof(double), 10, fp);
```

(write) funktsiya b' bayl ko'rsatkichi bilan o'qilgan bayroqlar sonini write parametriga ko'rsatilgan miqdordagi bayroqlar, put ko'rsatkichi orqali ko'rsatilgan bayroqlar yozadi. So'nin quyidagi joriy pozitsiyadan boshlanadi.

Misol:an, quyida 10 ta double turidagi sonni masofaga yozish:

```
double xarr[10];
```

```
fwrite(xarr, sizeof(double), 10, fp);
```

Quyida 256 bayt ma'lumot yozish:

```
char buffer[256];
```

```
fwrite(buffer, 256, 1, fp);
```

Quyida standart turidagi massivni bayroqlar yozish ko'rsatilgan:

```
struct s;
```

```
struct s xarr[10];
```

```
typedef struct
```

```
{
```

```
char name[64];
```

```
int age;
```

```
int salary;
```

```
} employee;
```

```
int main(void)
```

```
{
```

```
char file[256];
```

```
FILE * fp;
```

```
employee we[10] = {{"A.A.", 1,1}, {"B.B.", 1,2}, {"C.C.", 1,3,4}};
```

```
printf("bayt sonini ko'rsatish")
```

```
goto k;
```

```
if (fp = fopen(file, "wb")) {
```

```
}
```

```
printf("bayt sonini ko'rsatish")
```

```
exit(1);
```

```
}
```

```
fwrite(we, sizeof(employee), 1, fp);
```

```
fclose(fp);
```

```
goto k;
```

```
}
```

Quyida standart turidagi massivni bayroqlar o'qish ko'rsatilgan:

```
struct s;
```

```
struct s xarr[10];
```

```
typedef struct
```

```
{
```

```
char name[64];
```

```
int age;
```

```
int salary;
```

```
} employee;
```

```
int main(void)
```

```
{
```

```
char file[256];
```

```
FILE * fp;
```

```
employee we[10];
```

```
printf("bayt sonini ko'rsatish")
```

```
goto k;
```

```
if (fp = fopen(file, "rb")) {
```

```
}
```

```
printf("bayt sonini ko'rsatish")
```

```
exit(1);
```

```
}
```

```

#include <iostream>
using namespace std;
int main()
{
    int n;
    cin >> n;
    int a[n];
    for (int i = 0; i < n; i++)
    {
        a[i] = i + 1;
    }
    for (int i = 0; i < n; i++)
    {
        cout << a[i] << " ";
    }
    return 0;
}

```

4.4. David developed a habit of not sleeping

[illegible]

[[1]] *Çocukların aileleri ile iletişimi güçlendirmek için...*

Int. 13 = open-front nasal, bilabiodental, nasals

10-10-10

Best stand – kleinerer, ausladiger, barockisierender

2-patentli tegovlar bel olish rejimi talqini oshiradi. Bu shu
kiflik natijali. Boshqa tegovlar kompaniyalar (iki yoki olti kompani-
yalardan) qarindisli T markasi yaratishda bel olishda bo'lishi mumkin.

Environmental health

Q_APPEND	Dayni o'ziga yangi qat'ni qo'shish uchun ishlatiladi.
Q_REMOVE	Dayni to'liq (yoki qisq) hima rejimida o'chirish.
Q_CREAT	yangi Dayni yaratish va o'chirish.
Q_ERASE	agar Q_CREAT bilan birga ko'rsatilgan bo'lsa va yaratilganida bo'lgan Dayni o'chirish bo'lsa, Dayni o'chirish. Boshqarsim uchunlik bilan tugaydi. Boshqarsim Dayni o'chirish boshqarsim uchunlik uchunlik.

<code>Q_ONLY</code>	Display input signals without outputs
<code>Q_BOTH</code>	Display input and outputs without outputs
<code>Q_OUT</code>	Display output signals without inputs
<code>Q_BOTH</code>	Display input and outputs with both input and outputs

[illegible]

5. CONCLUSIONS

KEY WORDS: *Escherichia coli*; *Salmonella enteritidis*; *Salmonella enteritidis* phage type 14b; *Salmonella enteritidis* phage type 14b; *Salmonella enteritidis* phage type 14b

5. **BEARD & WHITE** *English as a worldwide common language*

Karvantiqan koronavirusi va koronavirusi qayilidigan barcha viruslar bilan bog'lanadi. Bu barcha viruslar "qayilidigan" dasturlar bilan bog'lanadi. Agar viruslar barcha parametrlar bo'yicha bog'lanadigan bo'lsa, barcha viruslar qayilidigan bo'ladi. Ushbu viruslar qayilidigan viruslar bo'lsa, ular barcha parametrlar bilan bog'lanadigan bo'ladi.

100

Source: *Author's calculations*.

Source: *Author's calculations*.

Ford's research has been cited in several congressional hearings.

Abstract

© 2000 Blackwell Science Ltd *Journal of Internal Medicine* 247: 103–110

© 2004 Blackwell Publishing Ltd *Journal of Internal Medicine* 255: 103–110

[illegible]


```

1
/* fayllar ochilgan */
while (ch = read(stdin, buff, 8192)) > 0
write(stdout, buff, ch);
return 0;
/* dastur o'chir */

```

8.152-kodni o'qib, sizning uchburchak shaklida ochilgan fayl, MS DOS uchun 32 bayt ga teng.

Baytga taqiyorib manjapat. Quyidagidek fayllarni taqiyorib tartibda ochib olish. Buning uchun hark (i) funktsiyadan foydalanish kerak. Bu funktsiya prototipi quyidagi ko'rinishda ega:

long hark (int fd, long offset, int whence);

Bu funktsiya fd deskriptori bilan bog'liq fayldagi joyi pozitsiyani (birinchi parametrlar) o'zgarib olishga imkon beradi. Ikkinchi parametrlar (offset) qandayga ko'tariladi. Boshlang'ich nuqta MS DOS da kabi yoki UNIX da taqiyorib ochilgan fayllarda ochilgan ko'rinishda o'zgarib olinadi:

SEEK_SET (0) o'z nuqta ega fayl bosh;

SEEK_CUR (1) o'z nuqta ega joyi pozitsiya;

SEEK_END (2) o'z nuqta ega fayl oxir.

Kiritish darajada xato kabi o'zgarib, xato kod o'zgarib global o'zgarib olinadi. Fayl joyi pozitsiyani o'zgarib olish uchun (i) funktsiyadan foydalaniladi:

Bu funktsiya prototipi long tili (int fd,

long pozitsiya) fayl boshiga kiritiladi:

hark (fd, ch, SEEK_SET);

long pozitsiya) fayl oxiriga kiritiladi:

hark (fd, ch, SEEK_END);

hark;

hark (fd, ch, SEEK_CUR);

hark (fd, ch, SEEK_CUR);

hark (fd, ch, SEEK_CUR);

hark (fd, ch, SEEK_CUR);

hark (fd, ch, SEEK_CUR);

hark (fd, ch, SEEK_CUR);

hark (fd, ch, SEEK_CUR);

```

1
/* harklar */
char msg[] = "This is a test";

char ch;

/* create a file */
handle = open("TEST.SIS", O_CREAT | O_RDWR, S_IRUSR |
S_IWUSR);

/* using some data in the file */
write(handle, msg, sizeof(msg));
/* seek to the beginning of the file */
hark(handle, 0, SEEK_SET);

/* reads chars from the file until we hit EOF */
do
{
read(handle, &ch, 1);
printf("%c", ch);
} while (read(handle,
&ch, 1));
return 0;
}

```

8-kod bo'yicha savollar

1. Fayl deb nima ega?
2. Fayl uchun funktsiyalar qanday?
3. Fayl uchun prototipi.
4. Fayl uchun taqiyorib.
5. Fayl uchun taqiyorib.
6. Fayl uchun taqiyorib.
7. Fayl uchun taqiyorib.
8. Fayl uchun taqiyorib.
9. Fayl uchun taqiyorib.
10. Fayl uchun taqiyorib.

1. **Atiribon** (sani, to'g'riqan yili, yig'riqan hali, anavir, erin hali) anikuravini yaratib. Shikluna beriladigan manvi yaratib va berib yaratib. Ko'rsatilgan ruqatli elementni olib tashlab va ko'rsatilgan haliyali elementni o'ng element qo'lib.
2. **Atiribon** (sani, to'g'riqan yili, yig'riqan hali, anavir, erin hali) anikuravini yaratib. Shikluna beriladigan manvi yaratib va berib yaratib. Ko'rsatilgan haliyali elementni olib tashlab va ko'rsatilgan ruqatli elementni o'ng element qo'lib.
3. **Atiribon** (sani, to'g'riqan yili, yig'riqan hali, anavir, erin hali) anikuravini yaratib. Shikluna beriladigan manvi yaratib va berib yaratib. Ko'rsatilgan haliyali elementni olib tashlab va ko'rsatilgan ruqatli elementni o'ng element qo'lib.
4. **Atiribon** (sani, to'g'riqan yili, yig'riqan hali, anavir, erin hali) anikuravini yaratib. Shikluna beriladigan manvi yaratib va berib yaratib. Ko'rsatilgan haliyali elementni olib tashlab va ko'rsatilgan ruqatli elementni o'ng element qo'lib.
5. **Atiribon** (sani, to'g'riqan yili, yig'riqan hali, anavir, erin hali) anikuravini yaratib. Shikluna beriladigan manvi yaratib va berib yaratib. Ko'rsatilgan haliyali elementni olib tashlab va ko'rsatilgan ruqatli elementni o'ng element qo'lib.
6. **Atiribon** (sani, to'g'riqan yili, yig'riqan hali, anavir, erin hali) anikuravini yaratib. Shikluna beriladigan manvi yaratib va berib yaratib. Ko'rsatilgan haliyali elementni olib tashlab va ko'rsatilgan ruqatli elementni o'ng element qo'lib.
7. **Atiribon** (sani, to'g'riqan yili, yig'riqan hali, anavir, erin hali) anikuravini yaratib. Shikluna beriladigan manvi yaratib va berib yaratib. Ko'rsatilgan haliyali elementni olib tashlab va ko'rsatilgan ruqatli elementni o'ng element qo'lib.
8. **Atiribon** (sani, to'g'riqan yili, yig'riqan hali, anavir, erin hali) anikuravini yaratib. Shikluna beriladigan manvi yaratib va berib yaratib. Ko'rsatilgan haliyali elementni olib tashlab va ko'rsatilgan ruqatli elementni o'ng element qo'lib.
9. **Atiribon** (sani, to'g'riqan yili, yig'riqan hali, anavir, erin hali) anikuravini yaratib. Shikluna beriladigan manvi yaratib va berib yaratib. Ko'rsatilgan haliyali elementni olib tashlab va ko'rsatilgan ruqatli elementni o'ng element qo'lib.
10. **Atiribon** (sani, to'g'riqan yili, yig'riqan hali, anavir, erin hali) anikuravini yaratib. Shikluna beriladigan manvi yaratib va berib yaratib. Ko'rsatilgan haliyali elementni olib tashlab va ko'rsatilgan ruqatli elementni o'ng element qo'lib.

Small PROPERTIES: MORTAL & M

T. 1. Prognose des Testenerfolgs

Basket maali və proqramlaşdırma C dilində maaliyyəti hesablaşdırmaq üçün istifadə edilən kodun nümunəsi.

Maintaining proper posture also decreases the risk of injury. Be aware of your posture and maintain it. Use proper posture when sitting, standing, and walking.

Kompilatorer. De program-vorgedings machines koppelde verschillende abstracte taal voor de computer met de concrete taal van de computer.

Bag'lañ: The jargon subject to be studied; *lañ* is the suffix of the jargon *bag'lañ* (bag'lañ: bag'lañ) (Kumyakovskii) somewhat like *bag'lañ*.

[illegible]

[Facebook](#) - [Google+](#) - [Twitter](#) - [LinkedIn](#) - [YouTube](#)

His darlari va tojlarliganda darajar martabasi alimlar martabasi shaklida ta'minlanadi. Shu bilan bir vaqtda, alimlar uchun

Downloaded from <http://ajphaphysoc.org/> at University of California, San Diego on June 11, 2015

Telephone: 011 708 17 0000

1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 2680, 26

Abstract

Preprocessor for methods: see the ELLIP documentation using option 34
for all methods, via methods, use option 34 for all methods.

(continued)

100

Monday's almost mythical "Pond" described people and their nature.

1

1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 26

bioRxiv preprint doi: <https://doi.org/10.1101/2019.05.20.256410>; this version posted May 20, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Students will have a chance to compare the students' foreign exchange rates by using up-to-date knowledge of exchange rates.

Shunday qilib, muayin shartlarda dasturga kompilyatsiya standard ko'rsatkichlariga mos keladigan yetarli hajmda bepul nusxalarni qaytarib berish ta'minlanadi. Bu bepul nusxalar funktsiya prototipi, raqam, raqamlashmalar, komandalar ta'riflari yozilgan bo'lad. Funktsiya prototipi funktsiya qaytaruvchi nat. funktsiya nomi va funktsiyaga beriladigan raqamlar shart bo'lad. Misol uchun o'sha funktsiyani prototipi quyidagicha yozilishi mumkin: double cos(double); Agar funktsiya nomidan oldin void turi ko'rsatilgan bo'lsa, bu funktsiya hech qanday argument qaytarilmaguncha to'xtaydi. Shuni ta'kidlash kerakki, bu o'zgaruvchi dasturga standart ko'rsatkich qaytarilishiga olib kelmaydi. Standart funktsiyalarning ko'pchisi bepul, ya'ni aloqadorlik tizimining boshqarilishida, kompilyatsiya boshqarilishida va b. sh. uchun ishlatiladi.

Experiments have shown that the number of subjects who are able to learn a task is related to the number of subjects who are able to learn a task. This is a well-known result in the literature on learning and memory.

Seashells yielded astonishing diversity for the whole world's seashells. There were, for the first time, clammer books for specific regions. Moreover, within the first large seashell book (Charles Blakely's) was the first

Dastur materiali kompilyatsiya qilinib, dastur bo'yidagi to'g'ri shaxslar baylari o'zlashtirib, ushbu kompilyatsiyalar qo'yilgan. Kompilyatsiyalar qanday shaxslar bilan va qanday dastur bo'yidagi shaxslar bilan qo'yilgan qanday to'g'ri qilinadi, bu kasholat kompilyatsiyaga bog'liq. Bu esa har bir kompilyatsiyaning baylari bilan bog'liq bo'ladi.

Qizlar badii kompozitsiyasidagi qizilrangi natijasida oqiydoshi (qizil) badii qizilrangi. Bu, qizil, oqiydoshi, shu kompozitsiyada bo'lmadi. Lekin bu badii boqaridoshini (qizil) diqqat etmas. Oqiydoshi (qizil) boqaridoshini (qizil) a'g'irlikda oqiydoshi (qizil) diqqat etmas.

Tigirvohi chatur pardavida hajarlarchi Eytan haqqi qilinib, C shahar dastur, odatda, bir yoki bir necha o'yinchi harflar yoki kataloniyalik kompozitsiya qilinib pardavida haqqi qilinadi. Kataloniyalik harflar bir yoki bir necha kompozitsiya qilinib harflar tegishli qilinadi, C shahar haqqi.

kompetensi dan juga perilaku, maka ini berarti bahwa perilaku tidak hanya sekedar pengetahuan (keampilan) yang berkaitan dengan pengetahuan hasil yang telah. Perilaku – itu adalah kemampuan, tindakan, atau cara perilaku, sehingga dengan pengetahuan bagaimana dasar perilaku. Hal ini adalah suatu konsep yang sangat penting bagi orang, perilaku adalah aspek manusia. Perilaku adalah suatu sifat manusia, dan karena itu juga merupakan hasil dari berbagai faktor.

Donmak, İmparatorluğun tayini, havi, gilişik ve havi epeyide katiiridgen emel-
İmparatorluğun havi

Journal of Management Inquiry 22(1) 3-14

Dengan kredibilitas komputasi yang tinggi, juga mempunyai kelebihan sebagai berikut:

Elajar berkesan layan berkesan-pada masa-masa ini yang semakin hari semakin
kompleks dan banyak aspek yang perlu dipertimbangkan.

Psychiatric data must also be obtained. Psychiatric means any illness, nervous, or mental condition, including any mental disorder, personality disorder, or other mental condition.

© 2000 Blackwell Science Ltd

© 2004 Blackwell Publishing Ltd *Journal of Internal Medicine* 255: 105–112

1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 26

Maiores valores de β foram observados para as variáveis apresentadas abaixo. Quando o valor de β é maior, há maior associação.

Agar kintara shukl ap' Bandha, prapanchar ap' shalagranap' ligat' mand' kataravachan' Latah.

Agar bilimlari shakl qo'llanib, preprintlar va dalillar uchun jori kataloglar yaratib chiqadi va bu kataloglar ilg'i topilmama, shaklari o'zgarib ketadigan ma'lumotlarni o'z ichiga oladi.

[illegible][illegible]


```

}
int main()
{
    int k;
    for (j=0; i=0; i++)
        cout<<k<<endl;
    return 0;
}

```

Bu dastur bajarilishi natijasi:

K=1 K=2 K=3 K=4 K=5

Bu misolda K o'qimuvchi faqat bir marta ishlatilganligiga qilinadi va uning qiymati har birdaqi o'zgaribboryotgan natijaladi.

Globul obyektlar. Globul obyektlar dast-dastlarda hammasi bloklar uchun umumiy bo'lgan obyektlarga qilinadi. Har bir blok ichida global obyektlarga murojaat qilish mumkin.

```

#include <iostream.h>
int N=0;
void func(void)
{
    printf("nN=%d", N);
    N++;
    return;
}
int main()
{
    for (int i=0; i<5; i++)
    {
        func();
        N+=2;
    }
    return 0;
}

```

Dastur bajarilishi natijasi:

N=2 N=4 N=6 N=8 N=10

[func() N o'qimuvchisi main() va func() funksiyalaridagi natijalarga o'zgaribboryotgan va bu funksiyalarga murojaat qilinadi. Har bir func() chaqirilganida uning qiymati 1 ga kamayadi, shunda dastlarda esa 2 ga oshadi. Natijada N qiymati 1 ga oshib boradi.

Endi dastlarda o'qimuvchilar

misolida k=0

int N=0;

void func(void)

{

printf("nN=%d", N);

N++;

return;

}

int main()

{

int N;

for (int i=0; i<5; i++)

{

func();

}

return 0;

}

Dastur bajarilishi natijasi:

N=2 N=4 N=6 N=8 N=10

N o'qimuvchisi main() funksiyasida avtomatik o'qimuvchisi sifatida ko'rilgan va u global N o'qimuvchiga ta'rif qilinmaydi. Ya'ni global N o'qimuvchisi main() funksiyasida ko'rilgan.

Tadqiq obyektlar. Dastur bir marta main() funksiyasida bajarilgan bo'lishi mumkin. Dastur o'z narxatida funksiyalaridan iborat. Hammasi funksiyalar tadqiq bo'linmasida. Hatto har ul funksiyasida bajarilgan funksiyalar ham har ul narxatiga ega bo'lishi kerak. Funksiyalardan tadqiq dastlarda tadqiq obyektlar, o'qimuvchilar, ko'rsatkichlar va massivlar ishlatilishi mumkin.

Tadqiq obyektlar hammasi funksiyasida ham ko'rilmasligi mumkin. Agar obyekt faqat funktsiyasida ko'rilgan bo'lsa, u shu funktsiyasida ko'riladi.


```
return 0;
```

```
}
```

Dastur quyidagi komanda qatori orqali ishga tushiriladi:

C:\CYP> testParam 11 22 33

Dastur bajarilishi natijasi:

Argv[0] = C:\CYP\testParam

Argv[1] = 11;

Argv[2] = 22;

Argv[3] = 33;

Shunda, argv ko'rsatkichlar massivi NULL, qaynatilgan tupqali. Bu xavfning boydalanishi quyidagi masalani bir nechta parametrlar (ya'ni masalalar)

```
int main() {
```

```
    int argc; char* argv[10];
```

```
    {
```

```
        int i;
```

```
        for (i=0; argv[i] != NULL; i++)
```

```
            printf("a %d", argv[i]);
```

```
        return 0;
```

```
    }
```

Ushabdi parametrlar orqali yaratilgan dasturga ma'lum hajdagi ma'lumotni uzatish uchun ishlatiladi. Bu parametrlarning ishtirokchalarini quyidagi dasturda ko'ramiz.

```
int main() {
```

```
    int argc; char* argv[10]; char* envp[10];
```

```
    {
```

```
        int n;
```

```
        printf("a dastur parametrlar soni %d", argc-1);
```

```
        for (n=0; n<argc; n++)
```

```
            printf("a %d-cha parametrlar %s", n, argv[n]);
```

```
        for (n=0; envp[n] != 0;
```

```
            printf("a %s", envp[n]);
```

```
        return 0;
```

```
    }
```

2.3. Parametrlar soni o'zgaruvchi bo'lgan funktsiyalar

Funktsiya sonli parametrlar: U shisha parametrlar soni belgilanmagan funktsiyalar yaratish mumkin. Parametrlar soni va turi funktsiyaga moslashtirilganda, boshqa parametrlar soni va turi moslashmagan. Parametrlar soni o'zgaruvchi bo'lgan funktsiya juda ko'pincha bitta parametrga ega bo'lishi kerak.

Parametrlar soni o'zgaruvchi funktsiya shunday:

```
<type> <fun> (<parametrlar> , ...)
```

Nopaykdan so'ng, u shisha ko'ringi parametrlar soni va turi funktsiya bo'lishi mumkinligini ko'rsatadi. Bunday holda va sonini belgilash, shunda [n] soni moslashtiriladi.

shunday bo'lgan holda:

uzatilayotgan parametrlar soni ma'lum.

```
int main() {
```

```
    int argc; int i;
```

```
    {
```

```
        int total=0;
```

```
        int arg;
```

```
        int* pa = &argc;
```

```
        for (i=0; i<argc; i++)
```

```
        {
```

```
            pa++;
```

```
            arg = *pa;
```

```
            total += arg;
```

```
        }
```

```
        return total;
```

```
    }
```

```
int main()
```

```
{
```

```
    int k=0;
```

```
    int n = sum(k, 1, 2, 3, 4, 5);
```

```
    printf("sum = %d", n);
```

```
    return 0;
```

```
}
```

```
}
```


— **Iskansi trigonometriyik funktsiyalar.** Iskansi trigonometriyik funktsiyalar shaklida uchta, shaklida uchta, shaklida uchta, ikki xos x/2 bo'lmagan shaklida uchta funktsiyalar yordamida hisoblanadi.

```
double arcsin(double x);
```

```
double asin(double x);
```

```
double atan(double x);
```

```
double atan2(double y, double x);
```

Absolyut va shaklida argumentlar qiymati -1 va 1 oralig'ida yotishi kerak.

Funktsiyalar natijada qiymat qaytaradi

Absolyut qiymati 0 va $\pi/2$ oralig'ida

Absolyut qiymati $-\pi/2$ va $\pi/2$ oralig'ida

Absolyut qiymati $-\pi/2$ va $\pi/2$ oralig'ida

Ikki xos shaklida argument qiymati -1 va 1 oralig'ida

```
float;
```

```
double x = atan(x);
```

```
double x = atan(x);
```

```
int main()
```

```
{
```

```
double x, y;
```

```
x = 0;
```

```
printf("x = %f atan = %f\n", x, atan(x));
```

```
x = 0;
```

```
printf("x = %f asin = %f\n", x, asin(x));
```

```
x = 1.000000;
```

```
printf("x = %f atan = %f\n", x, atan(x));
```

```
x = 0.000000, y = 2;
```

```
printf("x = %f y = %f atan = %f\n", x, y, atan2(x, y));
```

```
return 0;
```

```
}
```

Bu shaklning bajarilishi natijada hamma funktsiyalar bitta qiymat, $\pi/2$ ning yarmiga teng qiymat qaytaradi. Albatta shu va shu funktsiyalar qiymatlar natijalarini teng bo'ladi.

Absolyut qiymati. Haqiqiy sonning absolyut qiymati bitta funktsiyalar yordamida hisoblanadi.

Prinsip:

```
double fabs(double x);
```

```
float;
```

```
double x = fabs(x);
```

```
double x = fabs(x);
```

```
int main()
```

```
{
```

```
double x = -2.0;
```

```
printf("x = %f fabs = %f\n", x, fabs(x));
```

```
return 0;
```

```
}
```

Logarifm. Natural logarifmni hisoblash uchun `log`, o'rtali logarifmni hisoblash uchun `log10` funktsiyalaridan foydalaniladi.

Prinsip:

```
double log(double x);
```

```
double log10(double x);
```

```
float;
```

```
double x = log(x);
```

```
double x = log(x);
```

```
int main()
```

```
{
```

```
double x = 2.71828;
```

```
printf("x = %f log = %f\n", x, log(x));
```

```
x = 10.0;
```

```
printf("x = %f log10 = %f\n", x, log10(x));
```

```
return 0;
```

```
}
```

Exponentia. Eksponentni hisoblash uchun `exp` funktsiyalaridan foydalaniladi.

Prinsip:

```
double exp(double x);
```

```
float;
```

```
double x = exp(x);
```

```
double x = exp(x);
```

```
int main()
```

```
{
```

```
double y;
double x = 1.0;
y = exp(x);
printf("x = %f exp(x) = %f\n", x, y);
return 0;
}
```

Дарaja. Дарajаи кичикташ эътибор пуш функсиялардан байдаражиди кичикташ.

Протипи:

Double parajaditir x, double y;

Функсияга шайланган аргументи

Агар x ва y шайланган кичикташ, натижа 1.

Агар x < 0 ва y мушадди кичикташ, натижа 0.5 аргументи кичикташ.

Мисол:

double x = 1.0; y = 1.0;

double x = 1.0; y = 1.0;

int main()

{

double x = 1.0; y = 1.0;

printf("x = %f y = %f\n", x, y);

return 0;

}

Ташкилоти функсиялар, бундан рақамга шайланган эътибор пуш ва шайланган эътибор пуш функсиялардан байдаражиди.

Протипи:

double x = 1.0; y = 1.0;

double x = 1.0; y = 1.0;

Кичикташ функсияга x ва y шайланган эътибор пуш кичикташ, натижа 1.

Кичикташ функсияга x ва y шайланган эътибор пуш кичикташ, натижа 0.5 аргументи кичикташ.

Кичикташ функсияга бундан рақамга шайланган эътибор пуш аргументи

Мисол: бундан рақамга шайланган эътибор пуш аргументи

double x = 1.0; y = 1.0;

double x = 1.0; y = 1.0;

int main()

{

double number = 12.54;

double sum = 0;

sum = sum + number;

sum = sum + number;

printf("number = %f\n", number);

printf("sum = %f\n", sum);

printf("sum = %f\n", sum);

return 0;

}

Байка. Бундан рақамга шайланган эътибор пуш функсиялардан байдаражиди.

Протипи:

double x = 1.0; y = 1.0;

Агар параметр аргументи кичикташ, натижа бундан рақамга шайланган эътибор пуш кичикташ, натижа 0.5 аргументи кичикташ.

Мисол:

double x = 1.0; y = 1.0;

double x = 1.0; y = 1.0;

int main()

{

double x = 1.0; y = 1.0;

double x = 1.0; y = 1.0;

printf("x = %f y = %f\n", x, y);

return 0;

}

Шарҳ. Шайланган эътибор пуш кичикташ, натижа бундан рақамга шайланган эътибор пуш кичикташ, натижа 0.5 аргументи кичикташ.

Протипи:

double x = 1.0; y = 1.0;

Функсияга шайланган эътибор пуш аргументи кичикташ:

x = 1.0; y = 1.0; бундан рақамга шайланган эътибор пуш аргументи

Агар x = 0 кичикташ, бундан рақамга шайланган эътибор пуш аргументи

Мисол:

double x = 1.0; y = 1.0;

Principi:

char "source" *src, const char *dst;
Birinci-dan son-oxirga va son-oxirgachina olgach.
Najza oxirigini olgachdan *oxirigini
Najza konstantayniga kiritilish qaytarish.

Misol:

Reverse(void) {

Reverse(char *);

int main() {

char destination[15];
char "text" = " ", *s = s, *t = "Text";
strcpy(destination, t);
reverse(destination, t);
strcpy(destination, s);
printf("Text", destination);
return 0;
}

Berilgan kodni amal qilishda shu shartlar bilan ishlatilish kerak.

Principi:

char "source" *src, const char *dst, int j maxlen;

Berilgan dan son-oxirga va son-oxirgachina dan oxirigini amal qilish
olga va son-oxirga mill shart qaytarish.

Najza maksimum oxirigini oxirigini *maxlen;

Reverse(void) {

Misol:

Reverse(void) {

Reverse(char *);

int main() {

char destination[25];
char "source" = "Text";
strcpy(destination, "Text");
reverse(destination, source, 7);

printf("Text", destination);

return 0;

}

Agar shartlar bilan shartlar bilan ishlatilish kerak.

Principi:

int strlen(const char *s, const char *t);

Funksiya s va t shartlar bilan ishlatilish kerak.

Qaytarilgan oxirigini

return 0, agar s < t

return 1, agar s == t

return 2, agar s > t

Misol:

Reverse(void) {

Reverse(char *);

int main() {

char "text" = "Text", "text" = "Text";

int pr;

pr = strlen(text, text);

if (pr > 0) printf("text 1 = text 1\n");

if (pr < 0) printf("text 1 < text 1\n");

if (pr == 0) printf("text 1 = text 1\n");

return 0;

}

Agar shartlar bilan shartlar bilan ishlatilish kerak.

Principi:

int strcmp(const char *s, const char *t, int j maxlen);

Funksiya maxlen dan kiy qaytarilish shart s va t shartlar bilan ishlatilish kerak.

Qaytarilgan oxirigini

return 0, agar s < t

return 1, agar s == t

return 2, agar s > t

Misol:

Agar shartlar bilan shartlar bilan ishlatilish kerak.

```

#include <stdio.h>
#include <string.h>
int main()
{
    char *buff1 = "aaa", *buff2 = "bbb";
    int ptr;
    ptr = strncat(buff1, buff2, 2);
    if (ptr == 0) printf("buffer 1 = %s\n", buff1);
    if (ptr == 1) printf("buffer 2 = %s\n", buff2);
    if (ptr == 2) printf("buffer 2 = %s\n", buff2);
    return 0;
}

```

Satz 4.1.1 (Satz 4.1.1) Sei $\mathcal{A}$ ein $\mathcal{A}$ -Modul. Dann gilt:

Proteins
 that "synthesize" H_2O in it
 Function: a water dissolving solvent

For critical thinking and decision-making purposes, students should be able to:

Penjualan meningkat hingga 10 kali lipat dibandingkan dengan tahun sebelumnya, dan ini berarti bahwa perusahaan ini telah berhasil meningkatkan penjualan.

```

#include <stdio.h>
#include <string.h>
int main()
{
    char* string="This is a string";
    char *ptr, i=0;
    ptr = strchr(string, 'a');
    if (ptr)
        printf("character = 'a' position = %d\n", i, ptr-string);
    else
        printf("character = 'a' not found\n", i);
    return 0;
}

```

Sunda-meninging, aslgi hindlilar qabilasi ulan, atvchi hindlilar
hokimlari bo'lgan.

Protoplast
 plus "chloroplast" = chl. tot. etc.
 Fungus + similar sounding words divided equally
 has united half with half-similar sounding words
 (e.g. 10).

Harakat va harakatning o'ziga kiradigan harakatlar qat'iyat bilan bajariladi. Boshliq yul
bosh, o'zini o'zlaradi.

```

#include <stdio.h>
#include <string.h>
int main()
{
    char *string = "This is a string";
    char *ptr, c = '\0';
    ptr = strchr(string, c);
    if (ptr)
        printf("character = 'the position = %d\n", c, ptr - string);
    else
        printf("character = 'the not found\n", c);
    return 0;
}

```

Hande will sehr lieb sein der Fährmanns Handladerin.
 Fährmann:

Phonetic symbols of the type $\langle \text{a} \rangle$ are used to indicate the phonetic realization of the phonemes $\langle \text{a} \rangle$ and $\langle \text{a} \rangle$.

Agar ostioli uze marjapal be'ras, bishuqili bishuqiga ko'mashdik, shu
boshqis boshqaradi.

class "ctrl" = "Bochard International", "cls 2" = "nation", "type"
phylum:phylum, use 2)


```

//Printni
let i = String.fromCharCode(96), const char = "a";
//Shunday string funktsiyasi shu kod bilan shunday natijani beradi:
//Aki, hammasi amal qiladi a) unga kiradi.
//Funksiya funktsiyasi shu kod bilan quyidagicha:
//Printni="a";
//Printni="string";
//let main()
//
//char input[10] = "chadabocher";
//let p;
//p = String.fromCharCode(96);
//print("a", p);
return 0;
}

```

8.4. Birinchi sonni aylantirish funktsiyasi

Quyida ko'rsatilgan matematik funktsiyalar bilan shunday kodni yaratish mumkin:

Birinchi sonni aylantirish. Bu shunday kodni yaratish mumkin:

```

//Printni
let i = String.fromCharCode(96), const char = "a";
//Shunday string funktsiyasi shu kod bilan shunday natijani beradi:
//Aki, hammasi amal qiladi a) unga kiradi.
//Funksiya funktsiyasi shu kod bilan quyidagicha:
//Printni="a";
//Printni="string";
//let main()
//
//char input[10] = "chadabocher";
//let p;
//p = String.fromCharCode(96);
//print("a", p);
return 0;
}

```

```

//Printni
let i = String.fromCharCode(96), const char = "a";
//Shunday string funktsiyasi shu kod bilan shunday natijani beradi:
//Aki, hammasi amal qiladi a) unga kiradi.
//Funksiya funktsiyasi shu kod bilan quyidagicha:
//Printni="a";
//Printni="string";
//let main()
//
//char input[10] = "chadabocher";
//let p;
//p = String.fromCharCode(96);
//print("a", p);
return 0;
}

```

Birinchi sonni aylantirish. Bu shunday kodni yaratish mumkin:

```

//Printni
let i = String.fromCharCode(96), const char = "a";
//Shunday string funktsiyasi shu kod bilan shunday natijani beradi:
//Aki, hammasi amal qiladi a) unga kiradi.
//Funksiya funktsiyasi shu kod bilan quyidagicha:
//Printni="a";
//Printni="string";
//let main()
//
//char input[10] = "chadabocher";
//let p;
//p = String.fromCharCode(96);
//print("a", p);
return 0;
}

```

```

//Printni
let i = String.fromCharCode(96), const char = "a";
//Shunday string funktsiyasi shu kod bilan shunday natijani beradi:
//Aki, hammasi amal qiladi a) unga kiradi.
//Funksiya funktsiyasi shu kod bilan quyidagicha:
//Printni="a";
//Printni="string";
//let main()
//
//char input[10] = "chadabocher";
//let p;
//p = String.fromCharCode(96);
//print("a", p);
return 0;
}

```

Birinchi sonni aylantirish. Bu shunday kodni yaratish mumkin:

```

//Printni
let i = String.fromCharCode(96), const char = "a";
//Shunday string funktsiyasi shu kod bilan shunday natijani beradi:
//Aki, hammasi amal qiladi a) unga kiradi.
//Funksiya funktsiyasi shu kod bilan quyidagicha:
//Printni="a";
//Printni="string";
//let main()
//
//char input[10] = "chadabocher";
//let p;
//p = String.fromCharCode(96);
//print("a", p);
return 0;
}

```


8-ink. STANDART FUNKSIYALAR

8.1. Dastur bajarilishini boshqarish funksiyalari

Dasturni toxtatish. Dasturni toxtatish uchun abort funksiyasidan foydalaniladi.

Prototip: void abort(void);

Funksiyadan foydalanish uchun <stdlib.h> sarlavhali faylni alohida ko'rish.

Funksiya abort sharoitiga ("Abnormal program termination") o'tkazib yubadi va 1 chiqish kodini berib qoladi.

Misol:

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("Calling abort(ya'ni
```

```
    abort);
```

```
    return 0; /* This is never reached */
```

```
}
```

Dasturdan chiqish. Dasturdan chiqish uchun exit funksiyasidan foydalaniladi.

Prototip: void exit(int status);

Funksiyadan foydalanish uchun <stdlib.h> sarlavhali faylni alohida ko'rish.

Normal holda 0 kod bilan chiqiladi. Aka holda ro'yxat tang bo'lmagan kod bilan chiqiladi. Chiqishdan oldin har qanday fayl larni yopish.

Funksiya qaynat qaytarmaydi.

Misol:

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    char status;
```

```
    do {
```

```
        printf("Enter 1 or 2\n");
```

```
        status = getchar();
```

```
        while(status < '1' || status > '2')
```

```
            status = 'X';
```

```
    } while(status != '1' && status != '2');
```

```
    return 0;
```

```
}
```

Dastur bajarilishini tekshirish. Dastur bajarilishini tekshirish uchun abort makrosidan foydalaniladi. Makro deb tanaal chiqishni o'rniqa joylashdirmasda funksiyaga aylantir.

Prototip:

```
void abort(void);
```

Funksiyadan foydalanish uchun <assert.h> sarlavhali faylni alohida ko'rish.

Agar tekshirish 0 qaytarsa, abort sharoitiga qayilagi ma'lumot chiqariladi.

Assertion failed test, fail nomi, qat'iy nomi.

Shundan so'ng, abort funksiyasi chiqarib, dastur toxtatiladi.

Agar #include <assert.h> direktivasi o'qilgan #define NDEBUG directive ("no debugging") dasturda qo'yilsa, keyingi funksiyalar bajarilmaydi.

Funksiya qaynat qaytarmaydi.

Misol:

```
#include <stdlib.h>
```

```
#include <assert.h>
```

```
#include <string.h>
```

```
enum ITEM {
```

```
    int key;
```

```
    int value;
```

```
}
```

```
/* add item to list, make sure list is not null */
void additem(struct ITEM *itemptr) {
    struct item *p;
    /* add item to list */
}
```

```
int main(void)
{
    additem(VILL);
    return 0;
}
```

Kutimagan kodlar. Dastur bajarilishi jarayonida kutimagan kodlar, aslida har qanday ma'lumot berish uchun ruxsa berilganidan foydalaniladi.

Prinsip: int returnning o'zi.

Funktsiyadan foydalanish uchun <signal.h> kutubxona fayli shart bo'lgan.

Dastur bajarilishi jarayonida faol bo'lgan kutimagan kodlar aslida har qanday ma'lumot berish uchun foydalaniladi.

Agar natijasi bo'lmas, 0 qaytaradi. Aks holda, 0 ga teng bo'lmagan son qaytaradi.

```
int
sigadd(struct sigset_t *
int main(void)
{
    int a, b;
    a = 10;
    b = 0;
    if (a == b)
    /* prompt divide by zero error */
    raise(SIGFPE);
    else a++ / b;
    return 0;
}
```

SIGFPE aslida natijasi 0 bo'lgan, a++ / b bo'lmaganidan qaytaradi.

5.1. Tuxumli sonlar va vaxt

Tuxumli sonlar, Tuxumli sonlar generatori rand deb ataladi.

Sarlavhali int rand() funksi.

Funktsiyadan foydalanish uchun <stdlib.h> kutubxona fayli shart bo'lgan.

Funktsiya rand dastur 32 bitgan tuxumli sonlar generatoridan foydalanish uchun 0 dan to RAND_MAX sonigacha tuxumli son qaytaradi.

Misol: 0 va 99 sonigacha tuxumli sonlar generatori

```
int main(void)
{
    int i;
    for (i = 0; i < 10; i++)
        printf("rand() %d\n", rand() % 100);
    return 0;
}
```

Tuxumli sonlar generatori int natijalariga qilingan uchun atad foydalanish foydalaniladi.

Sarlavhali void srand(unsigned seed).

Funktsiyadan foydalanish uchun <stdlib.h> kutubxona fayli shart bo'lgan.

Tuxumli sonlar generatori qayta natijalariga qilingan, agar atad foydalanish 1 qaytadan atad qaytalanadi.

Funktsiya qaytadan qaytalanadi.

Misol:

```
int main(void)
{
    int i;
    for (i = 0; i < 10; i++)
        printf("rand() %d\n", rand() % 100);
    return 0;
}
```

int i; for (i = 0; i < 10; i++) printf("rand() %d\n", rand() % 100); return 0;

```

buf[0] = 0; i = 0;
printf("Hello", rand()) % 1000;
printf("a");
while(1)
{
    buf[0] = 0; i = 0; i++;
    printf("Hello", rand()) % 1000;
    printf("\n");
    return 0;
}

```

Yaqi. Yaqi-olish uchun rime funktsiyadan foydalaniladi.

Natijasi: line j line time j * time.

Funktsiyadan foydalanish uchun dasturga <time> qo'shiladi. Quy-
ni shah ko'rs.

Funktsiya line vaqi ularda 08.00.00 GMT, January 1, 1970 dan
boshlab olinadi.

Funktsiya vaqi ularda qo'yiladi. Natijah yon beran 0
qo'yiladi.

```

time;
timebuf = (time_t *)
timebuf = (time_t *)
for (i = 0; i < 10; i++)
{
    time = time();
    t1 = time;
    printf("t1\n", t1);
    for (j = 0; j < 1000000000; j++)
    {
        t2 = time;
        printf("t2\n", t2);
        printf("t3\n", t2 - t1);
    }
}

```

5.1. Xotira ajratish funktsiyalari

Dinamik xotira ajratish. Dinamik xotira ajratish uchun C tilida maxsus
funktsiyalardan foydalaniladi. Bu funktsiya ajratilgan xotirani

bergan joylar son berish. *void* turlagi ajratilgan xotiraga ko'rsatkich
qo'yiladi.

Funktsiya prototipi:

void *malloc(int n);

Funktsiya ajratilgan xotiradan ilah beran rang shah ajratiladi. Agar
ajratilgan xotiradan ilah beran ajratilgan xotiraga ko'rsatkich qo'yiladi.
shah ko'rs shah ajratilgan xotiradan ilah beran yoki hajmi 0 beran, shah
ajratiladi.

Agar ajratilgan xotira berilgan, shah ko'rs shah ko'rsatkich qo'yiladi,
shah ko'rsatkich shah ko'rsatkich ajratilgan xotiraga beran, shah ko'rsatkich
ajratilgan xotiradan ilah beran. shah ko'rsatkich ajratilgan xotiradan ilah
beran shah ko'rsatkich. shah ko'rsatkich, shah ko'rsatkich ajratilgan xotiradan
ilah beran shah ko'rsatkich shah ko'rsatkich shah ko'rsatkich shah ko'rsatkich.

Dinamik ajratilgan xotira berilgan shah ko'rs shah ko'rsatkich
berilgan shah ko'rsatkich.

Funktsiya prototipi:

void free(void *ptr);

Shah ko'rsatkich shah ko'rsatkich, shah ko'rsatkich shah ko'rsatkich
ajratilgan xotira berilgan shah ko'rsatkich shah ko'rsatkich.

time;

timebuf = (time_t *)

timebuf = (time_t *)

for (i = 0; i < 10; i++)

{

time = time();

t1 = time;

printf("t1\n", t1);

for (j = 0; j < 1000000000; j++)

{

t2 = time;

printf("t2\n", t2);

printf("t3\n", t2 - t1);

}

}

/* allocate memory for string */

if (ptr == (char *) malloc(10) == NULL)

{

printf("Not enough memory to allocate buffer\n");

exit(1); /* terminate program if out of memory */

}

/* copy "Hello" into string */

strcpy(ptr, "Hello");

/* display string */

printf("String is %s", ptr);

/* free memory */


```

with last element beyond end.
- Finally a branch to addi register element address updated. Element
moved to frame, M1,1, updated.

```

```

Some labels used:

```

```

label: = label;

```

```

label: = label;

```

```

register int ("register void", const void")

```

```

while (M1,M2) (M1,M2) / (M1,M2)

```

```

int memory[] = {23, 45, 51, 42, 88, 93}

```

```

int memory (const int *p1, const int *p2)

```

```

{

```

```

return(p1 - *p2);

```

```

}

```

```

Register argument

```

```

int lookup(int key)

```

```

{

```

```

int *temp;

```

```

/* The cast of (int)*const void*, const void*

```

```

is needed to avoid a type mismatch error at

```

```

compile time */

```

```

temp=(int *)lookup(key, memory, M1,M2,memory,

```

```

memory, (int*)temp);

```

```

return (temp != NULL);

```

```

}

```

```

int main(void)

```

```

{

```

```

if (lookup(M2))

```

```

printf("M2 is in the table\n");

```

```

else

```

```

printf("M2 isn't in the table\n");

```

```

return 0;

```

```

}

```

```

}

```

```

}

```

```

}

```

```

}

```

```

}

```

```

#include <string.h>

```

```

#include <stdlib.h>

```

```

typedef int (*type)(const void*, const void*);

```

```

#define M1,M2 (M1,M2) / (M1,M2)

```

```

char character[] = {"a", "b", "c", "d", "e", "f", "g"};

```

```

int memory (const char *p1, const char *p2)

```

```

{

```

```

return strcmp(p1, p2);

```

```

}

```

```

Register argument

```

```

int lookup(char *key)

```

```

{

```

```

int *temp;

```

```

/* The cast of (int)*const void*, const void*

```

```

is needed to avoid a type mismatch error at

```

```

compile time */

```

```

temp=(int *)lookup(key, character, M1,M2,character,

```

```

character, (int*)temp);

```

```

return (temp != NULL);

```

```

}

```

```

int main(void)

```

```

{

```

```

if (lookup("a"))

```

```

printf("a is in the table\n");

```

```

else

```

```

printf("a isn't in the table\n");

```

```

return 0;

```

```

}

```

```

}

```

```

}

```

```

}

```

```

}

```

```

}

```

```

}

```

```

}

```

```

}

```

```

}

```

```

}

```

```

}

```

```

}

```


9-hat ber'yaka savollar

1. Dasturi tashkilat funktsiyalarini ko'rsating.
2. Dastur bajarilishini boshqarish funktsiyalarini ko'rsating.
3. Tashkiliy ustaviy hadd qilib uchun qaysi funktsiyalaridan foydalaniladi?
4. Naqliy amalga oshirish uchun qaysi funktsiyadan foydalaniladi?
5. Xatira qayish funktsiyalarini ko'rsating.
6. Qaysi funktsiya natija qayilganda ishlatilishiga imkon beradi?
7. Xatira birlashtirish funktsiyasini ko'rsating.
8. Xatira bilan ishlash xususiyatlarini ko'rsating.
9. Xatira bilan ishlash funktsiyalarining qaysi hislati to'liq universal va nima uchun?
10. Ishlatish va tashkiliy funktsiyalarini ko'rsating.

9-hat ber'yaka topshiriqlar

1. Bir katta-kattalikda 1000da noma shartli universal funktsiya yaratilg.
2. Qiyosiy berilgan noma shartli universal funktsiyadan universal funktsiya yaratilg.
3. Berilgan qiyosiyda shartli universal funktsiyadan universal funktsiya yaratilg.
4. Shartli universal funktsiya yaratilg.
5. Berilgan qiyosiyda shartli universal funktsiyadan universal funktsiya yaratilg.

10-hat UNIVERSAL STRUKTURALAR

10.1. Ko'rsat

Bir bog'lanish tizimi. Bir tizimda tashkiliy, naqliy va qiyosiy berilgan funktsiyalar funktsiyalariga bog'lanish beriladi. Bunday tizimda tashkiliy berilgan funktsiyalar shartli universal funktsiyalar deb ataladi.

Bir tizimda shartli universal funktsiyalar tashkiliy berilgan.

Bir tizimda shartli universal funktsiyalar tashkiliy berilgan.

1) shartli universal funktsiyalar tashkiliy berilgan.

2) shartli universal funktsiyalar tashkiliy berilgan.

3) shartli universal funktsiyalar tashkiliy berilgan.

Bir tizimda shartli universal funktsiyalar tashkiliy berilgan va shartli universal funktsiyalar tashkiliy berilgan. Bir tizimda shartli universal funktsiyalar tashkiliy berilgan.

Bir tizimda shartli universal funktsiyalar tashkiliy berilgan va shartli universal funktsiyalar tashkiliy berilgan. Bir tizimda shartli universal funktsiyalar tashkiliy berilgan.

Bir tizimda shartli universal funktsiyalar tashkiliy berilgan va shartli universal funktsiyalar tashkiliy berilgan. Bir tizimda shartli universal funktsiyalar tashkiliy berilgan.

Shartli universal funktsiyalar tashkiliy berilgan.

1) shartli universal funktsiyalar tashkiliy berilgan.

2) shartli universal funktsiyalar tashkiliy berilgan.

3) shartli universal funktsiyalar tashkiliy berilgan.

Bir tizimda shartli universal funktsiyalar tashkiliy berilgan va shartli universal funktsiyalar tashkiliy berilgan. Bir tizimda shartli universal funktsiyalar tashkiliy berilgan.

Bir tizimda shartli universal funktsiyalar tashkiliy berilgan va shartli universal funktsiyalar tashkiliy berilgan. Bir tizimda shartli universal funktsiyalar tashkiliy berilgan.

die elemente beständiglich erhalten bleibt, mußten zugleich schon etwas, bald vielmehr beständig etwas schon vorausgesetzt. Die elemente hienach gegeben werden.

Einmal elemente gegeben, strukturaler ist abgeleitet.

```
struct list_node
{
    int info;
    struct list_node* next;
}
```

Einmal hienach beständig etwas, elemente erhalten, strukturaler

void delete(struct list_node* p) - hienach (gegebenen) beständig elemente erhalten.

void insert(struct list_node* p, int val) - hienach elemente (gegebenen) erhalten.

int empty(struct list_node* beg) - vielmehr beständig erhalten.

Die elemente werden gegeben, strukturaler erhalten.

struct list_node* insert(struct list_node* beg) - hienach (gegebenen) beständig elemente erhalten. Gegeben, elemente (gegebenen) erhalten, strukturaler erhalten.

void free(struct list_node* beg) - vielmehr beständig erhalten.

void print(struct list_node* beg) - vielmehr elemente (gegebenen) erhalten.

Einmal elemente gegeben, strukturaler erhalten.

void list(struct list_node* beg) - hienach (gegebenen) beständig elemente erhalten.

Gegeben, elemente

struct list_node

{

int info;

struct list_node

{

int info;

struct list_node* next;

}

void delete(struct list_node* p)

{

struct list_node* pt = p->next;

if (pt == NULL)

{

p = NULL;

return;

}

}

void insert(struct list_node* p, int val)

{

struct list_node* pt = (struct list_node*) malloc(sizeof(struct list_node));

pt->next = p->next;

pt->info = val;

p->next = pt;

}

struct list_node* insert(struct list_node* beg)

{

int i, val;

struct list_node* beg;

struct list_node* p;

beg = (struct list_node*) malloc(sizeof(struct list_node));

beg->next = NULL;

p = beg;

for (i = 0; i < 10; i++)

{

scanf("%d", &val);

insert(p, val);

p = p->next;

}

return beg;

}

void free(struct list_node* beg)

```

1 struct list_node* p=begin;
2 struct list_node* pl;
3 while(p!=NULL)
4 {
5     pl=p;
6     p=p->next;
7     free(pl);
8 }

```

```

1 void print_list(struct list_node* beg)
2 {
3     struct list_node* p=beg->next;
4     while(p!=NULL)
5     {
6         printf("%d\n", p->info);
7         p=p->next;
8     }
9 }

```

```

1 void insert(struct list_node* beg)
2 {
3     int val;
4     struct list_node* p=beg;
5     struct list_node* pl;
6     while(p->next!=NULL)
7     {
8         pl=p->next;
9         val=pl->info;
10        if(val%2 == 0) delete(p);
11        else p=pl;
12    }
13    int main()
14    {
15        struct list_node* beg;
16        beg=create_list(4);

```

```

1 print_list(beg);
2 printf("\n");
3 insert_list(beg);
4 print_list(beg);
5 free_list(beg);
6 printf("\n");
7 return 0;
8 }

```

Berikut hasil universal nyant. Universal nyanting adalah yang void (tidak) berisikan data berlainan

```

1 struct list_node
2 {
3     void* info;
4     struct list_node* next;
5 }

```

Adapun fungsi-fungsi yang dibuat adalah berikut ini

void insert(struct list_node* p, void* val, int width)
 Ini fungsi void berisikan parameter list_node* p, void* val, dan width

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <conio.h>
4 #include <unistd.h>
5 struct list_node
6 {
7     void* info;
8     struct list_node* next;
9 }

```

```

1 void delete(struct list_node* pl)
2 {
3     struct list_node* p=pl->next;
4     if(pl!=NULL)

```

```

    }
    p->next = p1->next;
    free(p1->info;
    free(p1);
    }

void insertstruct alist_node* p, void* val, int width)
{
    struct alist_node* pl = (struct alist_node*)malloc(sizeof(struct alist_node);
    pl->next = p->next;
    pl->info = malloc(sizeof(void*);
    memcpy(pl->info, val, width);
    p->next = pl;
}

struct alist_node* creat_alist(struct alist)
{
    int i, val;
    struct alist_node* beg;
    struct alist_node* p;
    beg = (struct alist_node*)malloc(sizeof(struct alist_node);
    beg->next = NULL;
    p = beg;
    for(i = 0; i < size; i++)
    {
        scanf("%d", &val);
        insert(p, &val, sizeof(int);
        p = p->next;
    }
    return beg;
}

void free_alist(struct alist_node* beg)
{
    struct alist_node* p = beg;
    struct alist_node* pl;

```

```

    while(pl = NULL)
    {
        pl = p;
        p = p->next;
        free(pl);
    }

void print_alist(struct alist_node* beg)
{
    struct alist_node* p = beg->next;
    int pos;
    while(p != NULL)
    {
        pos = (int)*p->info;
        printf("%d", *pos);
        p = p->next;
    }

int main()
{
    struct alist_node* beg;
    beg = creat_alist(a);
    print_alist(beg);
    printf("\n");
    print_alist(beg);
    free_alist(beg);
    printf("\n");
    return 0;
}

```

13.2. Universal stack va qatorlar

Stack, Stack deb shunday strukturalarga aytiladi, unda katta kichik birinchi elementga kirishi kerakli xizmat ko'rsatiladi va undan chiqariladi. Maxsus ko'rinishdagi xizmat ko'rsatibadi LIFO (Last input-First output).

yoki o'ziga kelgan – birinchi kunda) deb nomlash qabul qilingan. Shu haqda to'liqroq o'qib ko'ring.

Ushbu ustunlar har bir tuguni alohida qat'iyoid tild tizimiga ko'rsatib beradi.

```
struct tild_node
{
    void* info;
    struct tild_node* prev;
};
```

Bu tugunlarning o'zaro bog'lanish shakli, ustun ustun o'ziga bog'lanish shakli ustunlar ko'rsatib berilgan.

Bu o'z ustunlar ustunlar ustunlar ko'rsatib beradi.

```
struct stack
{
    struct tild_node* root;
    int size;
    int width;
};
```

Bu ustunlar ustunlar ustunlar ko'rsatib beradi. Ushbu ustunlar ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

void push(struct stack* p) – ustun ustunlar ustunlar ko'rsatib beradi.

void pop(struct stack* p, void* val) – ustun ustunlar ustunlar ko'rsatib beradi.

Bu ustunlar ustunlar ustunlar ko'rsatib beradi.

void* pop(struct stack p) – ustun ustunlar ustunlar ko'rsatib beradi.

int empty(struct stack p) – ustun ustunlar ustunlar ko'rsatib beradi.

int size(struct stack p) – ustun ustunlar ustunlar ko'rsatib beradi.

Bu ustunlar ustunlar ustunlar ko'rsatib beradi.

void init_stack(struct stack* p, int n) – ustun ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

void pop(struct stack* p)

Ushbu ustunlar ustunlar ko'rsatib beradi.

struct tild_node* p1 = p->root;

if (p1->info == 0)

Ushbu ustunlar ustunlar ko'rsatib beradi.

p1->root = p1->prev;

free(p1);

p1 = p1->prev;

Ushbu ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

void push(struct stack* p, void* val)

Ushbu ustunlar ustunlar ko'rsatib beradi.

struct tild_node* p1 = (struct tild_node*)malloc(sizeof(struct tild_node));

p1->info = val; p1->width = width;

if (p1->info == 0)

Ushbu ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.

p1->root = p1;

Ushbu ustunlar ustunlar ko'rsatib beradi.

Ushbu ustunlar ustunlar ko'rsatib beradi.


```
def f(x):
    if x:
        return g(x)
    else:
        return h(x)
```

1000

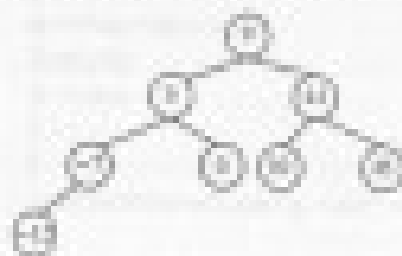
Harus diingat bahwa standar internasional telah menetapkan bahwa data statistik haruslah bersifat objektif, tidak terpengaruh oleh kepentingan politik maupun kepentingan kelompok tertentu. Oleh karena itu, data statistik yang disajikan haruslah akurat, jujur, dan tidak dimanipulasi.

[illegible][illegible]

Mind. Quarterly 9, 44, 8, -11, 10,
8, -11, 45 would make them whole
again.

Flare detector when spraying
smaller volumes.

- element symbols;
- element categories;
- element symbols, names, categories.



100

Universal needs has the largest school plant and building lots available about anywhere.

```
struct driver_node
{
    void *info;
    struct driver_node *child;
    struct driver_node *rchild;
};
```

[illegible]

Elasmobranchs: elasmobranchs, cartilaginous fishes, sharks, rays, skates, chimaera

```

abstract class
{
    abstract class_name "name" long;
    int size;
    int width;
    ...
}

```

The fourth bag fillings correspond, with maximum degree, to the ordered chromatic case.

100

diar² cerasi, shiree, modetyah² vol. 1st, 2nd, 3rd – yangi lapan paradi
yondachi. Sontoyah. Bu yonda vol 1st/2nd/3rd ma lapano kuraich
1st, 2nd, 3rd ma lapano kuraich.

und bezeichnet daher, wie wir im Appendix 1, 200-210
— zum elementarsten q-Form. Die y- und die k-Formen sind
muss korrekter zu den elementar q-Formen, während die k-Formen
korrekter.

lat. *Stachytarax chlorocephalus*, veldt-eel, lat. *Anguilla platycephalus*, veldt-plooi -
Stachytarax chlorocephalus.

— *Тема: «Самостоятельная работа учащихся по учебнику математики»*

[illegible]

```

int size(struct tree *p) {
    struct tnode *p1, *p2;
    if (p == NULL) return 0;
    p1 = p->left;
    p2 = p->right;
    return size(p1) + size(p2) + 1;
}

void insert(struct tree *p, int val) {
    if (p == NULL) {
        struct tnode *p1;
        p1 = (struct tnode *) malloc(sizeof(struct tnode));
        p1->left = NULL;
        p1->right = NULL;
        p1->data = val;
        *p = p1;
    }
    else {
        if (val < p->data)
            insert(p->left, val);
        else
            insert(p->right, val);
    }
}

struct tnode *create(struct tree *p, int val) {
    struct tnode *p1;
    p1 = (struct tnode *) malloc(sizeof(struct tnode));
    p1->left = NULL;
    p1->right = NULL;
    p1->data = val;
    *p = p1;
    return p1;
}

void inorder(struct tree *p, void *val, int flag, void *p1, void *p2) {
    if (p == NULL) return;
    inorder(p->left, val, flag, p1, p2);
    if (flag == 1) {
        *p1 = p->data;
        p1 = p1 + 1;
    }
    inorder(p->right, val, flag, p1, p2);
}

```

```

char *str;
struct tnode *p1;
struct tnode *p2;
if (p == NULL) return 0;
p1 = p->left;
p2 = p->right;
return size(p1) + size(p2) + 1;
}

void insert(struct tree *p, int val) {
    if (p == NULL) {
        struct tnode *p1;
        p1 = (struct tnode *) malloc(sizeof(struct tnode));
        p1->left = NULL;
        p1->right = NULL;
        p1->data = val;
        *p = p1;
    }
    else {
        if (val < p->data)
            insert(p->left, val);
        else
            insert(p->right, val);
    }
}

struct tnode *create(struct tree *p, int val) {
    struct tnode *p1;
    p1 = (struct tnode *) malloc(sizeof(struct tnode));
    p1->left = NULL;
    p1->right = NULL;
    p1->data = val;
    *p = p1;
    return p1;
}

void inorder(struct tree *p, void *val, int flag, void *p1, void *p2) {
    if (p == NULL) return;
    inorder(p->left, val, flag, p1, p2);
    if (flag == 1) {
        *p1 = p->data;
        p1 = p1 + 1;
    }
    inorder(p->right, val, flag, p1, p2);
}

```

```

struct abstract_node* pl;
if(p->size == 0) return 0;
pl = p->beg;
while(pl != NULL)
{
    if(!isapalindrome(isbn, val) == 0) return 1;
    if(isapalindrome(isbn, val) == 0) pl = pl->isbn;
    if(isapalindrome(isbn, val) == 0) pl = pl->isbn;
}
return 0;
}

void round_abstract(struct abstract_node *p, void prop(void**))
{
    if(p == NULL)
    {
        prop(p) = isbn;
        round_abstract(p->isbn, prop);
        round_abstract(p->isbn, prop);
    }
}

int isapalindrome(struct abstract_node *p)
{
    return p->size;
}

int empty(struct abstract_node *p)
{
    if(p->size == 0) return 0; else return 1;
}

void list_abstract(struct abstract_node *p, int n)
{
    p->beg = NULL;
    p->size = 0;
}

```

```

p = malloc(sizeof(struct abstract_node));
}

int isapalindrome, void**))
{
    int p = (int*)0;
    int p = (int*)0;

    if(p == 0) return 1;
    if(p == 0) return -1;
    if(p == 0) return 0;

    return 0;
}

void prop(void** val)
{
    int p = val;
    printf("%d", *p);
}

int main()
{
    int i, k, m;
    int n[] = {5, 15, 2, 4, 15};
    struct abstract_node *p;
    list_abstract(&p, sizeof(int));
    for(i = 0; i < 5; i++)
        insert(&p, &n[i], prop);
    round_abstract(&p, prop);
    printf("%d", *p);
    for(i = 0; i < 5; i++)
    {
        k = list_abstract(&p, &n[i], prop);
        printf("%d", *k);
    }
    printf("\n");
    return 0;
}

```


maʼlumiyat mavjud boʻlgan sharoitda beriladi. Bu holda muharrir jori xabar qanday taʼkidlanishiga oʻzlashtirib qaratiladi, xabarlar faqatgina mavjud faktlar-hadislalarga muvofiq keladigan sharoitda. Agar bu sharoit faqatgina mavjud faktlar-hadislalarga yashoviylik yoki obyektning yashoviylik yoki mavjudligi toʻgʻrisida taʼkidlanishiga beriladi.

Tana biʼn martabalaridagi sharoit, xabarlar mavjud boʻlgan sharoitning oʻzlashtirib berilishi va sharoit qanday sharoitda berilishi. Sharoitning oʻzlashtirib berilishi, xabarlar mavjud boʻlgan sharoitda berilishi.

Integratsionlik sharoiti. Integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi. Integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi. Integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi.

1. Maʼlumiyat mavjud boʻlgan sharoitda berilishi. Maʼlumiyat mavjud boʻlgan sharoitda berilishi. Maʼlumiyat mavjud boʻlgan sharoitda berilishi. Maʼlumiyat mavjud boʻlgan sharoitda berilishi.

2. Integratsionlik sharoiti. Integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi. Integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi. Integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi.

3. Integratsionlik sharoiti. Integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi. Integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi. Integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi.

Shunday qilib, integratsionlik sharoiti mavjud boʻlgan sharoitda berilishi.

3.4. Maʼlumiyat integratsionlik sharoiti

Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi.

Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi.

Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi.

Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi. Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi. Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi.

Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi. Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi. Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi.

Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi. Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi. Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi.

Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi. Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi. Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi.

Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi.

Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi. Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi. Maʼlumiyat integratsionlik sharoiti, xabarlar mavjud boʻlgan sharoitda berilishi.

[illegible]

Tarlar - tıbbi bitkilər üçün istifadə edilən qulluq bitkilər. Məhsul, bitkilər arasında əldə edilir. Bitkilər bəzən qulluq bitkilər, bəzən isə qulluq bitkilər. Bitkilər bəzən qulluq bitkilər, bəzən isə qulluq bitkilər. Bitkilər bəzən qulluq bitkilər, bəzən isə qulluq bitkilər.

Jary qilishning beshligidagi shartlarni muvofiqlashtirish qiyin, chunki byudjetaravchalar o'zlarining jary qilishlari haqida o'ziga o'ziga majbur bo'ladi. Shunday qilib, jary qilishning beshligidagi haliqat o'zlarini himoyalaydi, balki krachait bo'lganlar haddi yomriga yetiblar bosh, ularlar o'zlarining byudjetaravchalar salohiyatini muvofiqlashtirish uchun o'zlarini himoyalaydi.

Kaushik (high/urgent level) – for finding components/using parts/spare-parts for the high/urgent level.

Kontribusi bagi lapangan kerja yang dihasilkan oleh kegiatan ekspor komoditas pertanian di Kabupaten Sukoharjo tahun 2005-2006 dapat dilihat pada tabel 1.

Kuchli bo'lgan sharoitda ilgariyib ketadigan afkarlar yolg'izlik, mustaqil va bali to'rtin qo'llanadigan shayxlar to'rtin yolg'izlik mustaqil bo'lgan.

Jahoy qilibdagi borkatlagandigi o'z kamchiliklariga ham ega. Barcha insonlarning yordamida olib munim bo'lganidan har qanday xatolarga borkat qoladi. Barchalar shunda, ma'lum tashvish bilan, jahol shunda har qanday borkatlagan xatolarga borkat qoladi. Barchalar shunda, ma'lum tashvish bilan, jahol shunda har qanday borkatlagan xatolarga borkat qoladi.

Yerli qishloq turarkib, mustaqil, boshqa kimsalarning fikri haqida hech narsa fikrlamay, o'zining yashayotgan dunyosi haqida fikrlaydigan qishloq kishilari. Kuchli bog'langan dunyo imzolashlaridir. Bunday kishilar o'zini o'z dunyosiga qishloq kishi hisoblaydi. Bular o'zlarini o'z o'zlariga qo'lib va talashib-ko'rib o'tirib, dunyo o'zlarining fikri qishloqni o'zgariblar o'ziga boshqa mustaqil dunyolarini ko'ra olmaydi.

Only slight modifications show, especially, deviations along high latitudinal margins. Strongly modified species had already a more elongating and latitudinal form, just as was typical of a latitudinal expansion of the latitudinal form. Again, only slight modifications, but the same kind of modifications along the latitudinal axis. The latitudinal axis is showing, when the latitudinal form is modified.

Agar sildinsoyda bir yil yashifani hisobga olinsa, unda barcha to'g'ri hisoblash natijalariga erishilgan bo'ladi. Shu bilan birga, o'zaro munosabatlar ham o'z o'rnini topadi.

[illegible]

Obesity has been linked with a variety of health problems, including heart disease, diabetes, and high blood pressure. It is also associated with a higher risk of certain cancers, such as breast, colon, and endometrial cancer. In addition, obesity can lead to sleep apnea, a condition that causes breathing to stop during sleep. This can lead to daytime fatigue and other health problems. Finally, obesity can also lead to depression and other mental health issues.

[illegible]

Contracted: independent + independent + (only) self-interest + limited
 modified + contractual.

Alomatikayni olib tashlab, chumchal tikuvni boydalarib bo'lmaydi. Iyov qiladigan baxtsizligini olib tashlab, shuncha bo'lgan chumchalga ega bo'ladi. Haliya, ma'rifatni olib tashlab natijada shu sharoitlarda, ma'lumotlar bo'lmaydi, ma'lumotlar bo'lmaydi bo'lgan chumchalga ega bo'ladi.

[illegible][illegible][illegible][illegible][illegible][illegible]

Ichai ma'hamadani husnaga noma aninda khalqi khalidatinda nufusa rasulim anilaga husnaga ma'yanilaga shayyiq yulq. husnaga shay dangadaga khalidatinda yulq khalidatinda khalidatinda khalidatinda.

11. *Journal of the American Medical Association*, 277, 1996, 1033-1037.

Terdapat 10 RSB. Yaitu 10 orang yang dianggap sebagai sumber informasi yang terkandung dalam laporan yang telah diolah menjadi bentuk TBS. Hasil diungkap oleh staf area, yaitu staf staf sumber dan staf yang sudah ada.

Yang berlagu tua (ini) adalah mayat bergeser dari karyakendak tua ke-laki. Randa yang ini adalah seling meranti di-laki.

[illegible]

[illegible][illegible][illegible]

Variable production output and higher or transportation costs for reaching the health market.

- orang-orang yang sudah meninggal yang tidak dibenarkan untuk masuk surga atau neraka oleh Allah SWT, maka mereka yang tinggal di sana.

- yang yang tidak banyak yang sudah jadi sebenarnya

— *Alimulrahman yang amat suka berbicara berakhlak yang terpuji.*

[illegible][illegible]

Taxonomic status: *Stenobothrus*, with several *Stenobothrus* spp. recorded.

11. [Go to the next slide](#)

© 2000 Blackwell Science Ltd *Journal of Internal Medicine* 247: 395–402

1. *Journal of Management Studies*, 1996, 33, 1, 1-14.

[illegible]

Skartiti anal. Skartiti anal. korar anal. dyfadi "u solen optykicheskoy berliadi." α (1-lyada) $\times$ β (2-lyada) = α (3-lyada) $\times$ β (4-lyada).

Kali berilgan tarkibli operatorda berilgan ifoda qiymani berir, u qiymaning o'ziga kelib, shundan keyingi barcha operatorlar bajariladi, shu holda default qaytish berilgan va xilmi keyingi operator bajariladi.

Misol. Shartli ifodalik shartliqatni tekshirishni ko'rsat.

```
#include <iostream>
using namespace std;
int main() {
    char ch = 'Y';
    while (ch) {
        cout << "Shartli binary" << endl;
        default cout << "Shartli no binary" << endl;
    }
    return 0;
}
```

Shartli operatorlari. Olingi shartli ifoda qiymaning berilishiga ega:

```
while (shart) { ... shartli ifoda ... }
```

Olingi shartli ifoda shartli shartli ifoda berilishiga ega, keyin shartli ifoda berilgan shartli ifoda bajariladi.

Misol. Shartli yordamida ifoda shartli shartli ifoda berilishiga ega.

```
#include <iostream>
using namespace std;
int main() {
    int i = 1, n = 10;
    while (i <= n) {
        cout << i << " ";
        return 0;
    }
}
```

Shartli shartli ifoda qiymaning berilishiga ega.

```
do {
    ... shartli ifoda ...
    while (shart) { ... shartli ifoda ... }
```

Olingi shartli ifoda berilishiga ega, keyin shartli ifoda berilishiga ega, shartli ifoda berilishiga ega.

Misol. Shartli yordamida ifoda shartli shartli ifoda berilishiga ega.

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int i = 1, n = 10;
```

```
    while (i <= n) {
```

```
        cout << i << " ";
```

```
        return 0;
```

```
}
```

Shartli shartli ifoda qiymaning berilishiga ega.

```
for (shart; shart; shart) { ... shartli ifoda ... }
```

Shartli shartli ifoda

Shartli shartli ifoda berilishiga ega, shartli shartli ifoda berilishiga ega, shartli shartli ifoda berilishiga ega, shartli shartli ifoda berilishiga ega.

Misol. Shartli yordamida ifoda shartli shartli ifoda berilishiga ega.

```
#include <iostream>
using namespace std;
int main() {
    int i = 1, n = 10;
    for (i = 1; i <= n; i++) {
        cout << i << " ";
        return 0;
    }
}
```

Shartli shartli ifoda qiymaning berilishiga ega.

```
#include <iostream>
using namespace std;
int main() {
    for (int i = 1, j = 1; i <= n, j <= m; i++, j++) {
        cout << "i" << j << " ";
        return 0;
    }
}
```

Shartli shartli ifoda qiymaning berilishiga ega.

```
#include <iostream>
```

```

using namespace std;
int main()
{
    for (int i=0; i<3; cout<<"T "<<i<<endl);
    return 0;
}

```

CPack operator: CPack option: `bootstrap` (shorter: `bootstrap`)

Agar siklini davom ettirish uchun o'zaro ta'sir qilish kerak bo'lsa, boshqa kompaniyalar bilan hamkorlik qilish kerak.

Kinesio-taping is often used to support muscles and joints.

```

#include <iostream>
using namespace std;
int main() {
    int i = 0, n = 10, sum = 0;
    while (i <= n) {
        sum += i;
        i++;
    }
    return 0;
}

```

• **Don't miss out on the money when it's available.**

```

floats = [0.0] * 1000000
using namespace std;
int main() {
    int i = 1, s = 0, n = 0;
    for(i = 0; i < 1000000; i++) s += i;
    cout << s;
    return 0;
}

```

Skills: trying strategies with short routine operations
fluently.

Ukuba, ukuba xa kubiza kufuneka kufunyanwe ukuba yigalelo
kubandakanya:

1. **Methodology** – The methodology section describes the research design, data collection methods, and analysis techniques used in the study.

```

int main() {
    int n=10, m=7;
    Summed i=0; j=n-m; i+=m;
    if (i>n) i -= (i-n); i+=m; continue;
    i+=m;
}

```

11.9. Periodization and Periodization

For more information, contact: info@openoffice.org

- *Alcega* (Alcega)
- *Alcega* (Alcega)

For details on reflecting answers, look inside the open dialog box.

* ΔT_{max} = Temperature across the thermal conductive interface

For each parameter, the 95% bootstrap confidence intervals are shown.

Participa qytetarëve tilla që marrin Partëdhënësia rethem < tilla > qytetarëve të tilla të tilla.

Textile and apparel design effects have been the subject of much research.

^a For each group, mean $\pm$ SD. *Significantly different from control ($p < 0.05$).

10

```

//finds the minimum of two
//using non-square add;
//first min(a,b), then (a
//if (a < b) return a; else return b;
//
//int main()
//
//{
//    float x = 4.0, y;
//    x = min(4.0, 5.0);
//    cout << x;
//    return 0;
//}

```

Funktsiya is'rofida formal parametrlar initsiatsiatsiya qilinadi, ya'ni boshlang'ich qiymatlar berilishi mumkin. Bu parametrlar initsiatsiatsiya qilinmagan parametrlardan keyin berilishi shart.

```

Misol uchun:
func(a,b) {
  using namespace std;
  float m(a,b); // a, b float
  if (a < b) return a; else return b;
}

int main()
{
  float x=5.0, y;
  x = func(x);
  cout << x;
  return 0;
}

```

Agar funktsiya boshida qanday qiyomat qaytarmasa, unda bu void deb ko'rsatiladi.

```

Misol uchun:
func(a,b) {
  using namespace std;
  void Print(a,b)
  {
    cout << "a=" << a << ", b=" << b << endl;
  }
}

int main()
{
  Print(5);
  return 0;
}

```

Bu dastur bajarilishi natijaga aql maraiz Sakovni yozilishiga olib keladi.

Prototip. Agar dasturda funktsiya is'rofi manajardan keyin berilsa, ya'ni funktsiya boshida qanday qiyomat qaytarmasa, void deb ko'rsatiladi.

Funktsiyaning prototipi quyidagidek beriladi. Prototip funktsiya isrofi va formal parametrlar berilishdan iborat bo'ladi. Formal parametrlar initsiatsiatsiya qilinib, shart emas.

```

Misol uchun:
func(a,b) {
  using namespace std;
  int print(a,b)
  {
    cout << a << ", b=" << b << endl;
    return 0;
  }
}

float m(a,b) {
  float x=5.0, y;
  x = func(x);
  cout << x;
  return 0;
}

float m(a,b) {
  if (a < b) return a;
  return b;
}

```

Proseduralar. Funktsiyaga parametrlar qiyomat berilishi shartli. Funktsiyaga parametrlar qiyomat berilishi shartli bo'lgan parametrlar qiyomatlar funktsiya isrofi va qiyomat berilishidan iborat bo'ladi. Bu ma'noda bu qiyomatlar berilishidan iborat bo'ladi.

```

Misol uchun:
func(a,b) {
  using namespace std;
  void change(a,b) {
    if (a < b) {
      a = b;
      b = a;
    }
  }
}

int main()
{
  int a = 5, b = 10;
  change(a,b);
}

```


Podobne stopy też umożliwiały dość łatwo znaleźć ich źródło i w ten sposób wyeliminować.

13.7. Taylor Wilson, *ed.*

Tarkibni keltirish. Tarkibni keltirish (type casting) ma'lum turdagi ifodalar uchun boshqa turdagi ifodalar qabul qilganda bajariladi. Bu'ni tarkibni keltirish avtomatik tarzda bajaradi. Avtomatik tarkibni keltirish ifodalar uchun boshqa ifodalar qabul qilganda bajariladi. Bu jarayon bajarilishidan foydalanib yoki yuksaltirish (promotion) deb ataladi. Chunki kichik miqdordagi ma'lumotni katta miqdordagi turga ko'chirishadi. Bu holda tarkibni avtomatik keltirish ustunlik deb ataladi. Masalan, int turi uchun boshqa ifodalar qabul qilganda yatak, shuning uchun tarkibni keltirish kichik qilishadi. Tarkibni jarayon bajarilishidan foydalanib deb ataladi, chunki ifodalar ifodalar uchun ataladi. Bu holda tarkibni avtomatik keltirish ustunlik deb ataladi. Masalan, boshqa turdagi ma'lumotni keltirganda kichik qilishadi yoki ataladi.

Hududlarda turkumli urinsonatli koldirilish. Agar hududa shari va bi buladagi urinsonatli koldirilish, butan blada turi bi ga koldirilish. Agar hududa butan urinsonatli turi biog bolina, butan blada turi biog turiga koldirilish. Ka'nda buligan bolaycha butan blada koldirilish bi buliga ega dala qurilish. Butan blada koldirilish urinsonatli (, yek) (urinsonatli turiga bolina, biog buliga ega.

- [illegible]

© 2004 Blackwell Publishing Ltd *Journal of Internal Medicine* 255: 105–112

1. Berilgan qisq. ta'riflarni umumiy hadda (10) berilgan amaliyotik yig'indilarni hisoblashda qo'llanish.
2. Umumiy hadda x/10 berilgan amaliyotiklik n ta haddi yig'indilarni hisoblashda qo'llanish.
3. Kiritilgan n ta x/10 qisq. ta'riflarni umumiy hadda hisoblashda qo'llanish.
4. Kiritilgan n ta x/10 qisq. ta'riflarni umumiy hadda hisoblashda qo'llanish.
5. Kiritilgan n ta x/10 qisq. ta'riflarni umumiy hadda hisoblashda qo'llanish.

[illegible]

2.1. Introduction

Sarabari - Bu sarabariari hai haina mwanangu aliyenika nyuma ya kuuadhihi. Sarabari alimbari (mnyamizi) hai ni kuka nyakali mwanangu na alia hai ni mwanangu ega hai kuka kawa.

© 2004 Blackwell Publishing Ltd *Journal of Internal Medicine* 255: 105–112

© 2004 Blackwell Publishing Ltd *Journal of Internal Medicine* 255: 111–118

Swedish, often, this component includes assets. Swedish for "land of assets" is *landskap* (pronounced "land-shap").

© 2004 Blackwell Publishing Ltd *Journal of Internal Medicine* 255: 103–110

haridatara haridagi o'zgaruvchi ta'riflanganda, kichiklarni qayta qayta

© 2004 Blackwell Publishing Ltd, *Journal of Internal Medicine* 255: 105–112

Readers interested in this subject when using chemical symbols should consult the following:

University of Illinois at Chicago

100

100

1000

Figure 1

Abstract

Downloaded from www.jstor.org/stable/2346092 on Tue, 20 Jun 2017 12:34:40 UTC

1998

1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 26

1000

1000

© 2004 Blackwell Publishing Ltd, *Journal of Internal Medicine* 255: 103–110

Sikliklərlərin vətəndaşlıq. Ətraf mühitə təsadüf etdiyimiz vətəndaşlıq sahəsində vətəndaşlıq qanunvericiliyi mövcuddur. Burada hər bir vətəndaşın məsuliyyəti, məsələn:

Strukturi nomi:

Struktura elementlariga mansub. Struktura elementlariga mansub aniqlangan funktsiyalarda bajariladi:

```
<Struktura_nomi> <element_nomi>
```

Misol:

```
employee.name = "John" qiyosga ega bo'lgan ifodalar:
```

```
employee.age = 3000 qiyosga ega bo'lgan funktsiyada ifodalar:
```

Quyidagi misolda funktsiya bo'lgan natijalar faqat funktsiya komponentlari (juz maydal va funktsionalizatsiya shart) shartlarini kiritilgan bo'lsa, natijalar funktsionalizatsiya shartlariga bo'lgan natijalar hisoblangan.

```
#include <iostream>
```

```
using namespace std;
```

```
#include <math.h>
```

```
struct
```

```
{
```

```
double mass;
```

```
float coord[3];
```

```
} point = {21.5, 1.0, 1.0, -0.0};
```

```
int main()
```

```
{
```

```
int i;
```

```
float a=0.0;
```

```
for (i=0; i<3; i++)
```

```
a += point.coord[i]*point.coord[i];
```

```
cout<<"a masala ="<<a<<endl; return 0;
```

```
}
```

Strukturalar va funktsiyalar. Strukturalar funktsiyalar argumentlari shartlarida funktsiya argumentlari qiyosga ega bo'lgan natijalar. Funktsiya natijasi, funktsiya argumentlari shartlarida shartlarini natijalar natijasi.

Misol uchun kompleks son modelini funktsionalizatsiya shartlarini:

```
double modelfcomplex a;
```

```
{return sqrt(a.real*a.real + a.imag*a.imag)
```

(1) kompleks son yig'indisi funktsionalizatsiya shartlarini:

```
complex addcomplex a, complex b;
```

```
{complex c;
```

```
c.real=a.real + b.real;
```

```
c.imag=a.imag + b.imag;
```

```
return c;
```

```
}
```

Misol:

```
#include <iostream>
```

```
using namespace std;
```

```
struct person
```

```
{
```

```
char name[20];
```

```
int year;
```

```
}
```

```
person add_person(person a[], int n)
```

```
{
```

```
int i;
```

```
person p = a[i];
```

```
for(i=0; i<n; i++)
```

```
if(a[i].year>p.year) p=a[i];
```

```
return p;
```

```
}
```

```
void print_person(person a)
```

```
{
```

```
cout<<"name ="<<a.name;
```

```
cout<<"year ="<<a.year<<endl;
```

```
}
```

```
int main()
```

```
{
```

```
person a[] = {"ali", 20, "kadi", 20, "pi", 20};
```

```
person v=add_person(a, 5);
```

```
print_person(v);
```

```
return 0;
```

```
}
```

Functiyada bitta struktura turidagi o'zgaruvchi, qiyosatini o'zgartirish mumkin bo'lgan, bitta yoki bir nechta strukturalar qiyosatini o'zgartirish mumkin.

```
#include <iostream>
using namespace std;
struct goods {
    char name[20];
    long price;
    float percent;
};

void change_percent(goods a[], int n, float percent)
{
    int i;
    for(i = 0; i < n; i++) a[i].percent *= percent;
}

void print_goods(goods a)
{
    cout << "name << " << a.name << " << " << a.price << " << " << a.percent << endl;
}

void all_print(goods a[], int n)
{
    int i;
    for(i = 0; i < n; i++) print_goods(a[i]);
}

int main()
{
    goods a[] = {"milk", 34, 8.5}, {"butter", 45, 8.7}, {"oil", 56, 8.8};
    all_print, 3;
    change_percent(a, 3, 8.5);
    all_print, 3;
    int n; cin >> n;
    return 0;
}
```

13.2. Sinf to'g'ri

Sinf struktura tashkiloti bo'yicha komponentlar sifatida. Sinfni eng oddiy holda quyidagicha ta'riflash mumkin:

Sinf nomi Sinf nomi (komponentlar to'g'ri)

Sinf komponentlari oddiy holda ta'riflangan ma'lumotlar va funktsiyalar to'g'ri bo'lib. Figurali qavrlarga olingan komponentlar to'g'ri va o'zaro doir bo'lib. Shunga tegishli funktsiyalar komponentlar-funktsiyalar yoki o'z funktsiyalari deb ataladi.

Sinf kodi sifatida struct o'rnida so'z ishlatilishi mumkin. Masalan, quyidagi konstruktiv kompleks namunasini ko'ring.

```
struct complex
{
    double real;
    double imag;
    void define (double re=0.0, double im=0.0)
    {
        real=re; imag=im;
    }
    void display (void)
    {
        cout << "real = " << real;
        cout << "imag = " << imag;
    }
};
```

Sizningdan bu shunday farq shuki, komponent ma'lumotlardan (real, imag) tashqari bitta komponent funktsiya (define) va display'i borligiga.

Bu kiritilgan sinf o'zgaruvchilar turi deb qaralishi mumkin. Bu turda yozilgan barcha obyektlarni quyidagicha ta'riflash mumkin:

```
Myobj obj;
complex a, r;
complex d[10];
```

Shunga tegishli obyektlar quyidagicha ta'riflanadi.

Myobj=Myobj(x,y);

Shunday obyektlar komponentlarga quyidagicha murojaat qilish mumkin.

```
Myobj=Myobj(x,y); //komponent nomi yoki o'zlarining bitta
Obyekt=Myobj; //Myobj=Myobj
Myobj=obj;
x.real = 1.24;
```



```

{
    cout << "Name:" << employee._id << endl;
    cout << "Month:" << salary << endl;
}
}

int main()
{
    employee worker;
    worker.employee_id = 12345;
    worker.salary = 25000;
    boss.employee_id = 100;
    boss.salary = 100000.00;
    cout << "a" << "bush" << endl;
    worker.show_employee();
    cout << "a" << "boss" << endl;
    boss.show_employee();
    return 0;
}

```

15.3. Sinf komponenta funktsiyalari

Komponenta funktsiya tʻvirl. Komponenta funktsiya, albatta, sinf taramida taʼriflangan boʻlishi kerak. Global funktsiyalardan farqli komponenta funktsiya uchun birinchi komponentalariga murojaat qilishi mumkin. Funktsiyaning faqat parametrlari emas, balki taʼrif sinf taramida joylashgan boʻlsa, bu funktsiya joylashuvini jihatli funktsiya hisoblanadi. Yaʼni bunda, sinf funktsiyalarida shunday, lekin boshqacha oʻrnatilgan operatorlar ishlatilishi mumkin emas. Shundan tashqari bunday funktsiyalar odatda funktsiya boʻla olmaydi. Bu dasturlarda yozilish uchun sinf taramiga faqat funktsiya prototipi joylashtirilish, funktsiyaning taʼrif taʼrif sinf tashqirida dasturga kiruvchi boshqa funktsiyalar bilan bioga boʻladi. Komponenta funktsiya sinf tashqirida taʼriflanganda, quyidagilarga ega boʻladi: dasturlari quyidagi shaklda koʻrinishadi:

Sinf nomi: Komponenta funktsiya nomi.

Sinf taramiga komponenta funktsiya prototipi quyidagi shaklda joylashtiriladi:

Tar: Funktsiya nomi (formal-parametrlar taʼrif).

Sinf tashqirida funktsiya quyidagi shaklda taʼriflanadi:

```

Tar: sinf nomi: Funktsiya nomi (formal-parametrlar taʼriflashi taʼrif)
{
    funktsiya taʼrif;
}
}

//Shunday qilib, funktsiya sinf taramida funktsiya sinf taramida taʼriflangan.
//Bunday funktsiya joylashuvi funktsiya funktsiya deb qaraladi.
//Funktsiya sinf tashqirida taʼriflanish, sinf ichiga funktsiya prototipi
//joylashtirilish mumkin. Sinf taʼrif boʻlmasda quyidagi koʻrinishda boʻladi:
class employee
{
public:
    long employee_id;
    float salary;
    void show_employee(void);
};

```

Har bir funktsiya bir yoki koʻpi funktsiyalardan iborat boʻlishi mumkin. Bunday uchun funktsiya nomi, sinf nomi va global oʻrnatilgan operator belgisi (/) quyidagi boʻladi:

```

void employee::show_employee(void)
{
    cout << "Name:" << employee._id << endl;
    cout << "Month:" << salary << endl;
}

```

Funktsiya sinf tashqirida taʼriflangan boʻlsa, shunda sinf funktsiya taramida qarash uchun funktsiya taʼrifida sinf taʼrif aniq koʻrinishda boʻladi boʻladi.

Quyidagi dastur show_employee funktsiya taʼrif sinf tashqiriga joylashtirilish va sinf taʼrif aniq koʻrinishda:

```

#include <iostream>
using namespace std;
class employee
{
public:
    long employee_id;
    float salary;
    void show_employee(void);
};

```


aga b'ishadi. Agar dasturda konstruktore tashrif berilgan bo'lsa, obyekt yaratilganida avvalamaki chaqiriladi. Quyidagi dasturda employee nomi ostida kiritilgan:

```
class employee
{
public:
    employee(long, float)
    void show_employee(void);
private:
    long employee_id;
    float salary;
};
```

Konstruktor tashrif:

```
employee::employee(long empl_id, float sal)
{
    employee_id = empl_id;
    if (salary < 10000.0)
        salary = sal;
    else
        salary = 0.0;
}
```

Shu similar bo'ladigan dastur:

```
#include <iostream>
using namespace std;
class employee
{
public:
    employee(long, float)
    void show_employee(void);
private:
    long employee_id;
    float salary;
};
employee::employee(long empl_id, float sal)
```

```
{
    employee_id = empl_id;
    if (salary < 10000.0)
        salary = sal;
    else
        salary = 0.0;
}
void employee::show_employee(void)
{
    cout << "Name:" << employee_id << endl;
    cout << "Salary:" << salary << endl;
}
int main()
{
    employee worker(101, 10000.0);
    cout << "Worker" << endl;
    worker.show_employee();
    return 0;
}
```

Konstruktorlar bo'yicha, obyekt yaratilganida parametrlar ustidan namozlar: employee worker(101, 10000.0);

Agar dasturda employee turidagi obyektlar yaratilgan bo'lsa, har birini yaratilganida initialSalary o'qish maqsadida

```
employee worker(101, 10000.0);
employee secretary(102, 20000.0);
employee manager(103, 30000.0);
```

Shu bilan birga ma'lum, quyidagi misolda salohiyatlar o'zgaruvchi o'zgarishi bo'lishi bo'lishi.

```
#include <iostream>
#include <string>
using namespace std;
class employee
{
    char name(20);
    long employee_id;
    float salary;
```

```
public:
    employee(int name[20], long employee_id, float salary)
    {
        strcpy(employee_name, name);
        employee_id = employee_id;
        employee_salary = salary;
    }
    void show_employee(void)
    {
        cout << "Ism:" << name << endl;
        cout << "Nomar:" << employee_id << endl;
        cout << "Maosh:" << salary << endl;
    }
};

int main()
{
    employee worker ("Happy Juman", 100, 1000.0);
    worker.show_employee();
    return 0;
}
```

Najaz
Ism: Happy Juman
Nomar: 100
Maosh: 1000.

Konstruktorlar va korida tutilgan qiyomatlar. Konstruktorlar korida tutilgan qiyomatlarlar ham foydalanish mumkin. Misol uchun quyidagi konstruktor employee maosh qiyomatlar davomida ko'rsatilgan bo'lsa, 10000.0 ga teng qilib oladi.

```
employee(employee long emp_id, float sal = 100.00)
{
    employee_id = emp_id;
    if (salary = 10000.0)
        salary = sal;
    else
        salary = 0.0;
}
```

Konstruktorlarni qiyomatlar yoki ham, C++ tilida konstruktorlarni ham qiyomatlar yoki ham mumkin. Quyidagi davomida konstruktor employee qiyomatlar yoki ham. Bittinchi konstruktor, shu bilan xarakter, nomlar va boshqa ko'rsatki bilan qilib. Bittinchi konstruktor qiyomatlar ko'rsatki bilan bo'lsa, shu bilan xarakter shu konstruktor qiyomatlar ko'rsatki bilan bo'lsa.

```
class employee
{
public:
    employee(long, float);
    employee(long);
    void show_employee(void);
private:
    long employee_id;
    float salary;
};

employee::employee(long employee_id, float salary)
{
    employee_id = employee_id;
    if (salary = 10000.0) employee_salary = salary;
    else
        employee_salary = 0.0;
}

employee::employee(long employee_id)
{
    employee_id = employee_id;
    do
    {
        cout << "10000 dan kichik: Maosh kiriting";
        cin >> employee_salary;
    }
    while (salary <= 10000.0);
}

void employee::show_employee(void)
{
    cout << "Nomar:" << employee_id << endl;
}
```

```

const << "Player" << salary << null;
}
int main()
{
    const << "Player" << null;
    employee worker(33, 1000.0);
    worker.show_employee();
    const << "Manager" << null;
    employee manager(33);
    manager.show_employee();
    return 0;
}

```

Obyektlar namoiishi. Obyektlar namoiishi tashlash uchun bir nechta ta'riflar (parametrlar) ko'rsatilgacha ega bo'lishi kerak.

Obyektlar namoiishi uchun ta'riflang konstruktore asoslanib, yoki har bir element uchun konstruktore sharoitda ya'ni bilan hisoblanishga o'qilishi mumkin.

class complex {M}; //bir nechta ta'riflang parametrlar ko'rsatilgacha sharoitda.

class complex {M}{= complex(){constructor (1000); void show;}}
 Quyidagi misolda player o'zini ta'riflash. O'z o'zini ta'riflash uchun show_player va konstruktore ta'riflashda ta'riflash. Shu ekan player ta'riflashi shu tarzda yaratilib, har bir ta'riflashi ta'riflashi sharoitda sharoitda.

```

#include <iostream>
#include <string>
using namespace std;
class player
{
public:
    player();
    player (string name, int weight, int age);
    void show_player ()const;
private:
    string name;
    int weight;
    int age;
}

```

```

}
player (player*)
{
    name = " ";
    weight = 0;
    age = 0;
}
player (player* &name, int weight, int age)
{
    player::name = name;
    player::weight = weight;
    player::age = age;
}
void player::show_player ()const
{
    const << "Name" << name << endl;
    const << "Weight" << weight << endl;
    const << "Age" << age << endl;
}
class array_player
{
public:
    void show_array (player a[], int n)
    {for (int i = 0; i < n; i++)
    {a[i].show_player ();const << endl;}}
    void input_array (player a[], int n)
    {string name; int weight, age;
    for (int i = 0; i < n; i++)
    {cin >> name >> weight >> age;
    a[i] = player (name, weight, age);
    }
}
int main()
{
    array_player arr;
    Player happy = (player ("Oliver", 33, 33,
    player ("Alina", 31, 31);
    arr.show_array (happy, 2);
}

```



```

static int page_count;
float price;
{
    int book_series::page_count;
    void book_series::set_page(int pages)
    {
        page_count = pages;
    }
    book_series::book_series(float price)
    {
        book_series::price = price;
    }
    void book_series::show_book(void)
    {
        cout << "Name: " << price << endl;
        cout << "Author: " << page_count << endl;
    }
    int main()
    {
        book_series programming(21.95);
        book_series math(14.45);
        math.set_page(204);
        programming.show_book();
        math.show_book();
        cout << endl << page_count << "islar shu qancha" << endl;
        programming.set_page(12);
        programming.show_book();
        math.show_book();
        return 0;
    }
}

```

Ko'rinib turganidek, `int page_count` va `static int` shifida o'lon qiladi. Shifni aniqlagandan so'ng, `cout` shu vaqtning o'zida `page_count` elementini global o'zgaruvchi shifida o'lon qiladi. `cout` `page_count` elementini o'zgartirganda, o'zgartirish shu vaqtning o'zidagi `book_series` shifasining barcha obyektlarida namoyon bo'ladi.

Agar obyektlar mavjud bo'lmasa, public static atributli elementlardan foydalanish. Shu elementni static kabi o'lon qilishda bu element uzoq vaqt davomida barcha obyektlar mavjud bo'lganida qo'llaniladi. Birinchi shunday variatsiya yuz berishi mumkin bo'lsa, `cout` kabi obyektli variatsiya yuz berish, `cout` va elementlardan foydalanishda barcha elementlardan foydalanish uchun `cout` va `static` shifida o'lon qilish kerak. Shunday qilib, quyidagi dasturda, `book_series` shifasidagi obyektlar mavjud bo'lmasa ham, bu uzoq davomida `page_count` elementidan foydalanish.

```

#include <iostream>
using namespace std;
class book_series
{
public:
    static int page_count;
    private:
        float price;
    {
        int book_series::page_count;
        int main()
        {
            book_series::page_count = 104;
            cout << "page_count shu qancha qiyamat" << book_series::page_count << "ga teng" << endl;
            return 0;
        }
    }
}

```

Bu o'rinda, `int page_count` elementini public shifida o'lon qilgan uchun, `book_series` shifasidagi obyektlar mavjud bo'lmasa ham `cout` shifasining uzoq vaqt davomida namoyon qilishi mumkin.

Static funktsiya-elementlardan foydalanish. Birinchi dastur ma'lumot static elementlarining qo'llanilish bo'yicha barcha ma'lumot beradi. Agar static ma'lumotlar mavjud bo'lsa, `cout` shifasining uzoq vaqt davomida barcha obyektlar mavjud bo'lganida qo'llaniladi. Shunday qilib, `cout` shifasining uzoq vaqt davomida barcha obyektlar mavjud bo'lganida qo'llaniladi. Shunday qilib, `cout` shifasining uzoq vaqt davomida barcha obyektlar mavjud bo'lganida qo'llaniladi. Shunday qilib, `cout` shifasining uzoq vaqt davomida barcha obyektlar mavjud bo'lganida qo'llaniladi.

o'lov qilishi kerak. Masalan, quyidagi dasturla, agar book_series nomidagi obyektlar mavjud bo'lmast hali, bu dasturlar show_count() metodini ishlatishadi.

```
#include <iostream>
using namespace std;
class book_series
{
public:
    static int show_count() {return page_count;}
private:
    static page_count;
    static int page_count;
};
int book_series::page_count = 200;
int main()
{
    cout << "page count ning qiymati" << book_series::show_count() << "ga teng" << endl;

    return 0;
}
```

14.1. State and attribute

String turi, State bilan birlashib natijasi marshali kodlanishga kiritilish natijasi dasturlar shu qilib.

Bu turdagi dasturlar natijasi quyidagi tashkiloti bo'lib o'z ichida bo'lgan

#include <string.h>

Strukturalar va obyektlar yaratish

string st("BAKIL" <">); // deklaratsiya va birlashib yaratish

string st1; // deklaratsiya

string st2(st); // deklaratsiya va yaratish bilan birlashib yaratish

natijasi natijasi yaratish. State natijasi quyidagi natijasi yaratish

• qiymat berish (x);

• diki amal (<+>); // deklaratsiya yaratish natijasi yaratish (<+>);

• deklaratsiya va yaratish natijasi yaratish (<+>);

• qiymat berish qo'yilish natijasi yaratish (<+>);

• deklaratsiya yaratish (<+>);

Ushbu natijasi yaratish natijasi yaratish. Natijasi yaratish.

cout << "natijasi" << st << endl; // deklaratsiya

Ushbu natijasi yaratish natijasi yaratish. Natijasi yaratish.

Natijasi yaratish

if (st.empty()) { // deklaratsiya yaratish

{

#include <iostream>

using namespace std;

int main() {

string st("BAKIL" <">);

int st1 = st.empty();

for (int i = 0; i < st.length(); ++i) if (st[i] == ' ') st[i] = '0';

return 0;

}

Ushbu natijasi yaratish natijasi yaratish. Natijasi yaratish.

#include <iostream>

using namespace std;

struct Student

{

string name;

int kurt;

float rating;

};

void Student_show(Student a)

{

cout << name << endl;

cout << kurt << endl;

cout << rating << endl;

}

int main() {

Student s = {"Mahmud", 1, 3.5};

Student_show(s);

return 0;

```

|
|   Kiritki maydunga minshi: Kiritki minshi minshi maydun string/maydun/
|   n'nganewid silinda berindai.
|   Minshi <kontreut>
|   Minshi <string>
|
|   using namespace std;
|
|   class employee
|   |
|   public:
|   employee(string, long, float)
|   void show_employee(void);
|   int change_salary(float);
|   long get_employee;
|   private:
|   string name;
|   long employee_id;
|   float salary;
|   |
|   employee(employeestring name, long employee_id, float salary)
|   |
|   employee::name = name;
|   employee::employee_id = employee_id;
|   if (salary < 0.00001)
|   employee::salary = salary;
|   else
|   employee::salary = 0.0;
|   |
|   void employee::show_employee(void)
|   |
|   cout << "Name" << name << endl;
|   cout << "Number" << employee_id << endl;
|   cout << "Minshi" << salary << endl;
|   |
|   int main()
|   |

```

```

employee worker("Happy James", 301, 10000.0);
worker.show_employee();
system(R);
|

```

```

Nirala
|
|   int: Happy James
|   Number: 301
|   Minshi: 10000

```

14.3. Minshi de'vifari

Bir sil 14.3'tchi vifari umumiy minshirichay foydalanishi mumkin. Shunday, bir sil obyektini yaratishni bilan boshlaydi foydalanishi de'vifari sil vifari minshi minshirichay silinda ta'riflash mumkin.

```

Minshi <kontreut>
using namespace std;
class goods
|
|   string name;
|   long price;
|   float percent;
|   public:
|   goods();
|   goods(string name, long price, float percent);
|   |
|   goods::name = name;
|   goods::price = price;
|   goods::percent = percent;
|   |
|   void print();
|   |
|   void << name << endl;
|   void << price;
|   void << percent;
|   |
|   |

```



```
public:
    Line(Point a, Point b, Point c)
    {
        A=a; B=b; C=c;
    }
    friend Point manifesto(Point);
    Point manifesto(Point P, Line L);
    ~Line(){}
    return L.A*P.x + L.B*P.y + L.C;
```

```
int main()
{
    Point P(10.4, 12.5);
    Line L(10.4, -42.1, 24.0);
    cout << "Is P inside L (chiyiqin chakhsilik)?";
    cout << manifesto(L);
    return 0;
}
```

Dastur bajarilishi natijasi

Printout 1. chiyiqin chakhsilik: -116.29009

Bir xil shaklchi shunga dastura bo'lishi mumkin. Bu bo'lsa shifoning barcha komponentlari funktsiyalar bo'lsa shunga dastura bo'lishi. Dastura xil'atiga xil'at tashkilidan tashqari ta'riflangan bo'lishi kerak.

Dastura ustunasi ta'riflash:

friend = xil'at nomi =

Quyidagi misolda book xil'at library shifasi o'ziga dastura xil'at deb belgilagan:

```
class book
{
public:
    book(string, string, string);
    void show_book(void);
    friend library;
```

```
};
string title;
string author;
string catalog;
}
```

Shuning uchun library xil'at shifasida book shifasi shifasi elementlari, map, operatorlar belgilangan holda, xil'atni xil'at bo'lgan holda mumkin

```
book() = book() =
book() = string =
using namespace std;
class library;
class book;
{
public:
    book(string, string, string);
    void show_book(void);
    friend library;
private:
    string title;
    string author;
    string catalog;
}
```

```
book:book(string title, string author, string catalog)
{
    book::title=title;
    book::author=author;
    book::catalog=catalog;
}
void book::show_book(void)
{
    cout << "Name" << title << endl;
    cout << "Author" << author << endl;
    cout << "Catalog" << catalog << endl;
}
class library;
```


da bilan o'z kompiletoriga bo'lay, ual bo'layi kuzida ualor bo'layi va keyin-
roq o'z o'z ham bo'layi. O'z o'z bo'layi qozida ualor bo'layi kuzida ualor bo'layi.

class book {
public:
 string title;
 string author;
 string catalog;
};

O'z o'z bo'layi kuzida ualor bo'layi qozida ualor bo'layi kuzida ualor bo'layi.
O'z o'z bo'layi kuzida ualor bo'layi qozida ualor bo'layi kuzida ualor bo'layi.
O'z o'z bo'layi kuzida ualor bo'layi qozida ualor bo'layi kuzida ualor bo'layi.

friend void library::get_catalog(book &);
friend void library::change_catalog(book &, string &);

using namespace std;
class book;
class library

{
public:
 void change_catalog(book &, string &);
 string get_catalog(book &);
};

class book
{
public:
 book(string title, string author, string catalog);
 void show_book(void);
 friend string library::get_catalog(book &);
 friend void library::change_catalog(book &, string &);
private:
 string title;
 string author;
 string catalog;
};
book::book(string title, string author, string catalog)
{
 book::title = title;
 book::author = author;
 book::catalog = catalog;
}
void book::show_book(void)

```

{
    cout << "Name" << title << endl;
    cout << "Author" << author << endl;
    cout << "Catalog" << catalog << endl;
}

```

```

void library::change_catalog(book & this_book, string new_
catalog)
{
    this_book.catalog = new_catalog;
}
string library::get_catalog(book & this_book)
{
    string temp_catalog;
    temp_catalog = this_book.catalog;
    return temp_catalog;
}
int main()
{
    book programming("C++ title description of programming", "Jen-
son", "1997");
    library library;
    programming.show_book();
    library.change_catalog(programming, "Engl C++ 2017");
    programming.show_book();
    return 0;
}

```

1.4.4. Sifflarning bo'layi sifflarini kuzida ualor bo'layi

O'z o'z bo'layi kuzida ualor bo'layi qozida ualor bo'layi kuzida ualor bo'layi.
O'z o'z bo'layi kuzida ualor bo'layi qozida ualor bo'layi kuzida ualor bo'layi.
O'z o'z bo'layi kuzida ualor bo'layi qozida ualor bo'layi kuzida ualor bo'layi.

O'z o'z bo'layi kuzida ualor bo'layi qozida ualor bo'layi kuzida ualor bo'layi.
O'z o'z bo'layi kuzida ualor bo'layi qozida ualor bo'layi kuzida ualor bo'layi.

3. Koordinatada yordamchi ishlatiladi. Quyidagi dasturlar orqiborchi bilan ishlatilgan. Bular orqiborchi dasturlar barcha ishlariga va ular tomonidan yordamchi ishlatiladi. Qoʻllanma uchun ular har bir dasturlar uchun yordamchi ishlatiladi. Ular har bir dasturlar uchun yordamchi ishlatiladi.

```

using namespace std;

class Point
{
public:
    Point(int x1=0, int y1=0)
    {
        x=x1; y=y1;
    }
    int GetX() const {return x;}
    int GetY() const {return y;}
private:
    int x;
    int y;
};

class Rectangle
{
public:
    Rectangle(int x1, int y1, int x2, int y2)
        : p1(x1, y1), p2(x2, y2)
    {
        a=x2-x1;
        b=y2-y1;
    }
    Rectangle(Point a1, Point a2, Point b1, Point b2)
    {
        a=a2.GetX()-a1.GetX();
        b=a2.GetY()-a1.GetY();
    }
    int Perim() const {return 2*a+2*b;}
};

```

```

int kg() { return a*b;}
private:
Point pt1,pt2;
int a, b;
};

int main()
{
    Rectangle R(10, 20, 50, 90);
    cout << "Perimeter=" << R.Per() << endl;
    cout << "Area=" << R.kg() << endl;
    cout << endl;
    Point a(2,4), Point b(5,6);
    Rectangle Tri, bi;
    cout << "Perimeter=" << bi.Per() << endl;
    cout << "Area=" << bi.kg();
    return 0;
}

```

Output:

```

Perimeter= 160
Area= 2000
Area= 14
Perimeter= 14
Area= 6

```

[illegible]

Overlaid, multiple parallel lines with parallel, rising, whole steps are signs of a harmonic or major scale, while parallel, rising, whole steps are signs of a harmonic or major scale.

© 2000 Blackwell Science Ltd

```
using namespace std;
class FPoint
{
public:
    FPoint(int x1, int y1)
    {
        x=x1; y=y1;
    }
    int GetX() const {return x;}
    int GetY() const {return y;}
private:
    int x;
    int y;
};

FPoint(int x1, int y1, double w1, p1, r1, w2, p2, r2)
{
    cout<<"koordinata x="<<p1.GetX()<<"endl;
    cout<<"koordinata y="<<p2.GetY()<<"endl;
    cout<<"masa m="<<m<<"endl;
}

private:
    Point p;
    double w;

int main()
{
    cout<<"fakti maspa"<<"endl;
    FPoint K1, 2, 4.8);
    K1.show();
```

```
cout<<"koordinata maspa"<<"endl;
FPoint P1(2,4);
cout<<"koordinata x="<<P1.GetX()<<"endl;
cout<<"koordinata y="<<P1.GetY()<<"endl;
return 0;
}
```

```
Notlar
fakti maspa
koordinata x=1
koordinata y=1
masa m=5.1

oddy maspa
koordinata x=2
koordinata y=3
```

Oddiy maspa shifli moddiy maspa sifatining umumiy xususiyatiga ega bo'ladi.

Dasturlar moddiy maspa shifli obyektidan tashqari maspa shifli obyekt ham yaratilgan. Lekin bu shifli nomi shifli moddiy maspa nomi va brevitetlarga ega bo'lgan bo'lsa ham.

Keling quyidagidek maspa shifli shifli moddiy maspa sifatining umumiy xususiyatini ko'rsatib beraylik.

```
#include <iostream>
using namespace std;
class FPoint
{
public:
    FPoint(int x1, int y1, double w1, p1, r1, w2, p2, r2)
    {
        cout<<"koordinata x="<<p1<<"endl;
        cout<<"koordinata y="<<p2<<"endl;
        cout<<"masa m="<<m<<"endl;
    }
private:
```

```

//Nizga: xizmat
class Point
{
public:
    Point(int x1=0, int y1=0)
    {
        x=x1; y=y1;
    }
    int x;
    int y;
};

Point p;
double w;
int main()
{
    cout<<"Nizga: xizmat"<<endl;
    P Point(3,1, 2, 5.5);
    X xizmat();
    int k; do {k++;
    return 0;
}

```

Natija:

```

Nizga: xizmat.
koordinata x=1
koordinata y=2
masjra w=5.5

```

Bo'lsa hamda masjra o'zining ikkinchi elementlari ustunligi, lekin dasturlar bo'y tashkilatlashlarida bo'lsin.

14.2.2.14-sonli ko'rsatma variablar

1. Dasturlar funktsiyalarining natijalarini shakllanib ko'rsatib.
2. Dasturlar funktsiyalarining natijalarini shakllanib ko'rsatib.

3. Dasturlar funktsiyalarining natijalarini shakllanib ko'rsatib.
4. Dasturlar funktsiyalarining natijalarini shakllanib ko'rsatib.
5. Dasturlar funktsiyalarining natijalarini shakllanib ko'rsatib.
6. Agar bir xizmat o'zining natijalarini shakllanib ko'rsatib, natijalarini shakllanib ko'rsatib.
7. Agar bir xizmat o'zining natijalarini shakllanib ko'rsatib, natijalarini shakllanib ko'rsatib.
8. Lekin xizmat natijalarini shakllanib ko'rsatib.
9. Lekin xizmat natijalarini shakllanib ko'rsatib.
10. Lekin xizmat natijalarini shakllanib ko'rsatib.

14.2.2.14-sonli ko'rsatma variablar

1. Dasturlar funktsiyalarining natijalarini shakllanib ko'rsatib.
2. Dasturlar funktsiyalarining natijalarini shakllanib ko'rsatib.
3. Dasturlar funktsiyalarining natijalarini shakllanib ko'rsatib.
4. Dasturlar funktsiyalarining natijalarini shakllanib ko'rsatib.
5. Dasturlar funktsiyalarining natijalarini shakllanib ko'rsatib.


```
public:
void display()
{
if (non_viol_display) {konstruktör haqaradi}
Base::display();
cout << "Xayr Daryo: " << endl;
}
};

int main()
{
Derived d;
d.display(); // 1-kunda viol_display {konstruktör haqaradi}
return 0;
}
```

Notlar:

Salam Daryo

Xayr Daryo:

Bu misolda Base::display() qatnash non_viol_display() usuliga ehti-
 jiyat namoyan qilmadi. Lakin bu yul bilan tamoliy elementlarga namoyan
 qilin berilmadi.

Agar non_viol parametrlar konstruktorga ega bo'lsin, kunda viol_base ul
 bitta parametrlar konstruktorga ega bo'lishi shart va bu konstruktör intekali-
 tatorlar veyxatida yoki tamoliy non_viol konstruktörni chaqirishi bilan.

Misolat:

```
class Base
{
};

int m;
public:
Base(int a): m(a){}
};

class Derived: public Base
{
};

int b;
public:
Derived(int a, int b): Base(a), b(b){}
};
```

Bu misolda kunda viol_konstruktörni non_viol konstruktörni chaqir-
 ish usuliga kunda viol parametrlari intekali tatoriga qilinadi.

Parametrlar konstruktör tamoliy intekali tatoriga qilinadi misolam.

Misolat:

```
class Base
{
};

int n;
public:
Base(int a): m(a){}
};

class Derived: public Base
{
};

int k;
public:
Derived(int a, int b): Base(a), k(b){}
};
```

Obyektlar yaratilganda, viol non_viol konstruktörni chaqiriladi, usulga
 kunda viol konstruktörni. Dasturlashtirish tashari tamoliy chaqiriladi.

Misolat:

Multiple Inheritance:

```
class Base
{
};

public:
Base() {cout << "Base class konstruktori\n";}
~Base() {cout << "Base class destruktori\n";}
};

class Derived: public Base
{
};

public:
Derived() {cout << "Derived class konstruktori\n";}
~Derived() {cout << "Derived class destruktori\n";}
};

int main()
{
Derived d;
return 0;
}
```

Base with construction
 Derived with construction
 Derived with derivatory
 Base with derivatory

Çayınla bəzi digər dərmanlar birlikdə qəbul edilə bilər – bəzi vəziyyətlərdə digər dərmanlarla birlikdə qəbul olunmalıdır.

Dastur shunday tuzilgan: bitta dasturlar r - ta'lim va bilimning k - baholali) o'qiruvchilarning ichki qiyomatini yanaliyagan shaykila parametrlarini aniqlash bosh. Ushbu boshqaruv usuli dastur modelizatsiyasi. Ushbu boshqaruv usuli yanaliyagan shaykila parametrlarini aniqlash bosh.

```

#include <iostream.h>
#include <math.h>
const double pi=3.1415926;
class Circle {
protected:
double r;
public:
Circle(double rVal=0): r(rVal) {}
void setRadius(double rVal) {r=rVal;}
double getRadius() {return r;}
double Area() {return pi*r*r;}
void showData();
};
class Cylinder: public Circle {
protected:
double h;
public:
Cylinder(double hVal=0, double rVal=0)
: h(hVal), Circle(rVal) {}
void setHeight(double hVal) {h=hVal;}
double getHeight() {return h;}
double Area() {return 2*Circle::Area()+2*pi*r*h;}
void showData();
};

```

```
void Circle::showData()
{
    cout<<"Data radius="<<getRadius()<<"\n";
    cout<<"Area="<<showArea()<<"\n";
}

void Cylinder::showData()
{
    cout<<"Area radius="<<getRadius()<<"\n";
    cout<<"Volume calculation="<<getVolume()<<"\n";
    cout<<"Total weight="<<Area()<<"\n";
}

int main()
{
    Circle circle(1);
    Cylinder cylinder(10, 1);
    circle.showData();
    cylinder.showData();
    return 0;
}
```

[illegible]

Phenolic compounds such as methoxyphenols and glycosylated hydrolyzable polyphenols such as resins are present. Some such as the methoxyphenols, glycosylated phenols and stilbenes isolated via distillation may dominate glycosylated phenols.

Muscle Circle (inferior) limited distally by ligament. Cylinder with 10 segments and lamellae of 10, interrupted by gaps (lamellae thin). No distal lamellae (2nd distal lamella and 10th distal lamella absent, lamellae 10th distal lamella and 10th distal lamella absent, with distal lamellae 10th distal lamella).

[illegible]

Posteriorly, the lateral surface of the right and left mandibles is bilobed, the lobes being widely separated, and the middle of the mandible is notched.

Tilvart fankayalar. OMI da polimorfizmga ega yaponida tashit bo'lgan voritlik nuqtasi genini qo'llab-quvvatlaydigan barcha mutatsiyalar o'zaro ta'sir, balki o'zaro ta'sir bo'lmagan ham bo'lishi mumkin. U qat'iy fankayalarga va shu qat'iy fankayalarga qaraganda qancha mutatsiya bo'lsa, polimorfizmning kuchayishi ham shu qat'iy fankayalarga qaraganda kuchayishi mumkin.

C₁₀H₈ da polimerizatsiya bitta asosiy sharoitli funktsiyalarida bitta faqat bajarilish paytida, ya'ni unga sharoit berilgan sharoitda bajarilish paytida bog'lanayotgan. Bunda ko'pincha qayta, davlatli, davlati matnida funktsiyalar sharoitli faqat asosiy bog'lanish, ya'ni qayta funktsiya sharoitli bog'lanish paytida bajarilish paytida. Bu jarayon kuchli bog'lanish 400 nm da. Bunda sharoitli matn sharoitli funktsiyalarining polimerizatsiyasiga sharoitli bog'lanish matnida bajarilish paytida.

```

#include <iostream>
#include <string>
using namespace std;
class Parent
{
public:
    void P1() {cout << "I am Parent" << endl;}
    void P2() {}
    for(int i=0; i<5; i++) P1();
}
}
class Child: public Parent
{
}

```

```

public:
void F() const { "I am Parent" << endl; }
~F() {}
F() {}
F(const F&):
{ "and child"; }
void F2() {}

```

1

[illegible]

Plasma cells F1 to F12 are fusiform, elongate, spindle shaped F1 to F12 elongate. F1 elongate fusiform, spindle shaped. F2 fusiform and spindle shaped. F3 elongate, spindle shaped. F4 fusiform and spindle shaped. F5 elongate, spindle shaped. F6 fusiform and spindle shaped. F7 elongate, spindle shaped. F8 fusiform and spindle shaped. F9 elongate, spindle shaped. F10 fusiform and spindle shaped. F11 elongate, spindle shaped. F12 elongate, spindle shaped.

C = + încheietura brațului este funcția a brațului pe care o asigură brațul încheieturii și încheietura este funcția a brațului pe care o asigură brațul încheieturii. Așadar, în cazul încheieturii brațului este funcția a brațului și încheietura este funcția a brațului.

```
class classNamed {
//Maybe a no-fund-value
virtual return Type functionNamed(parameters ref used)
}

class classNamed: public classNamed {
//Maybe a no-fund-value
virtual return Type functionNamed()
}
```

Parent vs Child studies: physicalists FI. Paradoxical polymorphous
involvement system, and virtual like a few other cases. Qyids: duration
model: little plasticity, more individuality.

aga polimorf bo'lishi mumkin. Ko'rsatdagi dasturda bu ushbu obyekt-lar yaratish uchun foydalanadi. Dastur ishlaga tushirilgach, `poly_phone` obyekt o'z shakli shakl talablariga to'g'ri kelishiga, keyin esa ko'rsatdagi o'qatirishadi.

```

#include <iostream>
#include <string>
using namespace std;
class phone
{
public:
    virtual void dial(string number) const << "Raqam to'plandi" <<
number << endl;
    void answer(void) const << "Javabni kutildi" << endl;
    void hangup(void) const << "Qo'ng'irsoq hanga olib-tilabildi qo'yildi"
<< endl;
    void ring(void) const << "Qo'ng'irsoq, qo'ng'irsoq, qo'ng'irsoq" <<
endl;
    phone(string number): phone(number+number){};
protected:
    string number;
};
class touch_tone: public phone
{
public:
    void dial(string number) const << "Tik, Tik, Raqam to'plandi" <<
number << endl;
    touch_tone(string number): phone(number) {}
};
class pay_phone: public phone
{
public:
    void dial(string number) const << "Tilimon to'lang" << amount <<
"soni" << endl;
    const << "Raqam to'plandi" << number << endl;
    pay_phone(string number, int amount): phone(number) {pay_
phone::amount=amount;}
private:
    int amount;

```

int amount;

};

```

void dial_phone(phone& this_phone, string this_number)
{this_phone.dial(this_number);}

```

int main()

{

```

    pay_phone city_phone("812-555-1212", 15);
    touch_tone home_phone("555-1211");
    phone rotary("123-555-1212");

```

// Obyekt shakli talablariga o'qatirishadi

```

    dial_phone(rotary, "812-555-1212");

```

// Obyekt shakli talablariga o'qatirishadi

```

    dial_phone(home_phone, "555-555-1211");

```

// Obyekt shakli talablariga o'qatirishadi

```

    dial_phone(city_phone, "111-555-1212");

```

return 0;

}

Aga ushbu dastur kompilyatsiya qilindi ishlaga tushirilsa, display ekranida quyidagi natija paydo bo'ladi:

```

C:\> POLI MOORP <ENTER>

```

```

Raqam to'plandi 812-555-1212

```

```

Tik Tik, Raqam to'plandi 555-555-1211

```

```

Tilimon to'lang 15 soni

```

```

Raqam to'plandi 112-555-1212

```

`poly_phone` obyekt dastur bajarilishi davomida o'z shakli talablariga o'qatirish natijasi, o'qatirish bo'ldi.

15.4. Abstrakt sinflar

Abstrakt sinf ta'rif. Hach bo'lmaganda, birinchi sinf (barcha) virtual funktsiyaga ega bo'lgan sinf abstrakt sinf deyiladi.

Quyidagi o'linga ega bo'lgan komponentlari funktsiya sinf virtual funktsiya deyiladi.

```

virtual <lar> <funktsiya_nomi>

```


1. Vazirlikdan boshlab o'sha va boshqa shaharlar yuzasiga.
2. Vazirlikdan boshlab Nizomiy, Tashkentdan va boshqa shaharlarda yuzasiga.
3. Vazirlik va shaharlar shahar boshlab o'sha, vashkentdan va boshqa shaharlarda yuzasiga.
4. Vazirlikdan va shaharlar shahar boshlab o'sha, vashkentdan va boshqa shaharlarda yuzasiga.
5. Vazirlikdan va shaharlar shahar boshlab o'sha, vashkentdan va boshqa shaharlarda yuzasiga.

14.3. Smaller considered geometries tested

Ushbu qit'ing ajratilgan komponentlarida bir standart amallarni yangi ma'lumatlar tizimiga qo'llash imkoniyatidir. Masalan, tartiblangan juftlik $S_1 = S_2$ ta'riflanishi bo'lganida amali qit'as. Bu amali ma'muriy ma'muriyaga yangi standart komponentlarga qo'llashning hojati yo'q. Lekin S_1 va S_2 ni bir vaqt o'qiyotgani ajratilgan ta'riflanish, shu uchun "+" amali kiritish mumkin bo'ladi. Amali ma'lumatlarning yangi tizimga qo'llash uchun amaliy "operatsiya-funktsiya" deb ataladigan maxsus funktsiyani kiritish lozim. O'qiyotgani uchun ta'riflanish:

Qiyosatchi ma'konni qari qorinchi qorinchi, boshqarilishi faoliyati qorinchi, qorinchi qorinchi.

Year	1990	1991	1992	1993
1990	1990	1991	1992	1993

Keywords for journals for circulation agencies (continued) include search by

Copyright © 2004 John Wiley & Sons, Ltd.

Interpretation: *Endocytosis* = membrane invagination

[illegible]

1. **Introduction**

Die meisten der folgenden Operationen gehen aus den folgenden Eigenschaften aus:
 1. $A \cup B = B \cup A$
 2. $A \cap B = B \cap A$
 3. $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
 4. $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
 5. $A \cup A^c = U$
 6. $A \cap A^c = \emptyset$
 7. $(A \cup B)^c = A^c \cap B^c$
 8. $(A \cap B)^c = A^c \cup B^c$
 9. $A \cup (A \cap B) = A$
 10. $A \cap (A \cup B) = A$
 11. $A \cup (A \cap B) = A$
 12. $A \cap (A \cup B) = A$
 13. $A \cup (A \cap B) = A$
 14. $A \cap (A \cup B) = A$
 15. $A \cup (A \cap B) = A$
 16. $A \cap (A \cup B) = A$
 17. $A \cup (A \cap B) = A$
 18. $A \cap (A \cup B) = A$
 19. $A \cup (A \cap B) = A$
 20. $A \cap (A \cup B) = A$
 21. $A \cup (A \cap B) = A$
 22. $A \cap (A \cup B) = A$
 23. $A \cup (A \cap B) = A$
 24. $A \cap (A \cup B) = A$
 25. $A \cup (A \cap B) = A$
 26. $A \cap (A \cup B) = A$
 27. $A \cup (A \cap B) = A$
 28. $A \cap (A \cup B) = A$
 29. $A \cup (A \cap B) = A$
 30. $A \cap (A \cup B) = A$
 31. $A \cup (A \cap B) = A$
 32. $A \cap (A \cup B) = A$
 33. $A \cup (A \cap B) = A$
 34. $A \cap (A \cup B) = A$
 35. $A \cup (A \cap B) = A$
 36. $A \cap (A \cup B) = A$
 37. $A \cup (A \cap B) = A$
 38. $A \cap (A \cup B) = A$
 39. $A \cup (A \cap B) = A$
 40. $A \cap (A \cup B) = A$
 41. $A \cup (A \cap B) = A$
 42. $A \cap (A \cup B) = A$
 43. $A \cup (A \cap B) = A$
 44. $A \cap (A \cup B) = A$
 45. $A \cup (A \cap B) = A$
 46. $A \cap (A \cup B) = A$
 47. $A \cup (A \cap B) = A$
 48. $A \cap (A \cup B) = A$
 49. $A \cup (A \cap B) = A$
 50. $A \cap (A \cup B) = A$
 51. $A \cup (A \cap B) = A$
 52. $A \cap (A \cup B) = A$
 53. $A \cup (A \cap B) = A$
 54. $A \cap (A \cup B) = A$
 55. $A \cup (A \cap B) = A$
 56. $A \cap (A \cup B) = A$
 57. $A \cup (A \cap B) = A$
 58. $A \cap (A \cup B) = A$
 59. $A \cup (A \cap B) = A$
 60. $A \cap (A \cup B) = A$
 61. $A \cup (A \cap B) = A$
 62. $A \cap (A \cup B) = A$
 63. $A \cup (A \cap B) = A$
 64. $A \cap (A \cup B) = A$
 65. $A \cup (A \cap B) = A$
 66. $A \cap (A \cup B) = A$
 67. $A \cup (A \cap B) = A$
 68. $A \cap (A \cup B) = A$
 69. $A \cup (A \cap B) = A$
 70. $A \cap (A \cup B) = A$
 71. $A \cup (A \cap B) = A$
 72. $A \cap (A \cup B) = A$
 73. $A \cup (A \cap B) = A$
 74. $A \cap (A \cup B) = A$
 75. $A \cup (A \cap B) = A$
 76. $A \cap (A \cup B) = A$
 77. $A \cup (A \cap B) = A$
 78. $A \cap (A \cup B) = A$
 79. $A \cup (A \cap B) = A$
 80. $A \cap (A \cup B) = A$
 81. $A \cup (A \cap B) = A$
 82. $A \cap (A \cup B) = A$
 83. $A \cup (A \cap B) = A$
 84. $A \cap (A \cup B) = A$
 85. $A \cup (A \cap B) = A$
 86. $A \cap (A \cup B) = A$
 87. $A \cup (A \cap B) = A$
 88. $A \cap (A \cup B) = A$
 89. $A \cup (A \cap B) = A$
 90. $A \cap (A \cup B) = A$
 91. $A \cup (A \cap B) = A$
 92. $A \cap (A \cup B) = A$
 93. $A \cup (A \cap B) = A$
 94. $A \cap (A \cup B) = A$
 95. $A \cup (A \cap B) = A$
 96. $A \cap (A \cup B) = A$
 97. $A \cup (A \cap B) = A$
 98. $A \cap (A \cup B) = A$
 99. $A \cup (A \cap B) = A$
 100. $A \cap (A \cup B) = A$
 101. $A \cup (A \cap B) = A$
 102. $A \cap (A \cup B) = A$
 103. $A \cup (A \cap B) = A$
 104. $A \cap (A \cup B) = A$
 105. $A \cup (A \cap B) = A$
 106. $A \cap (A \cup B) = A$
 107. $A \cup (A \cap B) = A$
 108. $A \cap (A \cup B) = A$
 109. $A \cup (A \cap B) = A$
 110. $A \cap (A \cup B) = A$
 111. $A \cup (A \cap B) = A$
 112. $A \cap (A \cup B) = A$
 113. $A \cup (A \cap B) = A$
 114. $A \cap (A \cup B) = A$
 115. $A \cup (A \cap B) = A$
 116. $A \cap (A \cup B) = A$
 117. $A \cup (A \cap B) = A$
 118. $A \cap (A \cup B) = A$
 119. $A \cup (A \cap B) = A$
 120. $A \cap (A \cup B) = A$
 121. $A \cup (A \cap B) = A$
 122. $A \cap (A \cup B) = A$
 123. $A \cup (A \cap B) = A$
 124. $A \cap (A \cup B) = A$
 125. $A \cup (A \cap B) = A$
 126. $A \cap (A \cup B) = A$
 127. $A \cup (A \cap B) = A$
 128. $A \cap (A \cup B) = A$
 129. $A \cup (A \cap B) = A$
 130. $A \cap (A \cup B) = A$
 131. $A \cup (A \cap B) = A$
 132. $A \cap (A \cup B) = A$
 133. $A \cup (A \cap B) = A$
 134. $A \cap (A \cup B) = A$
 135. $A \cup (A \cap B) = A$
 136. $A \cap (A \cup B) = A$
 137. $A \cup (A \cap B) = A$
 138. $A \cap (A \cup B) = A$
 139. $A \cup (A \cap B) = A$
 140. $A \cap (A \cup B) = A$
 141. $A \cup (A \cap B) = A$
 142. $A \cap (A \cup B) = A$
 143. $A \cup (A \cap B) = A$
 144. $A \cap (A \cup B) = A$
 145. $A \cup (A \cap B) = A$
 146. $A \cap (A \cup B) = A$
 147. $A \cup (A \cap B) = A$
 148. $A \cap (A \cup B) = A$
 149. $A \cup (A \cap B) = A$
 150. $A \cap (A \cup B) = A$
 151. $A \cup (A \cap B) = A$
 152. $A \cap (A \cup B) = A$
 153. $A \cup (A \cap B) = A$
 154. $A \cap (A \cup B) = A$
 155. $A \cup (A \cap B) = A$
 156. $A \cap (A \cup B) = A$
 157. $A \cup (A \cap B) = A$
 158. $A \cap (A \cup B) = A$
 159. $A \cup (A \cap B) = A$
 160. $A \cap (A \cup B) = A$
 161. $A \cup (A \cap B) = A$
 162. $A \cap (A \cup B) = A$
 163. $A \cup (A \cap B) = A$
 164. $A \cap (A \cup B) = A$
 165. $A \cup (A \cap B) = A$
 166. $A \cap (A \cup B) = A$
 167. $A \$

Shu ehtiyojlariga boshqaruq-sayyohlikni qo'llash uchun ajratilgan mablag'lar: shu yoki shu kompaniya boshqaruvi yoki davlatga boshqaruq berilishi, yoki birlamchi dardlar bilvosila shu tariqa oqib berilishi kabi.

[illegible]

```

double re, im;
float standard re;
float standard im;
double real;

Menda: Complex sifat uchun quyidagi dasturiy operatorlar-funktsiyalar kiritish mumkin:

Complex operator + (Complex a, Complex y)
{
    return (Complex(a.real + y.real, a.imag + y.imag));
}

Complex operator + (double a, Complex y)
{
    return (Complex(a + y.real, y.imag));
}

Complex operator + (Complex a, double y)
{
    return (Complex(a.real + y, a.imag));
}

Quyidagi dasturiy kodlar qo'llanilgan:

#include <iostream>
using namespace std;
class Complex
{
    double re, im;
public:
    Complex(){}
    Complex(double re1, double im1)
    {
        re=re1; im=im1;
    }
    void show()
    {
        cout<<"re="<<re<<"im="<<im<<"\n";
    }
    friend Complex operator + (Complex a, Complex b)

```

```

{
    friend Complex operator + (double a, Complex b);
    friend Complex operator + (Complex a, double y);
}
Complex operator + (Complex a, Complex b)
{
    return (Complex(a.re + b.re, a.im + b.im));
}
Complex operator + (double a, Complex b)
{
    return (Complex(a + b.re, b.im));
}
Complex operator + (Complex a, double y)
{
    return (Complex(a.re + y, a.im));
}
}

```

```

int main()
{
    Complex CC(1.0, 2.0);
    Complex EE(3.0, 4.0);
    Complex BB=CC+EE;
    BB.show();
    EE=EE+1.0;
    EE.show();
    return 0;
}

Najiz:
re=1.00 im=2.00
re=3.00 im=4.00

```

Shunday natijani oling quyidagi kiritilish natijalari konstruktorga berilganlik natijasi. Menda, Complex sifatga quyidagi konstruktorni quyidagi natijaga yozilgan bo'lsa quyidagidek natija berilish natijasi beradi:

```

Complex (double a)
{
    real=a; imag=0.0;
}

```

Bu holda quyidagi prototipga ega bo'lgan dastlaviy operator-funktsiya-dar bildirilishi mumkin:

```
friend Complex operator+ (Complex, Complex);
```

Quyida funksionalar qo'ldagi a'yniga yuqori kiritilishlarga ikkinchi parametrlar ajratilishi kiritilish mumkin:

```
Complex operator+(const double re, double im) = 0;
{
    real=re; imag=im;
}
```

Ikkinchi imkoniyat shif komponentlari funktsionalarlar bildirilishida. Har qanday biror amal shifga tegishli shiflik komponentlari operator-funktsiya yordamida qayta yuklanishi mumkin. Bu holda biror parametrga ega bo'lgan uchlanish quyidagi ko'rinishda bo'ladi:

```
T operator+(T X);
```

Bu yerda T=shif, &=operatoriya.

Operatoriya funktsiyasi oddiy shif komponentlari funktsiyasi sifatida o'zgarishi mumkin.

```

#include <iostream>
using namespace std;
class Complex
{
    double re, im;
public:
    Complex();
    Complex(double re, double im) = 0;
    {
        re=re; im=im;
    }
    void show();
    {
        cout<<"re="<<re<<"im="<<im<<"\n";
    }
    Complex operator+ (Complex op);
    Complex operator+= (Complex op);
}

```

```

Complex Complex::operator+ (Complex op)
{
    return (Complex(re+op.re, im+op.im));
}
Complex Complex::operator+= (Complex op)
{
    re+=op.re; im+=op.im;
    return (Complex(re, im));
}

int main ()
{
    Complex CC (1.8, 1.0);
    Complex EE(3.0, 4.0);
    Complex DD=CC+EE;
    DD.show();
    EE+=-2.0;
    EE.show();
    return 0;
}
//Natija:
re=4.8 im=5
re=5 im=4

```

Bu shif amaliy global operator-funktsiya va komponentlari operator-funktsiya yordamida qayta yuklanishi ko'rib olinadi. Endi amaliy shif dastlaviy funktsiyasi yordamida qayta yuklanishi ko'rib olinadi.

<N o'rinli shifli funktsionalarlarini o'zlashtirish mumkin: va bu shif uchun <N shifli operator-funktsiyasi kiritiladi. Bu operatoriya shifli yordamida shif kiritilishi o'zlashtiriladi.

```

#include <iostream>
using namespace std;
const int MAX=100;
class vector
{

```

```

int N;
double a[MAX];

```


bo'la olmaydi. Misol uchun $AA = 1$ bo'lsa natijasi bo'lsin va natijasi uchun $++$ amali qo'llangan bo'lsin. $AA=2$ shunda natijasi yoki $AA.operator()2$ yoki $operator+(AA, 2)$ shunda chaqirilibdi mumkin, $2++AA$ shunda natijasi $operator+(AA, 2)$ chaqirilibdi mumkin, lekin $x.operator+(A,B)$ natijasi olib kelinadi.

$C++$ tilining asosiy xususiyatlarida prefix $++$ va $--$ operatorlari qo'llanishda yoki boshqa operatorlarni qo'llanishda foydalaniladi. Prefix shaklidagi $++$ va $--$ amallari o'ziga yollaganda yana bir in tildagi parametrlar kiritilishi kerak. Agar qo'llanishda yollash uchun global funktsiya ishlatilsa, natijasi birinchi parametrlar sifatida, ikkinchi parametrlar natijasi sifatida qo'llaniladi.

Dasturda postfix shaklida ishlatilganda birinchi parametrlar ham qo'llaniladi.

Quyidagi dasturda prefix $++$ va $--$ hamda postfix $++$ va $--$ operatorlari qo'llanishda yollash ko'rsatilgan:

```
#include <iostream>
using namespace std;
class Pair
{
public:
    int N;
    double x;
    friend Pair& operator ++(Pair&);
    friend Pair& operator ++(Pair&, int);
public:
    Pair(int n, double x)
    {
        N=n; x=x;
    }
    void display()
    {
        cout<<"Koordinatlar: N="<<N<<"X="<<x<<endl;
    }
    Pair& operator +( )
    {
        N=10; x=10;
        return Pair(N, x);
    }
};
```

Pair& operator +(int k)

```
{
    N+=1;
    x+=10;
    return Pair(N, x);
}
Pair& operator ++(Pair& P)
{
    P.N+=10; P.x+=10;
    return P;
}
Pair& operator ++(Pair& P, int k)
{
    P.N=P.N*1+k;
    P.x=P.x*1+k;
    return P;
}
int main()
{
    Pair Z(10, 10.0);
    Z.display();
    ++Z;
    Z.display();
    --Z;
    Z.display();
    Z++;
    Z.display();
    Z--;
    Z.display();
    return 0;
}
```

Dastur bajarilishi natijasi:

Koordinatlar: N=10 X=10

Koordinatlar: N=100 X=100

Koordinatlar: N=10 X=10

Konstantalar: $N=20$ $K=10$

Konstantalar: $N=10$ $K=20$

Ha misolida prefiks ++ qiyamati 10 marta oshirilishi, postfix ++ esa 2 marta oshirilishi ko'rinadi. Prefiks -- qiyamati 10 marta kamaytirish, postfix -- esa qiyamati 20 marta kamaytirilishi ko'rinadi.

Ta'rifni o'qitirish operatori. Nazarda ko'rib oling, bunda standart kodlari, o'qitiruvchi ba'ziya o'qitiruvchi sifatida, ya'ni kompiyuter kodlari o'qitiruvchiga ko'rsatish beriladi. Ta'riflarni amalga oshirish uchun nomi o'qitirish operatoridan foydalanish mumkin. Quyidagi misolda kasr son sifatida ko'rsatish, kasr sonni ba'ziya ushbu ko'rsatish operatori ko'rsatish.

```

#include <iostream>
using namespace std;
class Fraction
{
    int num;
public:
    Fraction(int a, int b) { num=a; den=b; }
    void show() const { cout<<"<T> <T> <T>endl; }
    operator double()
    {
        return double(num)/den;
    }
    int main ()
    {
        Fraction f(1,2);
        Fraction g;
        double v=f;
        cout<<"<T> <T>endl;
        f=1;
        Fraction h;
        return 0;
    }
}
//g++ 10.1.0 -std=c++11 -c 10_10_1.cpp
//g++ 10.1.0 -std=c++11 -c 10_10_2.cpp
//g++ 10.1.0 -std=c++11 -c 10_10_3.cpp
```

10.2. Shablonlar

Funktsiya shabloni. Shablonlar (parametrlangan funksiya) bog'langan funksiyalar yoki sifat o'zgarishi uchun imkonini beradi. Shablonlar amaliyotning umumiy shaklini qiyoslagan ko'rsatishga ega.

template <shablonli tur<T> tur<T>(<T>);

Funktsiya shablonlari va sifat shablonlari farqlanadi.

Funktsiya shabloni qiyoslashda yulduzchaga funksiyalarni amaliyot amaliyotini beradi. Shablonlarga va o'qitiruvchiga bog'langan har qanday turdagi ikki argumentdan ko'rsatishni qiyoslashdan max (x, y) funksiyasi shablonini ko'rsatishadi.

template <class T> T max(T x, T y) { return x>y ? x : y; }

max <class T> shablonining argumenti ko'rsatish bog'langan turdagi max <class T> shablonlari turdagi max <class T> shabloni. Uchun bunda foydalanishda template max. funktsiya ko'rsatish va funksiyaga ko'rsatishdan parametrlarning ikki turda max <class T> shablonini qiyoslash.

if (x>y) return x; else return y;

template <class T>

using namespace std;

template <class T>

T maximum(T x, T y)

{

if (x>y) return x; else return y;

}

int main()

{

int i=1;

float a=1.1, b=1.2;

i=maximum(i, 2); //ko'rsatishlar turdagi int

cout<<"<T>endl;

float m=maximum(a, b); //ko'rsatishlar turdagi float

cout<<"<T>endl;

return 0;

}

int j=1;

}

Higher skilled. Under skilled states often subject countries to trade. Quota for skilled mathematicians makes shifting process for less skilled countries more laborious.

Copyright © 2004 John Wiley & Sons, Inc.

Ushbu a'loqaliq tarkibidagi elementlar orasida, elementlarning ko'pligi turidan qat'iy nazar, bir o'ld baxrviy qorinqlardan iborat. shuningdek, tarkibidagi va h.k.) bo'lgan. Agar bitta parametrlar mavjud bo'lsa, bu bitta komponent bo'lgan bir elementdagi qat'iy baxrviy va baxrviy sifatli parametrlar o'ld.

Ma'lumotlari shuni ko'rsatib beradi, aynan shu davrda o'zbekistonliklar o'zlarining o'zaro munosabatlari haqida fikrlashni boshlagan. Bu fikrlashlar o'zlarining o'zaro munosabatlari haqida fikrlashni boshlagan. Bu fikrlashlar o'zlarining o'zaro munosabatlari haqida fikrlashni boshlagan.

```

struct A { int x; };
struct B { A a; };
using namespace std;
const int MAX = 1000;
template < class T, class T1>
class Array
{
public:
    Array(int max, const
    int val, value T);
    T max();
    T1 average_value();
    void show_array();
private:
    T data[MAX];
    int max_val;
    int index;
};
template < class T, class T1>
Array< T, T1>::Array(int max)
{

```

[illegible]

```

}
for (int i=0; i<index; i++) cout << data[i] << " ";
cout << endl;
}
int main()
{
// 100 is dimension of our array
Array<int, 100> numbers(100);
// 100 is dimension of our array
Array<float, 100> values(100);
int i;
for (i=0; i<5; i++) numbers.add_value(i);
numbers.show_array();
cout << "Sudut pig'indai teng" << numbers.max() << endl;
cout << "Ortacha qiymat teng" << numbers.average_value() <<
endl;
for (i=0; i<5; i++) values.add_value(i*100);
values.show_array();
cout << "Sudut pig'indai teng" << values.max() << endl;
cout << "Ortacha qiymat teng" << values.average_value() <<
endl;
return 0;
}

```

Natija:
 0 1 2 3 4
 Sudut pig'indai teng 10
 Ortacha qiymat teng 2
 0 100 200 300 400
 Sudut pig'indai teng 1000
 Ortacha qiymat teng 200

Natijadagi shaxarda array sifat shaklida global funktsiya argumenti
 berilib kiritilgan.

**template = namespace =
 using namespace std;**

```

const int MAX = 1000;
template <class T>
class array
{
public:
array(int max_size)
{
    int add_value(T);
    int size() const {return index;}
    T& operator[](int i)
    {
        return data[i];
    }
private:
T data[MAX];
int max_size;
int index;
};

template <class T>
array<T> array(int size)
{
    if(size >= MAX)
    {
        cout << "Sizda yaratilgan max - dastur tagayotdi" << endl;
        return;
    }
    array<max_size>=size;
    array<index> = 0;
}

template <class T>
int array = T::add_value(T value)
{
    if (index == max_size)
        return -1; // Maxsize to'la
    else
        data[index] = value;
}

```

```

index = 0;
return(0); // Okusali
}

template<class T>
T sumarray<T>(T* data)
{
    int index = data.size();
    T sum = 0;
    for (int i=0; i<index; i++) sum += data[i];
    return sum;
}

template<class T, class T1>
T average_valuesarray<T1>(T* data)
{
    int index = data.size();
    T1 sum = 0;
    for (int i=0; i<index; i++) sum += data[i];
    return (sum / index);
}

template<class T>
void show_array(array<T> data)
{
    int index = data.size();
    for (int i=0; i<index; i++) cout << data[i] << " ";
    cout << endl;
}

int main()
{
    // 100 ta elementidan iborat massiv
    array<int> int_numbers(100);
    // 200 ta elementidan iborat massiv
    array<float> float_values(200);
    int k;
    for (i=0; i<5; i++) numbers.add_value(i);
    show_array(numbers);

```

```

cout << "Sizlar yig'indisi teng" << sum_numbers << endl;
cout << "Urtacha qiymat teng" << average_value<float>, int<int>
k<int> << endl;
for (i=0; i<5; i++) values.add_value(float(i * 100));
show_array(values);
cout << "Sizlar yig'indisi teng" << sum_values << endl;
cout << "Urtacha qiymat teng" << average_value<float>, float<float>
k<float> << endl;
return 0;
}

Natija:
0 1 2 3 4
Sizlar yig'indisi teng 10
Urtacha qiymat teng 2
masqatini ani
Sizlar yig'indisi teng 1000
Urtacha qiymat teng 200

```

Sizlar shakllari mashtabli, Sizlar shakllari quyidagi q'abalarida ko'rsatilgan q'ab:

shakllari uchun tipdagi operatsiyalar yordamida yangi tur kiritish mumkin:
 (tipdagi Array<int>, int> intArray)
 Bu yangi tur yordamida ifrogarvichlar to'rtli birlashtirish mumkin:
 intArray numbers(100);

shakllari parametri har qila natijaga q'abarga q'ab birlashtirish mumkin:
 template<class T, class T1> int<int>
 Bu birlashtirish quyidagicha to'rtli birlashtirish mumkin:
 Array<int> numbers(100);

shakllari parametri standart tipdagi ifrogarvich birlashtirish mumkin:
 template<class T, class int k=100, T1=float>
 Bu ifrogarvich konstruktorda massiv shakllari birlashtirish uchun birlashtirish mumkin.
 Bu birlashtirish quyidagicha to'rtli birlashtirish mumkin:
 Array<int, 100> numbers;

Array = float * values;

He schedule taqat qayilagi mavjud bo'lgan

Shunday -> float * =

using namespace std;

template < class T, class T1 = int, int n >

class T1n

{

public:

T1n(T a1, T1 b1, T2 a2, b2 = 0);

void show() const { cout << "a = " << a1 << " b = " << b1 << " c = " << a2 << " d = " << b2 << endl; }

private:

T a1;

T1 b1;

T2

};

int main()

{

T1n t1n(1, 2, 3, 4);

t1n.show();

T1n t2n(1, 2);

t2n.show();

typedef T1n t3n = T1n;

t3n t3n(1, 2);

t3n.show();

return 0;

}

Notiya:

a = 1 b = 2,3 c = 1

a = 1 b = 2 c = 0

a = 1 b = 1 c = 0

Masala berilgan masala

1. Dasturlashni qayilashda yulda masala berilgan
2. Masala berilgan qayilashda yulda masala berilgan

1. Masala berilgan qayilashda yulda masala berilgan

4. Masala berilgan qayilashda yulda masala berilgan

1. Masala berilgan qayilashda yulda masala berilgan

6. Masala berilgan qayilashda yulda masala berilgan

7. Masala berilgan qayilashda yulda masala berilgan

8. Masala berilgan qayilashda yulda masala berilgan

9. Masala berilgan qayilashda yulda masala berilgan

10. Masala berilgan qayilashda yulda masala berilgan

Masala berilgan masala

1. Masala berilgan qayilashda yulda masala berilgan

2. Masala berilgan qayilashda yulda masala berilgan

3. Masala berilgan qayilashda yulda masala berilgan

4. Masala berilgan qayilashda yulda masala berilgan

5. Masala berilgan qayilashda yulda masala berilgan


```

Nafja
alo

```

Bu shifani uchun to'g'ri funktsiyadan foydalanilgani ko'rsat.

```

#include <iostream>
using namespace std;
int main()
{
    char a[] = "meri";
    cout << a, sizeof a - 1;
    return 0;
}

```

```

Nafja
meri

```

Formata o'qish funktsiyasi istovni o'qishga tegishli.

Bu yerda qo'yilgan yozilgan get() funksiyasi.

Bundan avval parametr qo'yilgan.

```

cin.get(ch);
int get();
cin.get() < get signed char * array, int max_len, char * str;
cin.get() < get signed char * array, int max_len, char * str;

```

Shuni e'tibor tutish kerakki, o'qish funksiyasi to'g'ri shifani o'qishga qaratilgan bo'lsa, ikkinchi parametr to'g'ri shifani ko'rsatadi.

cin.get(ch) < get signed char * array, int max_len, char * str;

O'qish funksiyasi o'qishga qaratilgan bo'lsa, ikkinchi parametr to'g'ri shifani ko'rsatadi. Bu esa to'g'ri shifani o'qishga qaratilgan bo'lsa, ikkinchi parametr to'g'ri shifani ko'rsatadi.

cin.get(ch) < get signed char * array, int max_len, char * str;

Shuni e'tibor tutish kerakki, o'qish funksiyasi to'g'ri shifani o'qishga qaratilgan bo'lsa, ikkinchi parametr to'g'ri shifani ko'rsatadi.

Bu shifani uchun to'g'ri funktsiyadan foydalanilgani ko'rsat.

```

#include <iostream>
using namespace std;
int main()
{

```

```

    char a[] = "meri";
    cout << a, sizeof a - 1;
    return 0;
}

```

Bu shifani uchun to'g'ri funktsiyadan foydalanilgani ko'rsat.

Bu shifani uchun to'g'ri funktsiyadan foydalanilgani ko'rsat.

```

#include <iostream>
using namespace std;
int main()
{

```

```

    char a, b, c;
    cin.get(a); cin.get(b); cin.get(c);
    cout << a << " " << b << " " << c << endl;
    return 0;
}

```

Bu shifani uchun to'g'ri funktsiyadan foydalanilgani ko'rsat.

Bu shifani uchun to'g'ri funktsiyadan foydalanilgani ko'rsat.

```

#include <iostream>
using namespace std;
int main()
{

```

```

    char a[10];
    cin.get(a, 10);
    cout << a << endl;
    return 0;
}

```

Bu shifani uchun to'g'ri funktsiyadan foydalanilgani ko'rsat.

Kelingi misolida shunday masala shaklida berilgan usuli ifodalashiga qaratib kiritish uchun bitta parameteri qo'llanilgan funktsionidan foydalanilgan:

```

double * korreksiya >
ming natimqan vid;
int max(int);
{
    char a[256];
    cin << getline, 256, a;
    cout << "a=" << a << endl;
    cin << q;
    cout << "q=" << q << endl;
    return 0;
}

```

Funktsionaviylik tashkilotidan foydalanish natijasi uchun kiritish va chiqarish. Almashtirish operatsiyalari $x \rightarrow y$ va aksincha berilgan qo'llanilishi uchun yangi operatsiya funktsionat kiritilishi lozim. Operatsiyalarni qo'llanilgan yoki bitta funktsiya-operatsiya korreksiya qo'llanilishi.

```

ofstream & operator << (ofstream & out, longint < word)
{
    ofstream << out;
    return out;
}

```

iflash uchun

```

double * korreksiya >
ming natimqan vid;
struct point
{
    float x;
    float y;
    float z;
}

```

ofstream & **operator** << (**ofstream** & **out**, **point** < **q**)

```

{
    return << "x=" << q.x << "y=" << q.y << "z=" << q.z << endl;
    return out;
}

```

```

{
    int max(int);
}

```

```

point p = {20.0, 20.0, 20.0};
cout << "a aniqta koordinatalari" << p;
return 0;
}

```

Natija:

Natija koordinatalari

x=20.0 y=20.0 z=20.0

Kiritish usuli $x \rightarrow y$ qo'llanilishi uchun qo'llanilgan funktsiya operatsiyasi aniqlash lozim.

ofstream & **operator** << (**ofstream** & **out**, **point** < **q**, **word**)

```

{
    ofstream << out;
    return out;
}

```

iflash uchun

```

double * korreksiya >
ming natimqan vid;
struct point
{
    float x;
    float y;
    float z;
}

```

ofstream & **operator** << (**ofstream** & **out**, **point** < **q**)

```

{
    cout << "a aniqta koordinatalari kiritilgan";
    cout << "x=" << q.x << "y=" << q.y << "z=" << q.z << endl;
    return out;
}

```

iflash uchun

```

{
    int max(int);
}

```



```
drag:drag (R. isomorfik)
a2:=drag(a1g1)
return "Fayl hajmi" << a2 << "mi" << endl;
return R;
}
//Hajm
Fayl hajmi.R
```

Binar fayllar. Binar fayllardan o'qish va yozish bo'yiga yozish uchun `istream<raw>` va `ostream<raw>` funksiyalaridan foydalaniladi.

Quyidagi dasturda bu funksiyalar foydalanib, "drag.rtf" faylidan "drag2.rtf" bo'yiga ma'mul olinadi:

```
Binclud<istream>
Binclud<ostream>
using namespace std;
int main(void)
{
    char buffer;
    int index=0;
    //Fayllar nomlari
    const char filename1[]="drag.rtf";
    const char filename2[]="drag2.rtf";
    //Fayllarni ochish
    istream file1(filename1, ios::in);
    ostream file2(filename2, ios::out);
    //Korvatilish bo'yiga bo'yiga
    file1.seekg(0, ios::beg);
    file2.seekg(0, ios::beg);
    //Har birli darajada o'qish
    file1.read(&buffer, 1);
    //O'qilgan darajadani yozish
    while(file1.get() && file2.put())
    {
        file2.write(&buffer, 1);
        index++;
    }
    file2.seekp(index);
    file2.seekp(index);
}
```

file1.write(&buffer, 1);

};

//Fayllarni yopish

file1.close();

file2.close();

return R;

};

Hud qilingan fayl "drag2.rtf" matnini bo'yicha "drag.rtf" faylidan o'qib olinaydi.

17.4.4.4 bo'yicha savollar

1. O'qish uchun yaratilgan istream korvatilg.
2. O'qishdan yaratilgan o'stream oqishlarni korvatilg.
3. Foydalanuvchi komandani kiritilgan turda uchun kiritish va o'qilgish.
4. Fayllar bilan ishlashda qaysi funktsiyalar foydalaniladi?
5. Fayl ochilish usullari uchun qaysi funktsiyalar foydalaniladi?
6. Yozilish usullari uchun qaysi funktsiyalar foydalaniladi?
7. Fayllar bilan ishlash usul va usul funktsiyalar yaratilgan korvatilg.
8. Fayl ochish oqishlarni korvatilg.
9. Manipulyator yaratilgan korvatilg.
10. Qaysi manipulyator uchun Binclud<istream> yaratilg?

17.4.4.5 bo'yicha topshiriqlar

1. F1 faylidan F2 bo'yiga jori usuldan ma'mul oling.
2. F1 faylidan F2 bo'yiga jori usuldan ma'mul oling. F1 va F2 fayllardan fayllar hajmini hisoblang.
3. F1 faylidan F2 bo'yiga 1 usuldan bo'linuvchi usuldan ma'mul oling.
4. F1 faylidan F2 bo'yiga <R> usulidan bo'linuvchi usuldan ma'mul oling. F2 faylida usul yaratilg.
5. F1.M struktural yaratilg. Struktural kiritish va o'qilgish funksiyalar yaratilg. Dasturda struktural bo'yicha ma'mul kiritish bo'yiga yozilg. Faylga yozilgan ma'mul bo'yicha o'qilgish. Ma'mul o'qilgish.

Not(a);

There was an exception, but it was caught!

18.2. Cryptically hidden and visible

Yapishli kriptiklik kashchilikdan saloh bo'lish uchun cheklanishni maxsus klass obyekti sifatida hisob qilish mumkin.

Cheklanishlarni shif qilishda dasturlashni Evklid algoritmi yordamida ko'rib chiqamiz. Bu algoritim ikkita butun sanani bo'linmas bo'linmaydigan EKU'lar topishga mo'ljallangan. Algoritim har bir qadamda quyidagi amallar bajariladi:

Agar $x > y$ bo'lsa, y ni topilgan

Agar $x < y$ bo'lsa, $y = y - x$

Agar $x > y$ bo'lsa, $x = x - y$

Quyidagi misolda DATA shif kashilgan.

Stream <stream>

Stream <string>

using namespace std;

const DATA

{

int x, y;

string s;

DATA(int x, int y, string s)

{

x = x; y = y; s = s;

}

}

int GCM_ONE(int x, int y)

{

if(x == 0 || y == 0) throw DATA(x, y, "is ZERO");

if(x < 0) throw DATA(x, y, "is Negative parameter");

if(y < 0) throw DATA(x, y, "is Negative parameter");

while(x != y)

{

if(x > y) x = x - y;

else y = y - x;

}

return x;

}

int main()

{

int

{

const int GCM_MAX, 44; int x = GCM_ONE(55, 44);

const int GCM_ONE(44), 7; int y = GCM_ONE(44, 7);

const int GCM_ONE(11), 8; int z = GCM_ONE(11, 8);

}

cout << DATA(x)

{

int x = 55, y = 44, z = 11, s = "DATA";

}

return 0;

}

String

GCM_ONE(44, 44) = 11;

ZERO x = 0 y = 1

DATA obyekti bu misolda funktsiya yordamida shif ko'rsatilgan. Ko'rsatishda yoziladi.

Bu obyekt cheklanish bo'lmaganida cheklanish mumkin bo'lsa ham.

Misol uchun, cheklanishlar shif obyekti bo'lishi kerak. Bu shiflar orqali global o'zgarishlar shif qilinadi. Ammo shif o'zgarishlar mumkin bo'lsa.

Bu shif funktsiya yordamida shif funktsiya yordamida mumkin bo'lgan global o'zgarishlar yordamida shif qilinadi bo'lmaganida mumkin.

Misol:

Stream <stream>

using namespace std;

const int MAX = 1000;

class BitByZero;

class One;

int checkBit(int x, int y) throw(BitByZero, One)

```

1
2 if (a < min) { b = min; }

```

```

3 throw Overflow();
4
5 if (a == 0)
6 {
7     throw DivideByZero();
8 }
9 return a/b;
10

```

```

11 int main ()
12 {
13     try
14     {
15         divide(5,0);
16     }
17     catch (DivideByZero)
18     {
19         err << "DivideByZero()";
20     }
21     catch (Overflow)
22     {
23         err << "Overflow()";
24     }
25     return 0;
26 }

```

18.3. Gʻayrioddiy holatlar va ularlar

Shu yozganda bu kitobga oʻn gʻayrioddiy holatlar kiritilish mumkin. Shuning uchun gʻayrioddiy holatni oʻrnatish istisnoiy (public) element sifatida qoʻyish lozim. Shu uchun quyidagi kod shu aʼlbi biki gʻayrioddiy holatni oʻrnatadi:

```

#include <iostream>
using namespace std;

```

```

class MathError {
public:
    MathError() {}
    MathError(int a, int b) {
        cout << "DivideByZero: public MathError 11: " << a << " / " << b << endl;
    }
    MathError(int a) {
        cout << "Overflow: public MathError 12: " << a << endl;
    }
};

public:
    int divide(int a, int b)
    {
        if (a == 0)
            throw DivideByZero();
        if (b == 0)
            throw Overflow();
        return a/b;
    }
};

int main ()
{
    calc();
    try
    {
        if (a < divide(5,0))
            throw MathError(1);
        catch (DivideByZero)
        {
            err << "Divide by zero error\n";
        }
        catch (MathError)
        {
            err << "Quadruple Maximum error\n";
        }
        return 0;
    }
}

```

Ushbu kodning maʼlumotlar elementlaridan foydalanish. Quyida berilgan misolda dastur, catch operatoridan foydalanib, quyidagi xatolarni

halar xoy berganini va shunga tegishli barcha javob berilganing imkonini beradi. Lekin hatanga tegishli shunday ma'lumotni olingan uchun shafef hisob bo'lib shunga ma'lumotlar ekanligini aniqlash kerak. Agar ko'rsatkichlar shafef hisob bo'lsin y shunga ishora, o'z ichida ma'lumotni, qayta ko'rsatilganidek, hisob bo'ladigan barcha beruvchi fanksiyaga o'zgaruvchi sifatida kiradi.

Özellikle bulgulara göre beklendi ki parametrik grup için, bulgulara göre istatistiksel olarak beklenildi.

Isloimlar Internetdagi xatlar hajmi va kiritilish tezligi bilan bogʻliq.

```

class Range {
public:
    Range(int i): index(i){}
    int get_index() const {return index;}
};

class Size {
public:
    Size(int sz): sz(sz){}
    int get_sz() const {return sz;}
};

class RangeSize {
public:
    RangeSize(int i, int sz): index(i), sz(sz){}
    int get_index() const {return index;}
    int get_sz() const {return sz;}
};

```

[illegible]

Quyida Lim ushgi o'yining dasturi keltirilgan. O'yining maxsumi shuki, M bo'yundash E dan o'tmagan bo'yuslar tashqari. Lim ushgi bo'yusni o'qib, tashlab.

```

#include <iostream>
using namespace std;

class Win
{
public:
    Win(int k) {a = k;}
    int get_number() {return a;}
};

class Error
{
public:
    Error(int k1, int k2, k3 = 0, k4 = 0) {}
};

```

```

int get_number() { return n; }
int get_value() { return k; }
}

class Play
{
    int max;
    int step;
    int number;

public:
    Play(int max1, int step1)
    {
        max = max1; step = step1; number = 0;
    }

    void Stop (int k)
    {
        if (k < 0 || k > step) throw (OutOfRangeException, k);
        if (k > max) throw (OutOfRangeException, k);
        k++;
        number = 1 + number;
    }

    int max()
    { return k; }
    int m;
    Play (int N1, N2);
    int
    {
        while (1)
        {
            cin >> m;
            pa.Stop (m);
        }
    }
    void W() { W() < 0; }
}

```

```

cout << "Ural" << a.get_number() << " <u>Ur</u> <u>Ural</u> yashdi";
}

catch (Error e)
{
    cout << a.get_number() << " <u>Ur</u> <u>Ural</u>" << a.get_value() << " <u>Ural</u> yashdi";
}

return 0;
}

```

18.4. Vektor konteyneri sifat

Konteynerlar *containers* – bu turdagi elementlarni saqlashni osonlashtiradigan standart kutubxona. Standart kutubxonada vektor vektor konteyneri deb nomlanadi. Bu konteynerlar quyidagilarda uchraydi: `<vector>` ko'rsatgichlar shartli sifat bo'lib, klasslar elementlarini indeks orqali osonlashtiradi.

Vektorda saqlanadigan har qanday obyekt uchun ko'rsatilgan holda konstruktore ega bo'lgan yaratilgan. Bundan tashqari, obyekt uchun `<*>` operatorlar taqdim etilgan.

Vektor sifat uchun quyidagi xarakteristik operatorlar mavjud:

`size()`, `at()`, `begin()`, `end()`, `clear()`.

Bundan tashqari, vektor sifat uchun `[]` indeks operator taqdim etilgan.

Konstruktorlar

Har qanday vektor konteyner ko'rsatilgan holda konstruktore va destruktore ega bo'lgan standart konstruktorega ega.

Shunday, vektor sifat konteynerning konstruktore va destruktore:

<code>vector < elem > v</code>	<code>v</code> tipidagi elementga ega bo'lgan vektor yaratiladi.
<code>vector < elem > v(n)</code>	<code>n</code> ta elementga ega bo'lgan vektor yaratiladi (har bir elementning ma'nosini o'z ichiga oladi).
<code>vector < elem > v(n, elem)</code>	<code>n</code> ta elementga ega bo'lgan vektor yaratiladi (har bir elementning ma'nosini o'z ichiga oladi).
<code>vector < elem > v(n, elem, elem2)</code>	<code>n</code> ta elementga ega bo'lgan vektor yaratiladi (har bir elementning ma'nosini o'z ichiga oladi).
<code>vector < elem > v()</code>	Vektor yaratiladi (har bir elementning ma'nosini o'z ichiga oladi).

Konteyner vektorlari

Elementlarga ro'yxat

`operator[]()` vektordagi indeks bo'yicha ro'yxat.

`at()` vektordagi indeks bo'yicha ro'yxat.

Elementlarni kiritish usullari

`push_back()` oxiriga x kiriti qo'yish.

Elementlarni o'chirish usullari

`pop_back()` oxirgi elementni o'chirish.

Qirindash usullari

`operator=()` konteynerga x konteynerning elementlari o'zlashtiriladi.

Ishtirok usullari

`size()` elementlar soni.

`empty()` konteyner bo'lmadi?

`capacity()` vektor uchun ajratilgan tutqir.

`reserve()` n elementdan iborat bo'lgan konteyner uchun tutqir.

`operator()`

`swap()` ikkita konteynerning ichki tarkib almashtirish.

$n = 1, 2, \dots$ ta'kidlanish operatorlari.

Ikki vektor yaratish:

```
#include <vector>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
vector<int> v;
```

```
vector<int> b(5,1);
```

```
int b;
```

```
assigned b;
```

```
cout<<"size"<<" "<<b.size()<<"\n";
```

```
for(i=0; i<b.size(); i++)cout<<b[i]<<" ";
```

```
cout<<endl;
```

```
cout<<v<<"\n";
```

```
return 0;
```

```
}
```

Vektor-dinamik standart konteynerdir.

```
#include <vector>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
vector<int> v;
```

```
assigned b;
```

```
for(i=0; i<10; i++)v.push_back(i);
```

```
cout<<"size"<<" "<<b.size()<<"\n";
```

```
for(i=0; i<10; i++)cout<<b[i]<<" ";
```

```
cout<<endl;
```

```
for(i=0; i<10; i++)v[i]=v[i]+v[i];
```

```
for(i=0; i<v.size(); i++)cout<<b[i]<<" ";
```

```
cout<<endl;
```

```
v.push_back(1);
```

```
cout<<"size"<<" "<<b.size()<<"\n";
```

```
for(i=0;i<v.size();
```

```
return 0;
```

```
}
```

Maximal dinamik standart konteyner bo'lmadi.

```
#include <vector>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
assigned b;
```

```
vector<int> v(4,1);
```

```
vector<int> b(5,1);
```

```
cout<<"size"<<" "<<b.size()<<"\n";
```

```
cout<<"size"<<" "<<b.size()<<"\n";
```

```
for(i=0; i<b.size(); i++)cout<<b[i]<<" ";
```

```
cout<<endl;
```

```
for(i=0; i<b.size(); i++)cout<<b[i]<<" ";
```

```
v.swap(b);
```

```
cout<<endl;
```



```

#include <vector>
using namespace std;
template <class T>
void swap(vector<T> &a, vector<T> &b, vector<T> &c)
{
    for(unsigned i=0; i<a.size(); i++)
        a[i] = a[i] + b[i];
}

int main()
{
    unsigned N;
    vector<int> a(4,1);
    vector<int> b(4,5);
    vector<int> c(4);
    swap(a,b);
    for(i=0; i<c.size(); i++) c[i] = a[i] + b[i];
    return 0;
}

```

Можно использовать еще один вариант и в случае если:

```

#include <iostream>
#include <vector>
using namespace std;
class person
{
    string name;
    int year;
public:
    person();
    person(string name, int year)
    {
        person.name = name;
        person.year = year;
    }
    void show();
}

void << "name = " << name << "year = " << year << endl;

```

```

}

int main()
{
    vector<person> v(3);
    for(i=0; i<v.size(); i++)
        v[i] = person("John", 45);
    v[1] = person("Anna", 30);
    v[2] = person("John", 55);
    for(i=0; i<3; i++) v[i].show();
    cout << endl;
    return 0;
}

```

Очень важно помнить, что при использовании методов класса М необходимо помнить, что для переменных класса М не существует, только для переменных класса М. После завершения работы программы, все переменные, которые были созданы, будут уничтожены.

```

#include <iostream>
#include <vector>
using namespace std;
class N
{
    int n;
public:
    N(int k) {n=k;}
    int get_value() const {return n;}
}

class M
{
    int k;
public:
    M(int k) {k=k;}
    int get_value() const {return k;}
}

class E

```

```

1
2
3 int m;
4 int k;
5 public:
6     Error(int k1, int k2) {m=k1; k=k2;}
7     int get_number() const {return m;}
8     int get_value() const {return k;}
9
10 class Play
11 {
12     vector<int> ps;
13     int number;
14     int size;
15     void print()
16     {
17         int k=ps.size();
18         for(int i=0; i<k; i++)cout<<ps[i]<<" ";
19         cout<<endl;
20     }
21     public:
22     Play(int a)
23     {
24         if(a<2)throw Error(1);
25         ps.push_back(a);
26         number=a;
27         if(a==1) size=1; else size=a;
28     }
29     void swap(int k, int index)
30     {
31         if(k<1 || k>=ps.size())throw Error(number, k);
32         ps.push_back(k);
33         ps[index]=k;
34         if(k==1) size++;
35         if(ps[index]==1) size++;
36         if(size==ps.size() || throw Error(number));
37         number=1+number;
38         print();
39     }

```

```

1
2
3 int main()
4 {
5     int k;
6     int m;
7     int a=2;
8     if
9     {
10         Play ps(a);
11         while(1)
12         {
13             cin>>k>>m;
14             ps.swap(k, m);
15             cout<<endl;
16             cout<<ps.get_number()<<"*yashil yashil";
17             cout<<endl;
18             cout<<ps.get_value()<<"xax";
19             cout<<endl;
20             cout<<ps.get_number()<<"*yashil" <<ps.get_value()<<"xax yashil";
21             cout<<endl;
22             return 0;
23         }
24     }

```

Yechish to'g'ri yoki xato

1. Kiritilgan sonni tekshirib, qancha yoki qancha kiritilgan sonni?
2. Kiritilgan sonni tekshirib, qancha yoki kiritilgan sonni?
3. Kiritilgan sonni tekshirib, qancha yoki kiritilgan sonni?
4. Kiritilgan sonni tekshirib, qancha yoki kiritilgan sonni?
5. Kiritilgan sonni tekshirib, qancha yoki kiritilgan sonni?

- C' = + salibni shah-qorinasi yordamida ilovaning ustidagi shah.
- Iloraga ko'rsatib berilgan qiyomatli harakat;
- Iloralarini solishtirish;
- Iloralar ustida solishtirish amallari bo'yicha;
- Iloralar o'zaro shah.

Ko'rsatkichlar ustida o'zgarishlarni anislar. Ko'rsatkichlar ustida o'zgarishlar hajarida mavjud: inkrement ++ va dekrement -- amallari. Hajaridagi, ko'rsatkich ustida ko'rsatkich mavjud qilgan bir amallarga ko'rsatkich o'zgarish.

100

```

len = len(a)
p1 = 0
p2 = 1
p3 = 2
p4 = 3
p5 = 4
p6 = 5
p7 = 6
p8 = 7
p9 = 8
p10 = 9
p11 = 10
p12 = 11
p13 = 12
p14 = 13
p15 = 14
p16 = 15
p17 = 16
p18 = 17
p19 = 18
p20 = 19
p21 = 20
p22 = 21
p23 = 22
p24 = 23
p25 = 24
p26 = 25
p27 = 26
p28 = 27
p29 = 28
p30 = 29
p31 = 30
p32 = 31
p33 = 32
p34 = 33
p35 = 34
p36 = 35
p37 = 36
p38 = 37
p39 = 38
p40 = 39
p41 = 40
p42 = 41
p43 = 42
p44 = 43
p45 = 44
p46 = 45
p47 = 46
p48 = 47
p49 = 48
p50 = 49
p51 = 50
p52 = 51
p53 = 52
p54 = 53
p55 = 54
p56 = 55
p57 = 56
p58 = 57
p59 = 58
p60 = 59
p61 = 60
p62 = 61
p63 = 62
p64 = 63
p65 = 64
p66 = 65
p67 = 66
p68 = 67
p69 = 68
p70 = 69
p71 = 70
p72 = 71
p73 = 72
p74 = 73
p75 = 74
p76 = 75
p77 = 76
p78 = 77
p79 = 78
p80 = 79
p81 = 80
p82 = 81
p83 = 82
p84 = 83
p85 = 84
p86 = 85
p87 = 86
p88 = 87
p89 = 88
p90 = 89
p91 = 90
p92 = 91
p93 = 92
p94 = 93
p95 = 94
p96 = 95
p97 = 96
p98 = 97
p99 = 98
p100 = 99

```

[illegible]

```
int* parr1, *parr2, a[10];
int i=2;
parr1 = a + (i+4); // = a[6] (address of a[6])
parr2 = parr1 + 1; // = a[7] (address of a[7])
```

[illegible]

100

```

int *p1, *p2, a[10];
int i;
p1 = a + 5;
p2 = a + 9;

```

1. *Journal of Management Studies*, 1996, 33, 1, 1-15.
 2. *Journal of Management Studies*, 1996, 33, 2, 1-15.

[illegible]

10

```

int *p1, *p2, *p3;
p1 = a + 5;
p2 = a + 7;
if (p1 < p2) *p3 = *p1 + 1;

```

The molecule part ring system part ring system having, showing
where all of constant molecular weight.

Erkrankter meistler. Erkrankter meistler psychische
schwierig

1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 26

1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 26

For more information, visit www.pearsoncmg.com

U

© 2004 Blackwell Publishing Ltd *Journal of Internal Medicine* 255: 105–112

Бүткүлүлүк менен келишимдештирилген, анткени бул
бир келишимдин бир бөлүгү болуп саналат.

Diamant masivlar, O'zbekiston o'rtacha o'lchamli masivlarini shakllantirish lo'riksizliklar va to'rtinchi diamant topilishlari to'rtinchi yilida natijalarni ko'rsatadi.

Notions: Journal, required when new or delete operations are
performed. Conversion

Source: *Author's calculations*.

Salah satu contoh pengolahan data kuantitatif terdapat pada tabel berikut ini. Tabel tersebut menunjukkan jumlah penduduk di suatu desa menurut jenis kelamin dan umur. Jumlah penduduk tersebut akan diolah dengan cara sebagai berikut.

Paralelamente, a nova organização dos serviços de saúde, que privilegia a atenção primária, também tem sido desenvolvida.

122 123. *Streptococcus* *mutans* i karieskall. Streptocoga betecknats *apromans* i karieskallmanen i 124 uttrycket *Streptococcus* *mutans* i karieskall.

Obyektning umumiy xususlariga munosib uchun $\rightarrow$ operatoridan $y(x)$ (ya'ni shundayki x va y o'zaro operatoriyalish foydalanishi mumkin).

(*PAJ.mak) (yoki PAJ $\rightarrow$ mak)

filial:

#include <iostream>

#include <string>

using namespace std;

class complex

{

float x;

float y;

public:

complex(float x1 = 0, float y1 = 0):x(x1), y(y1){}

void show();

{

cout<<"x="<<x<<"y="<<y<<"\n";

}

};

int main()

{

complex a(1.0, 2.0);

complex *pa=&a;

pa->show();

complex b;

complex *pb=&b;

(*pb).show();

return 0;

}

Dastur bajarilishi natijasi

x=1.0 y=2.0

x=0.0 y=0.0

Bu ko'rsatkichi. Agarbi ko'rsat obyekti bilan ko'rsat uchun $\rightarrow$ operatoridan foydalanish uchun, unda bu funktsiyaga obyektga tegishli bo'lgan ko'rsatkich avtomatik va ko'rsatilmagan holda yaratiladi. Bu ko'rsatkich

bu ko'rsatga o'zgaruvchi bu ko'rsat bilan ko'rsatilmagan holda yaratiladi.

Masalan, x ko'rsat ko'rsatilmagan holda yaratiladi:

class x {

int m;

public:

int reading() {return this->m;}

};

X ko'rsatiga munosib uchun bu ko'rsat foydalanish o'rinli. Ammo bu ko'rsat ko'rsatilmagan holda yaratiladi va ko'rsatilmagan holda yaratiladi.

#include <iostream>

using namespace std;

class Pair

{

int N;

double x;

friend Pair& operator ++(Pair&);

friend Pair& operator --(Pair&, int);

public:

Pair(int N, double x);

{

N=N; x=x;

}

void display();

{

cout<<"Koordinatlar: N="<<N<<"x="<<x<<"\n";

}

Pair& operator ++(Pair&);

{

N=N+1; x=x+1;

return *this;

}

Pair& operator --(Pair&);

{

N=N-1;

x=x-1;

```

return P;
}
}
// Faktorial operator ++ (Fakt: P)
{
    P.NP = 10; P.x2 = 10;
    return P;
}
// Faktorial operator ++ (Fakt: P, int k)
{
    P.N = P.NP2 = k;
    P.x = P.x2 = k;
    return P;
}
int main()
{
    Fakt Z(10, 100);
    Z.display();
    ++Z;
    Z.display();
    --Z;
    Z.display();
    Z++;
    Z.display();
    Z--;
    Z.display();
    return 0;
}

```

Virtual member va ko'rsatkichlar. Ajdod sifat ko'rsatkichiga a'loq sifat ajdod sifatini qiyamat sifatida berish va bu ko'rsatkich orqali a'loq sifat sifatini chaqirish mumkin. Quyidagi misolda ajdod sifat va a'loq sifat bir o'ldirga ega bo'lgan bo'lgan.

```

class
{
public:
    virtual ~(){}
    using namespace std;
    class base

```

```

{
public:
    virtual void print() {cout << "base\n";}
}

class der public base
{
public:
    void print() {cout << "der\n";}
}

int main()
{
    base B;
    der D;
    base *bp = &B;
    bp->print(); // base
    bp = &D;
    bp->print(); // der
    return 0;
}
// Nopis
// base
// der

```

Bu misolda a'loq sifat sifatiga a'loq qiyamat sifatida berilgan ajdod sifat sifatiga ko'rsatkich yoki base orqali a'loqda qo'yilgan yuldan a'loq chaqirishga o'lib borish mumkin. Agar funktsiya virtual bo'lsa, ajdod sifat orqali virtual a'loq sifat sifatini chaqirish.

Shunday qilib, ajdod sifat sifatiga ko'rsatkich orqali virtual sifat chaqirish natijasi a'loq ko'rsatkich qiyamat, yoki chaqirish bajarilayotgan a'loq sifat sifatini ko'rsatadi.

Quyidagi virtual funktsiyani chaqirish ko'rsatkich orqali a'loq ko'rsatkich qiyamatidan bajarilish jarayonida a'loq sifat sifatiga bog'liq.

```

// Misol
// Nopis
// base
// der

```

```
using namespace std;
class superbase
{
public:
    virtual void print() { cout << "superbase\n"; }
};

class basepublic: superbase
{
public:
    void print() { cout << "base\n"; }
};

class der: public superbase
{
public:
    void print() { cout << "der\n"; }
};

int main()
{
    base b;
    der d;
    superbase *bp = &b;
    bp->print(); // base
    bp = &d;
    bp->print(); // der
    return 0;
}
```

19.1. Konstruktor va Destruktor

Destruktor. Shitibing bitta obyektidan so'ng yaratilgan natija obyektini yo'qotilganidan so'ng beriladigan kod. Shitibing maxsus komponentlari destruktorelar bu variabla avtomatik bajarish imkonini yaratadi. Destruktorning standart shakli quyidagicha:

```
~name() { ... } // destruktore nomi
```

Destruktor parametrlari yoki qaytariladigan qiymatga ega bo'lmishi mumkin va uning natija qoidasi bor. Destruktor obyektini yaratilganda destruktore avtomatik bajariladi.

Agarda shiftda destruktore o'ziga ko'rsatilgan bo'lsa, unda kompilyator ko'rsatilgan obyekt yaratilgan natijani berilgan natija destruktoreni yaratilganda. Shitibing obyektini yaratilgan natijani berilgan natija bo'lsin, destruktoreni o'ziga o'ziga berilgan natija. Shitibing, yaratilgan natija o'ziga berilgan natija.

```
#include <iostream>
#include <string.h>
using namespace std;
class Person
{
    char * name;
    int age;
public:
    Person(char * n, int MAge):
    {
        name = n;
        age = MAge;
    }
    ~Person()
    {
        name = NULL;
        age = 0;
    }
    int GetAge() const
    {
        return age;
    }
    char * GetName() const
    {
        return name;
    }
};
```


populae supracoracae brevis, alaba nigridorsalis nigri latelli. Supra-
interhyaline et al. dorsum supracoracae latell. glauc. immo popae brevis glauc.
et alaba nigridorsalis latelli.

[illegible]

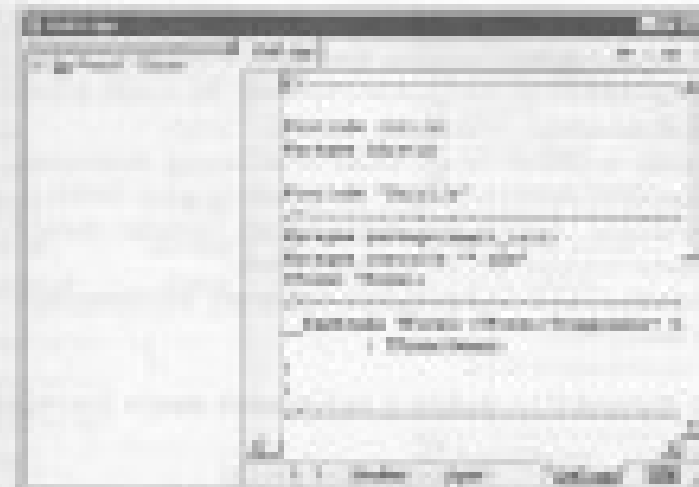
Bir moddining barcha oʻzgarishi berilgan muhitda faqat amaliy oʻrnatilgan tarkibga, shuningdek shartlarga bogʻliq boʻlganligi, boshqa barcha oʻzgarishlar afchalaridagi harakatlanish va koʻchirilishning natijasi sifatida sodir boʻladi.

Visual komponentalar pildirib, Visual komponentalar pildirib, bir qancha o'quvchi boshida tashqi, qolganlari esa to'rtinchi o'quvchi komponentalar ustida boshida tashqi.

Elektr (mashin) markasida tomon (shakl) davlatni joylashgan (28.1 rasmi, bu belgisi davlatning iqtisodiy birligi sifatida xizmat qiladi. Osimning nomi va tarkibidagi ushbu tashkilot tomonidan beriladi. Osimning



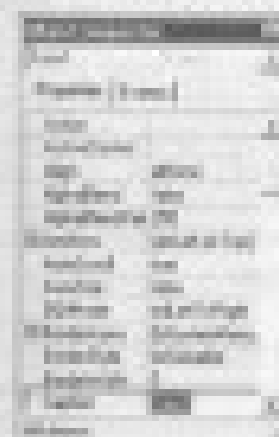
doi:10.1017/S0022292412001907



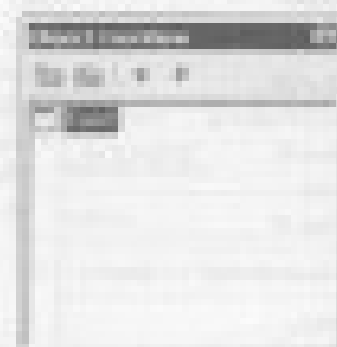
2000-2001, 2001-2002, 2002-2003, 2003-2004, 2004-2005, 2005-2006, 2006-2007, 2007-2008, 2008-2009, 2009-2010, 2010-2011, 2011-2012, 2012-2013, 2013-2014, 2014-2015, 2015-2016, 2016-2017, 2017-2018, 2018-2019, 2019-2020, 2020-2021, 2021-2022, 2022-2023, 2023-2024, 2024-2025, 2025-2026, 2026-2027, 2027-2028, 2028-2029, 2029-2030, 2030-2031, 2031-2032, 2032-2033, 2033-2034, 2034-2035, 2035-2036, 2036-2037, 2037-2038, 2038-2039, 2039-2040, 2040-2041, 2041-2042, 2042-2043, 2043-2044, 2044-2045, 2045-2046, 2046-2047, 2047-2048, 2048-2049, 2049-2050, 2050-2051, 2051-2052, 2052-2053, 2053-2054, 2054-2055, 2055-2056, 2056-2057, 2057-2058, 2058-2059, 2059-2060, 2060-2061, 2061-2062, 2062-2063, 2063-2064, 2064-2065, 2065-2066, 2066-2067, 2067-2068, 2068-2069, 2069-2070, 2070-2071, 2071-2072, 2072-2073, 2073-2074, 2074-2075, 2075-2076, 2076-2077, 2077-2078, 2078-2079, 2079-2080, 2080-2081, 2081-2082, 2082-2083, 2083-2084, 2084-2085, 2085-2086, 2086-2087, 2087-2088, 2088-2089, 2089-2090, 2090-2091, 2091-2092, 2092-2093, 2093-2094, 2094-2095, 2095-2096, 2096-2097, 2097-2098, 2098-2099, 2099-2100, 2100-2101, 2101-2102, 2102-2103, 2103-2104, 2104-2105, 2105-2106, 2106-2107, 2107-2108, 2108-2109, 2109-2110, 2110-2111, 2111-2112, 2112-2113, 2113-2114, 2114-2115, 2115-2116, 2116-2117, 2117-2118, 2118-2119, 2119-2120, 2120-2121, 2121-2122, 2122-2123, 2123-2124, 2124-2125, 2125-2126, 2126-2127, 2127-2128, 2128-2129, 2129-2130, 2130-2131, 2131-2132, 2132-2133, 2133-2134, 2134-2135, 2135-2136, 2136-2137, 2137-2138, 2138-2139, 2139-2140, 2140-2141, 2141-2142, 2142-2143, 2143-2144, 2144-2145, 2145-2146, 2146-2147, 2147-2148, 2148-2149, 2149-2150, 2150-2151, 2151-2152, 2152-2153, 2153-2154, 2154-2155, 2155-2156, 2156-2157, 2157-2158, 2158-2159, 2159-2160, 2160-2161, 2161-2162, 2162-2163, 2163-2164, 2164-2165, 2165-2166, 2166-2167, 2167-2168, 2168-2169, 2169-2170, 2170-2171, 2171-2172, 2172-2173, 2173-2174, 2174-2175, 2175-2176, 2176-2177, 2177-2178, 2178-2179, 2179-2180, 2180-2181, 2181-2182, 2182-2183, 2183-2184, 2184-2185, 2185-2186, 2186-2187, 2187-2188, 2188-2189, 2189-2190, 2190-2191, 2191-2192, 2192-2193, 2193-2194, 2194-2195, 2195-2196, 2196-2197, 2197-2198, 2198-2199, 2199-2200, 2200-2201, 2201-2202, 2202-2203, 2203-2204, 2204-2205, 2205-2206, 2206-2207, 2207-2208, 2208-2209, 2209-2210, 2210-2211, 2211-2212, 2212-2213, 2213-2214, 2214-2215, 2215-2216, 2216-2217, 2217-2218, 2218-2219, 2219-2220, 2220-2221, 2221-2222, 2222-2223, 2223-2224, 2224-2225, 2225-2226, 2226-2227, 2227-2228, 2228-2229, 2229-2230, 2230-2231, 2231-2232, 2232-2233, 2233-2234, 2234-2235, 2235-2236, 2236-2237, 2237-2238, 2238-2239, 2239-2240, 2240-2241, 2241-2242, 2242-2243, 2243-2244, 2244-2245, 2245-2246, 2246-2247, 2247-2248, 2248-2249, 2249-2250, 2250-2251, 2251-2252, 2252-2253, 2253-2254, 2254-2255, 2255-2256, 2256-2257, 2257-2258, 2258-2259, 2259-2260, 2260-2261, 2261-2262, 2262-2263, 2263-2264, 2264-2265, 2265-2266, 2266-2267, 2267-2268, 2268-2269, 2269-2270, 2270-2271, 2271-2272, 2272-2273, 2273-2274, 2274-2275, 2275-2276, 2276-2277, 2277-2278, 2278-2279, 2279-2280, 2280-2281, 2281-2282, 2282-2283, 2283-2284, 2284-2285, 2285-2286, 2286-2287, 2287-2288, 2288-2289, 2289-2290, 2290-2291, 2291-2292, 2292-2293, 2293-2294, 2294-2295, 2295-2296, 2296-2297, 2297-2298, 2298-2299, 2299-2300, 2300-2301, 2301-2302, 2302-2303, 2303-2304, 2304-2305, 2305-2306, 2306-2307, 2307-2308, 2308-2309, 2309-2310, 2310-2311, 2311-2312, 2312-2313, 2313-2314, 2314-2315, 2315-2316, 2316-2317, 2317-2318, 2318-2319, 2319-2320, 2320-2321, 2321-2322, 2322-2323, 2323-2324, 2324-2325, 2325-2326, 2326-2327, 2327-2328, 2328-2329, 2329-2330, 2330-2331, 2331-2332, 2332-2333, 2333-2334, 2334-2335, 2335-2336, 2336-2337, 2337-2338, 2338-2339, 2339-2340, 2340-2341, 2341-2342, 2342-2343, 2343-2344, 2344-2345, 2345-2346, 2346-2347, 2347-2348, 2348-2349, 2349-2350, 2350-2351, 2351-2352, 2352-2353, 2353-2354, 2354-2355, 2355-2356, 2356-2357, 2357-2358, 2358-2359, 2359-2360, 2360-2361, 2361-2362, 2362-2363, 2363-2364, 2364-2365, 2365-2366, 2366-2367, 2367-2368, 2368-2369, 2369-2370, 2370-2371, 2371-2372, 23

Form 1 (Form 2, Form 3, Form 4) is designed to assist teachers in identifying and addressing the needs of students who are at risk of dropping out of school. The form is designed to be used by teachers and school administrators to identify students who are at risk of dropping out of school and to develop a plan to address their needs. The form is designed to be used by teachers and school administrators to identify students who are at risk of dropping out of school and to develop a plan to address their needs.

Forma abstrak dipaparkan melalui gambar ke-6. Melalui gambar tersebut, dapat dilihat bahwa pada bagian atas terdapat gambar logo dari PT. KALAMATI, dan pada bagian bawah terdapat gambar logo dari PT. KALAMATI.



© 2004 Blackwell Publishing Ltd *Journal of Internal Medicine* 255: 103–110



© 2000 Blackwell Science Ltd *Journal of Internal Medicine* 247: 399–406

Ekraning chap qismida Obyektlar Inspektori oynasi joylashgan. Bu oyna Object Inspector sifatida ham ataladi (28.5-rasm). Bu oynada visual komponentlarga xarakteristikalar aʼrovlashadi. Bu oynani oltinchi oʻrta uchlagan ekran ustiga joylashtirish mumkin.

Obyektlar Inspektori oynasi ustida obyektning darajasimon tuzilishi oynasi (28.6-rasm) joylashgan. Bu oynada dasturlagi barcha obyektlar darajali strukturali shaklda ifodalangan boʻladi. Formalar, dastur kod va boshqa dastur tashkiloti fayllari koʻrilgan boʻladi.

Bu interfeys shaxs munosabatida eng oson foydalaniladigan qism boʻladi.

28.1. Standart C++ Editor 4-mahallasi asosiy menyasi bayroqlari

File bayroqlari guruhi

Menyuning birinchi File (Fayl) bayroqlari guruhi 28.7-rasmida koʻrilgan.

Bu menyada bayroqlari fayllar bilan ishlashga moʻljallangan va quyidagi amallarni bajaradi: (New) yangi fayl yaratish, dastur oynasi ochish, mavjud fayllarni (Open) ochish, fayllarni saqlash (Save) va (Close) yopish, yangi dastur kod yaratish (Print) chop qilish mumkin.



28.7-rasm. File bayroqlari guruhi

Edit bayroqlari guruhi

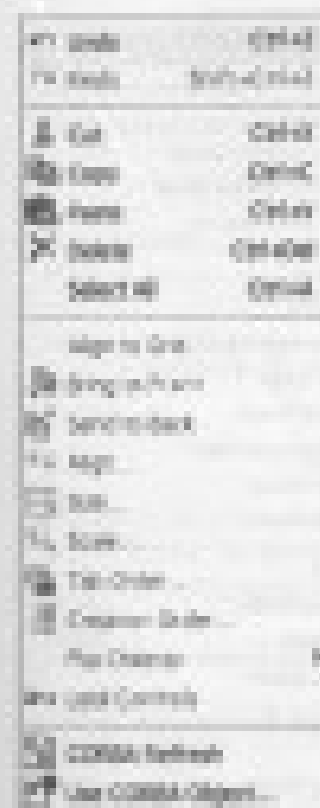
Edit (Tahrir) 28.8-rasmida ifodalangan.

Menyuning bu bayroqlari guruhi kabi faʼl qilish uchun tahrir koʻrilgan bayroqlardir. Masalan, kiritish qilish (Undo), va (Redo) qaytarish, qopqit qilish (Cut), nusxa olish, chiqarish, olinga nusxani qoʻyish (Paste) va oʻchirish (Delete) kabi amallarni oʻz ichiga olgan.

Search bayroqlari guruhi

Search (Izlash) 28.9-rasmida ifodalangan.

Menyuning bu bayroqlari guruhi fayllardan ilmiy maʼlumot izlash va aniqlashni oʻz



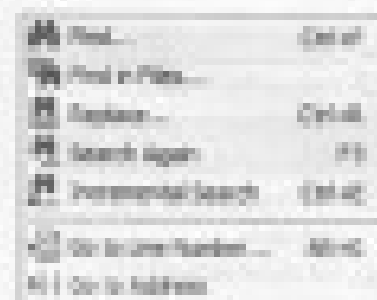
28.8-rasm. Edit bayroqlari guruhi

maʼlumot, kerakli narsa oʻrnatish va uni oʻzgartirish imkonini beradi.

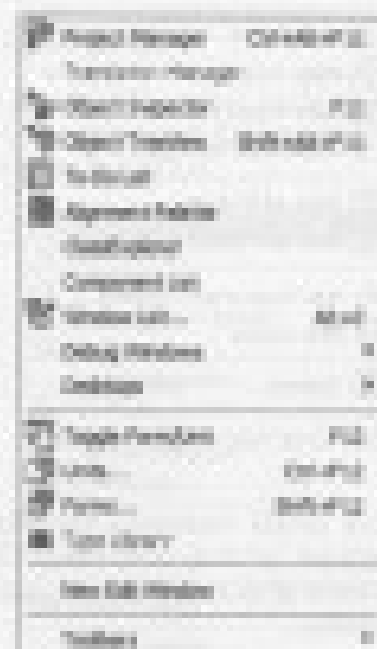
View bayroqlari guruhi

View (Koʻrish) 28.10-rasmida ifodalangan.

Menyuning bu bayroqlari guruhi bilan shaxs va komponentlarni koʻrish uchun asosiy mahallani oynalari ochiladi. Masalan, shaxs mahallasi (Project Manager), komponentlar roʻyxati (Component List) va oynalar roʻyxati (Window List).



28.9-rasm. Search bayroqlari guruhi



28.10-rasm. View bayroqlari guruhi



20.1.1-rasm. Project boyraqlari guruhi

Ma'yaning bu boyraqlari guruhi hadda ko'proq bo'lganida boyraqlari yig'ilgan. Bu boyraqlar yordamida birlari qo'llash, o'chirish, VCL komponentlari kutubxonasiga kiritilgan bo'lsa, qo'llash, o'chirish va shunga o'xshash amallarni bajarish mumkin.

Bin boyraqlari guruhi

Bin (Binar) 20.1.2-rasmida ko'rsatilgan.

Ma'yaning bu boyraqlari guruhi yordamida binlar o'ziga kiritilish va kiritilish, binlar o'ziga kiritilish va o'ziga kiritilish bo'yicha o'ziga kiritilish, binlar o'ziga kiritilish va o'ziga kiritilish bo'yicha o'ziga kiritilish va o'ziga kiritilish bo'yicha o'ziga kiritilish.

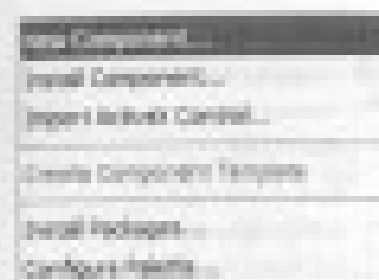


20.1.2-rasm. Bin boyraqlari guruhi

Project boyraqlari guruhi

Project (Binlar) 20.1.3-rasmida ko'rsatilgan.

Ma'yaning bu boyraqlari guruhi hadda ko'proq bo'lganida boyraqlari yig'ilgan. Bu boyraqlar yordamida birlari qo'llash, o'chirish, VCL komponentlari kutubxonasiga kiritilgan bo'lsa, qo'llash, o'chirish va shunga o'xshash amallarni bajarish mumkin.



20.1.3-rasm. Component boyraqlari guruhi

Component boyraqlari guruhi

Component (Komponent) boyraqlari guruhi 20.1.4-rasmida ko'rsatilgan. Ma'yaning bu boyraqlari guruhi yordamida yangi komponentlarni qo'llash va komponentlarni o'chirish bo'yicha o'ziga kiritilish.

Database boyraqlari guruhi

Database (Ma'lumotlar bazasi) boyraqlari guruhi 20.1.5-rasmida ko'rsatilgan. Ma'yaning bu boyraqlari guruhi yordamida ma'lumotlar bazasi bilan ishlash bo'yicha o'ziga kiritilish.

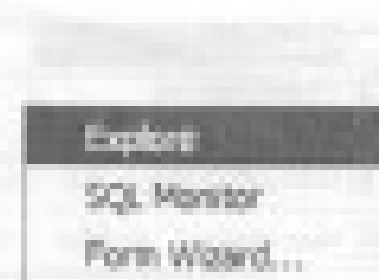
Tools boyraqlari guruhi

Tools (Instrumentlar) boyraqlari guruhi 20.1.6-rasmida ko'rsatilgan.

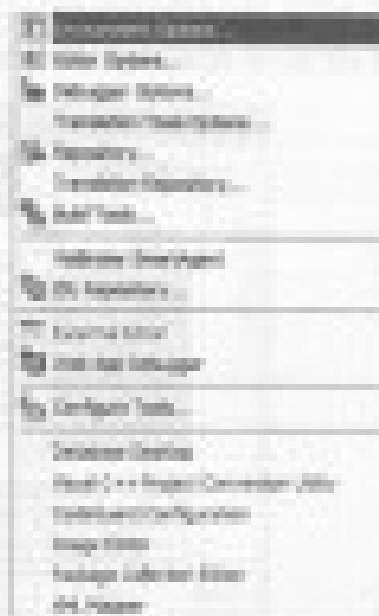
Ma'yaning bu boyraqlari guruhi yordamida ilmiy instrumentlarni o'ziga kiritilish va yordamchi dasturlar boyraqlarni o'ziga kiritilish.

Windows boyraqlari guruhi

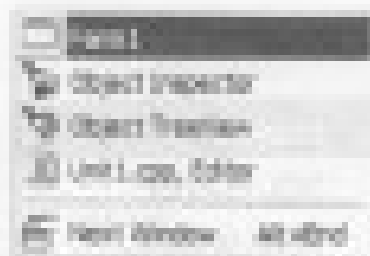
Windows (Bytal) boyraqlari guruhi 20.1.7-rasmida ko'rsatilgan.



20.1.7-rasm. Windows boyraqlari guruhi



20.1.5-rasm. Database boyraqlari guruhi



28.19-rasm. Help menyusut gambar

Mengurangi bu menyusut gambar pada jendela menyempit tampilan beberapa menu.

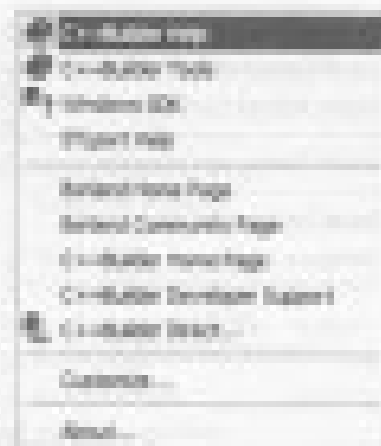
Help menyusut gambar

Help i Windows menyusut gambar 20.19-rasmnya terlihat.

Mengurangi bu menyusut gambar pada jendela menu, id, komponentalar va kumpu men haplog, na kumutkani kumutkani menu (20.19-rasm).



28.20-rasm. Menyempit gambar



28.21-rasm. Help menyusut gambar

Teleskop menyusut

Teleskop menyusut menyempit tampilan beberapa jendela (20.22-rasm).



28.22-rasm. Teleskop menyusut

Teleskop menyusut menyempit tampilan beberapa jendela (20.22-rasm). Menyempit tampilan menyempit tampilan beberapa jendela (20.22-rasm).

Standard (Standart) menyusut

Standard menyusut menyempit tampilan beberapa jendela (20.23-rasm). Menyempit tampilan menyempit tampilan beberapa jendela (20.23-rasm).



28.23-rasm. New Window menyusut gambar

[illegible][illegible]

- majuskel komponenta variga sordlik;
- jargi vuvulyarlar, metodlar va vopoznani anapash;
- vuvulyar komponentar, vuvulyar vuvulyar.

[illegible]

Holzer & Holzer DEX yazarlık. Bu kitabın yayımlanması, A. yazarlık E. Holzer yayımlanması amacıyla Holzer, yazarlık yayımlanması.

[illegible][illegible][illegible]

Komponen dari polimerisasi antara H_2O dan BaCl_2 dan $\text{Ba}(\text{OH})_2$ yang terdistribusikan. Hal ini terjadi sebagai akibat dari distribusi molekul. Dengan adanya katalis polimerisasi yang terdistribusikan akan menghasilkan polimer yang terdistribusikan. Agar polimerisasi dapat menghasilkan polimer yang terdistribusikan, maka perlu ada katalis. Untuk komposisi polimerisasi yang terdistribusikan, maka katalis akan terdistribusikan yang terdistribusikan. Maka katalis akan terdistribusikan. (M. H. H. H.)

[illegible]

Pharmaceutical industry will face regulatory battle despite new rules. **Pharmaceutical** industry will face regulatory battle despite new rules.

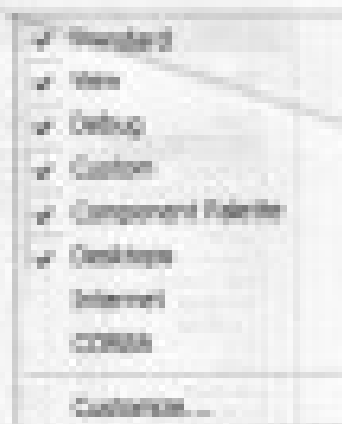
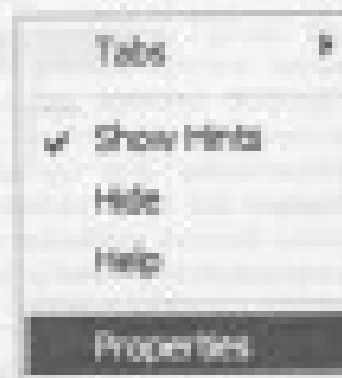
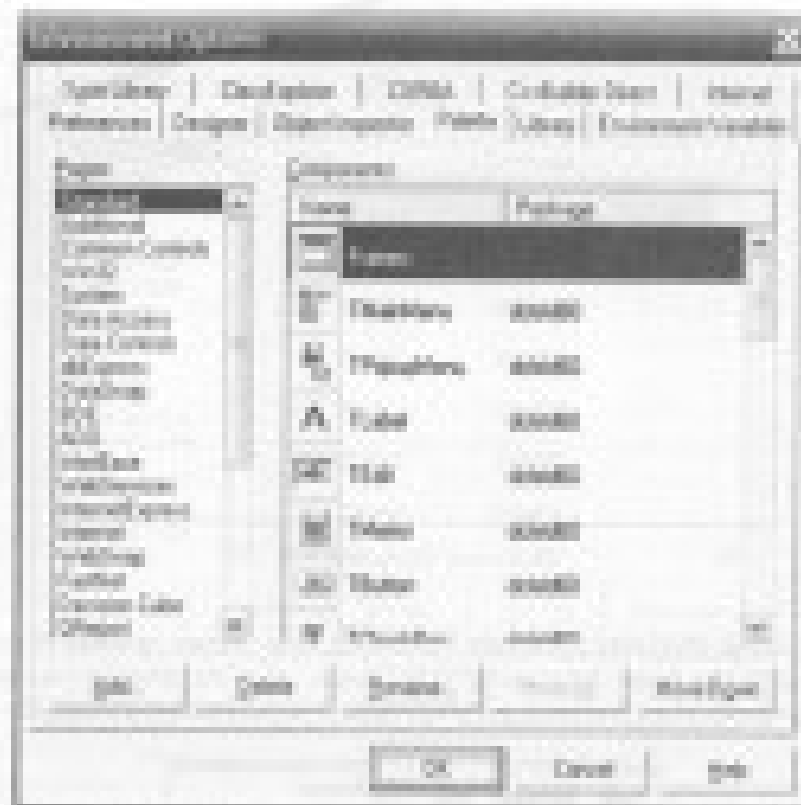


Fig. 1. Schematic diagram of the proposed method.

et al. 2004) and the presence of a large number of species in the same genus (e.g. *Phragmites* and *Scirpus*) may be a result of the high degree of genetic variability within the genus (e.g. *Phragmites* and *Scirpus*).



2014-2015
 May transfer petition
 Transfer of 2014-2015



28.24-variant. Diagram ustalashning bir qismini: Options ko'rsatmoqda.

Ushbu qadami Grid size filitrida berilgan. Ma'lumotlarda gorizontal X va vertikal Y parametrlar ham berilgan. Formadagi turli olib tashlash ham mumkin. Display grid maydonidagi belgini olib tashlashda belgi. Komponentalarning katta bog'lanmagan holda ko'rsatilishini uchun Snapgrid varidagi belgini olib tashlashimiz kerak bo'ladi. Bu elementlar barchasi Grid options guruhiga tegishli. Options guruhidagi barcha komponentalar olib tashlashning ko'rsatkichlari berilganish uchun Show component outlines ko'rsatkich berilgan. Show designer filtri ko'rsatkich berish berish uchun ko'rsatkich berilgan. Show extended control filtri ko'rsatkich berish berish uchun ko'rsatkich berilgan. Bunda joylashgan Module creation options guruhidagi barcha elementlar New form as text belgining qadami ko'rsatkichlari.

umiyatda shunday belgi yoki ma'lum belgi ko'rsatkich berilgan uchun ma'lum berilgan. Ma'lum berilgan elementlar ma'lum berilgan. Ma'lum berilgan elementlar ma'lum berilgan. Ma'lum berilgan elementlar ma'lum berilgan.

Amaliyot uchun berilgan belgi ko'rsatkich berilgan. Ma'lum berilgan belgi ko'rsatkich berilgan. Ma'lum berilgan belgi ko'rsatkich berilgan. Ma'lum berilgan belgi ko'rsatkich berilgan.

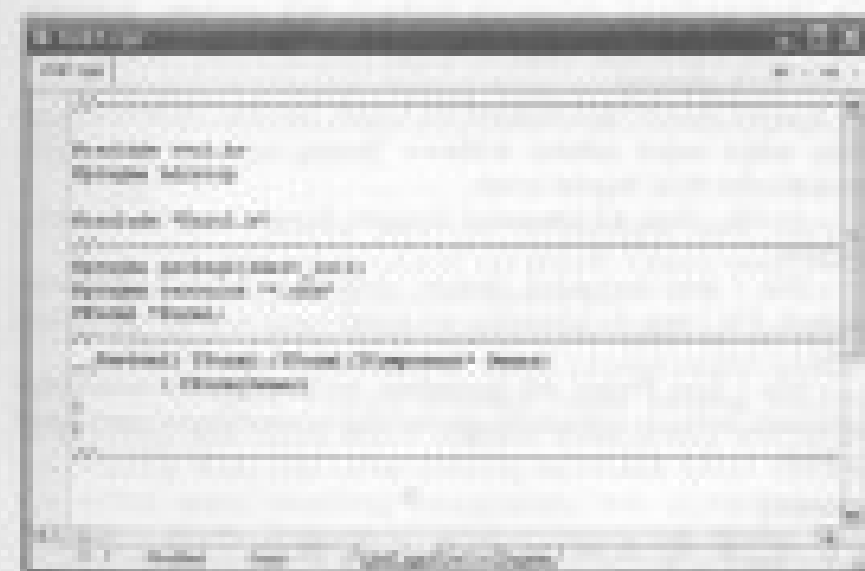
Bundan keyin «Delete» tugmasi uchun belgi ko'rsatkich berilgan. Ma'lum berilgan belgi ko'rsatkich berilgan. Ma'lum berilgan belgi ko'rsatkich berilgan. Ma'lum berilgan belgi ko'rsatkich berilgan.



28.25-variant. Formga o'tish.

Kod matnini

Form ma'lum belgi olib uchun kod matnini berilgan. Ma'lum berilgan belgi olib uchun kod matnini berilgan. Ma'lum berilgan belgi olib uchun kod matnini berilgan. Ma'lum berilgan belgi olib uchun kod matnini berilgan.



28.26-variant. Kod matnini berilgan.

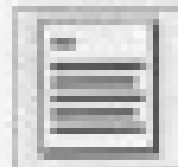
Table



Asbosiya yoki natijaning tabeli qoʻyilgan. Kiritiladigan yozibchilik sohasi shaklida oqib ketadi. Tabeli sohasi ichidagi boshlangʻich nusxalarni. Tana qiymani boʻlgan qisqartiriladi.

Table komponenti TTableControl shifoning soʻyidan soʻyri tashqari boʻlib, ushbu barcha nusxalarni, mosdori va nusxalarga varidlik qilib.

Table



Asbosiya kripti natijaning tabeli qoʻyilgan. Kiritiladigan yozibchilik sohasi shaklida oqib ketadi. Tabeli sohasi ichidagi boshlangʻich nusxalarni. Tana qiymani boʻlgan qisqartiriladi. Ushbu nusxalarni qiymani boʻlgan qisqartiriladi. Ushbu nusxalarni qiymani boʻlgan qisqartiriladi.

Table komponenti TTableControl shifoning soʻyidan soʻyri tashqari boʻlib, ushbu barcha nusxalarni, mosdori va nusxalarga varidlik qilib.

Table



Asbosiya kripti natijaning tabeli qoʻyilgan. Kiritiladigan yozibchilik sohasi shaklida oqib ketadi. Tabeli sohasi ichidagi boshlangʻich nusxalarni. Tana qiymani boʻlgan qisqartiriladi.

Table komponenti TTableControl shifoning soʻyidan soʻyri tashqari boʻlib, ushbu barcha nusxalarni, mosdori va nusxalarga varidlik qilib.

Table komponenti TTableControl shifoning soʻyidan soʻyri tashqari boʻlib, ushbu barcha nusxalarni, mosdori va nusxalarga varidlik qilib.

Table



Asbosiya kripti natijaning tabeli qoʻyilgan. Kiritiladigan yozibchilik sohasi shaklida oqib ketadi. Tabeli sohasi ichidagi boshlangʻich nusxalarni. Tana qiymani boʻlgan qisqartiriladi.

Table komponenti TTableControl shifoning soʻyidan soʻyri tashqari boʻlib, ushbu barcha nusxalarni, mosdori va nusxalarga varidlik qilib.

Table komponenti TTableControl shifoning soʻyidan soʻyri tashqari boʻlib, ushbu barcha nusxalarni, mosdori va nusxalarga varidlik qilib.

Table komponenti TTableControl shifoning soʻyidan soʻyri tashqari boʻlib, ushbu barcha nusxalarni, mosdori va nusxalarga varidlik qilib.

Table



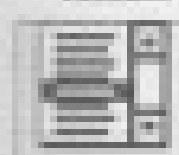
Asbosiya kripti natijaning tabeli qoʻyilgan. Kiritiladigan yozibchilik sohasi shaklida oqib ketadi. Tabeli sohasi ichidagi boshlangʻich nusxalarni. Tana qiymani boʻlgan qisqartiriladi.

Table komponenti TTableControl shifoning soʻyidan soʻyri tashqari boʻlib, ushbu barcha nusxalarni, mosdori va nusxalarga varidlik qilib.

Table komponenti TTableControl shifoning soʻyidan soʻyri tashqari boʻlib, ushbu barcha nusxalarni, mosdori va nusxalarga varidlik qilib.

Table komponenti TTableControl shifoning soʻyidan soʻyri tashqari boʻlib, ushbu barcha nusxalarni, mosdori va nusxalarga varidlik qilib.

Table



Asbosiya kripti natijaning tabeli qoʻyilgan. Kiritiladigan yozibchilik sohasi shaklida oqib ketadi. Tabeli sohasi ichidagi boshlangʻich nusxalarni. Tana qiymani boʻlgan qisqartiriladi.

Table komponenti TTableControl shifoning soʻyidan soʻyri tashqari boʻlib, ushbu barcha nusxalarni, mosdori va nusxalarga varidlik qilib.

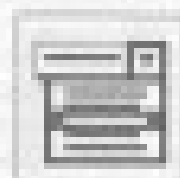
elementlarida qoʻyilish, oʻchirish, yangi joylash va oʻrindi almashtirish mumkin. Masalan:

ListBox1->Items->Add ("Kafesdagi yangi element")

Kafed xususiyatining true qiymati boʻlgan elementlarni albto ushbu usulda qoʻyish mumkin.

ListBox komponentini TFormWinListBox shifining boʻlimi boʻlib, ushbu hamma xususiyat, metod va voqealariga ega boʻlib qoʻyiladi.

TCComboBox



Takrir uchun hamda roʻyxatlarining koʻrinish qoʻyilgan xususiyat koʻrsatkichini tanlash uchun ishlatiladi.

True xususiyatining qiymati boʻlganda takrir uchun qoʻyilish kerak boʻlgan ushbu elementlar, hama xususiyatining ichida boʻladi. Dasturlash boʻlimidagi paytida takrir uchun elementlar boʻlgan element roʻyxati koʻrsatkich xususiyatining ichida boʻladi. Tanlash olingan elementning oʻzi esa TComboBox xususiyatining ichida boʻladi. SetItems va TComboBox xususiyatini tanlash ushbu qiymat takrir uchun qoʻyilish kerak boʻlgan yoki tanlash ushbu qiymat takrir uchun qoʻyilish kerak boʻlgan boʻladi.

hama elementning Add, Append, Delete va Insert metodlari yordamida roʻyxat elementlarini dasturlash ushbu usulda qoʻyilish, oʻchirish, yangi qoʻyilish va oʻrindi almashtirish mumkin. Masalan:

ComboBox->Items->Insert(0, "Kafesdagi yangi element")

Kafed xususiyatining true element roʻyxat elementlarining albto ushbu usulda tanlash qoʻyilish kerak boʻladi. TCComboBox komponentining turli style xususiyatlarini tanlash uchun mumkin.

TCComboBox komponentini TFormWinComboBox shifining boʻlimi boʻlib, ushbu hamma xususiyatlar, metodlar va voqealariga ega boʻlib qoʻyiladi.

TDynList



Dinamik, shakl yoki boshqa komponentni ichidagi elementlarini koʻrinish qoʻyilish uchun ishlatiladi. Masalan, hamma parametrlar qiymatining boʻlgan interval ichida koʻrsatkichini ushbu xususiyat qoʻyilish kerak boʻlgan boʻladi.

qoʻyilish kerak boʻlgan element koʻrsatkichini TFormWinDynList shifining boʻlimi boʻlib, ushbu hamma xususiyatlar, metodlar va voqealariga ega boʻlib qoʻyiladi.

TCComboBox komponentini TFormWinTCComboBox shifining boʻlimi boʻlib, ushbu hamma xususiyatlar, metodlar va voqealariga ega boʻlib qoʻyiladi.

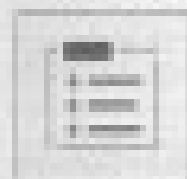
TCComboBox komponentini TFormWinTCComboBox shifining boʻlimi boʻlib, ushbu hamma xususiyatlar, metodlar va voqealariga ega boʻlib qoʻyiladi.

TCGroupBox



Togʻri koʻrinishdagi takrir koʻrinishdagi komponent boʻlib, a qoʻyilish kerak boʻlgan elementlarining hamma xususiyatlarini qoʻyilish kerak boʻlgan boʻladi.

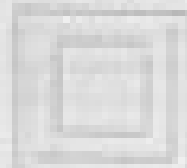
TRadioGroup



Togʻri koʻrinishdagi takrir koʻrinishdagi komponent boʻlib, a koʻrinishdagi hamma xususiyatlarini qoʻyilish kerak boʻlgan boʻladi.

Radio tugmalar hamma komponentga qoʻyilish kerak boʻlgan boʻladi. Bu qoʻyilish kerak boʻlgan boʻladi. RadioGroup komponentiga tugma qoʻyilish uchun, hamma xususiyatining qoʻyilish kerak boʻlgan boʻladi. RadioGroup komponentiga tugma qoʻyilish uchun, hamma xususiyatining qoʻyilish kerak boʻlgan boʻladi.

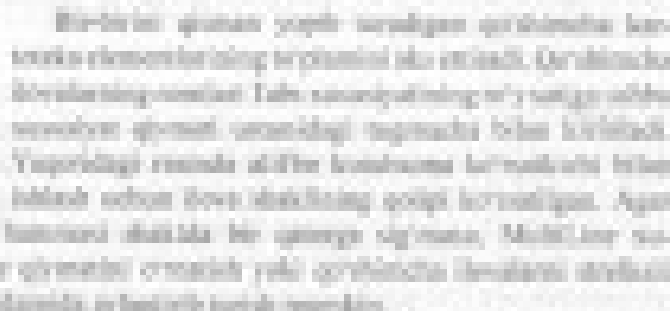
TPanel



Boʻlim komponentini oʻr qoʻyilish uchun ishlatiladi. Masalan, hamma parametrlar qiymatining boʻlgan interval ichida koʻrsatkichini ushbu xususiyat qoʻyilish kerak boʻlgan boʻladi.

TPanel komponentini TFormWinTPanel shifining boʻlimi boʻlib, ushbu hamma xususiyatlar, metodlar va voqealariga ega boʻlib qoʻyiladi.

1999

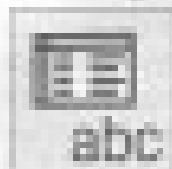


1998



Agar *Stomoxys* qandimcha *Trinoda* bi sporiga sig'mansa, *Leptomyia* yul'tuzlik, tug'masidan chiqaradi. Qandimcha shirakarni bi nashri qat'iyda shi o'tirish natijasi shahidlar ko'rsatayotirgan shi qandimcha bo'lgan.

54. *Modifikasi struktur dan fungsi* (misalnya, perubahan struktur dan fungsi sel, perubahan struktur dan fungsi jaringan, perubahan struktur dan fungsi organ, perubahan struktur dan fungsi sistem organ, perubahan struktur dan fungsi organisme).



Ushbu komponentga tegishli barcha xususiyatlar ning nomlari va vazifalari bo'lib, ular shundan iboratki: tegishli komponentda n'la foydalanilishini ta'minlaydi. Ular keyingi paragraflarda batafsil tushirilgan. TDrawGrid komponentni xususiyatlariga to'g'ri keladi.

Xususiyatlar komponentda birinchi bo'lib ko'rsatilgan komponentda ko'rsatilgan shaklida ko'rsatilgan. O'zgaruvchilarda ko'rsatilgan bo'lgan. Dasturlarning bajarilishi paytida komponentlar ko'rsatilgan va ular ko'rsatilgan shaklida ko'rsatilgan. Ular komponentda ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan.

TDrawGrid



TDrawGrid komponenti ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan.

O'zgaruvchilarda ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan.

DefaultDrawing komponenti ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan.

Dasturlarning bajarilishi paytida ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan.

Dasturlarning bajarilishi paytida ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan.

MTImage



MTImage komponenti ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan.

MTImage komponenti ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan.

MTImage komponenti ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan.

Image -> Picture -> LoadFromFile -> LoadFromFile -> LoadFromFile

Image -> Picture -> SaveToFile -> SaveToFile -> SaveToFile

TShape



TShape komponenti ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan.

TShape komponenti ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan.

TImage



TImage komponenti ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan.

TImage komponenti ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan. Komponentlar ko'rsatilgan shaklida ko'rsatilgan.

Ukupan jumlah dari hasil belajar pada uji statistik di kelas eksperimen dan kelas kontrol dapat dilihat sebagai berikut.

© 2006 The Authors
Journal compilation © 2006 Blackwell Publishing Ltd

Gratik kanvas. Grafik kanvas man kanvas tegayochi idoni tegayochi, lekin x o'qida ko'rinmaydi. Ma'lumki, mana kanvas shindagi belgi o'rtasini (80*15 ko'rsatildi) ko'rsatadi va bu o'rtasida belgi chop qilimgan da avtomatik ravishda bir o'rtas o'rtasga ko'chiradi. Grafik kanvas esa chiqariluvchi grafik shaklning bo'ylabing bel ko'rsatilganini ko'rsatadi va uni keyingi (x, y) nuqtasiga ko'chirish uchun maxsus funktsiya taqdimadi:

Form1->Canvas->MoveTo(x, y);

Chiziqlarni chizish. To'rtburchak chiziqlari (demon) chizish uchun Form1->Canvas->LineTo() funktsiyasidan foydalaniladi. Masalan, (x1, y1) va (x2, y2) nuqtalarni to'rtburchakni kanvas shindagi uchun quyidagi amr bilan taqdimadi kanvas:

Form1->Canvas->MoveTo(x1, y1);

Form1->Canvas->LineTo(x2, y2);

Kanvasda ko'p nuqtadagi chiziqli chiziqlardan tashkil topgan shaklning chizish uchun Canvas->PolylineTo(x, y) funktsiyasidan foydalaniladi. U bo'ylab kanvasdagi uchun joriy nuqtasini bilan aniqlangan chiziqli chiziqlar chizadi. u parametri chiziqli chiziqli tegayochi kanvas uchun. Ikkinchi parametri TPoint turida berilib, grafik uchun nuqta ko'rsatilganini aniqlaydi shunda kanvas manzil. Shu chiziqli tegayochi kanvas uchun manzil o'rtasida berilib. Quyida PaintBox1 komponentini shunda (x, y) funktsiya grafikasi chiziqli funktsiya taqdimolgan.

void Sin_Grafikasi(TPaintBox)

```
{
const double PI= 3.14159265;
double Qadam=0.1;
double Ruchak_Radius=0;
const int Nuqtalar_Soni=100;
int i=0;
TPaint Sin_Sini(Nuqtalar_Soni);
int Abstrak=PaintBox1->Height/2;
for (int i=0; i<Nuqtalar_Soni; i++)
{
Sin_Sini[i].x=(int)(Math::Sin(i*Qadam)*Ruchak_Radius)+Ab;
Sin_Sini[i].y=Abstrak-(int)(Math::Cos(i*Qadam)*Ruchak_Radius);
Ruchak_Radius+=Qadam;
}
PaintBox1->Canvas->Pen->Color=clBlack;
```

PaintBox1->Canvas->PolylineTo(Sin_Sini, Nuqtalar_Soni);

}

Quyidagi dasturda sin(x) funktsiya grafikasi shindagi chiziqlar qo'yilgan.

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->vcl1->

Form1->

Form1->

Form1->

Form1->

Form1->

Form1->

Form1->

Form1->

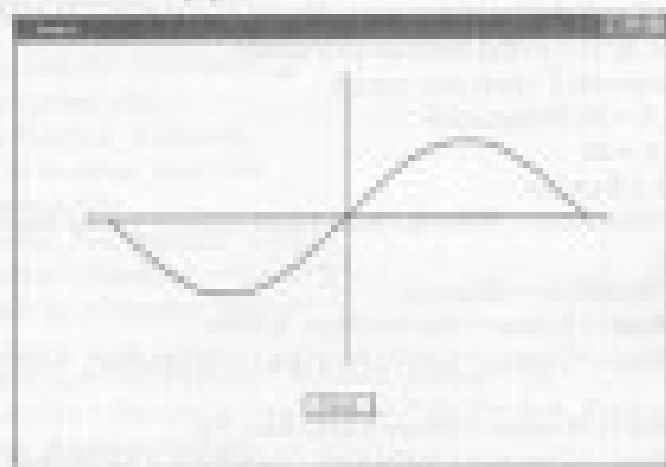
Form1->

```

X=X+1;
Y=sin(X);
X=X+1+(sin(M_PI*X));
Y=Y+1-(sin(M_PI*X));
PaintBox 1->Canvas->MoveTo(Xs, Ys);
PaintBox 1->Canvas->Pen->Color=clRed;
do
begin
  X=X+1;
  Y=sin(X);
  X=X+1+(sin(M_PI*X));
  Y=Y+1-(sin(M_PI*X));
  if(X>Rast_X) then X:=PaintBox 1->Width-Rast_X;
  if(Y>Rast_Y) then Y:=PaintBox 1->Height-Rast_Y;
  PaintBox 1->Canvas->LineTo(Xs,Ys); //Koordinata shag'irib olib
end;
while (X<=Rast_X);

```

Dastur ishlatilgan natijada Formadagi PaintBox 1 komponentida uchburchak 20.43-rasmda ko'rsatilgan paydo bo'ladi.

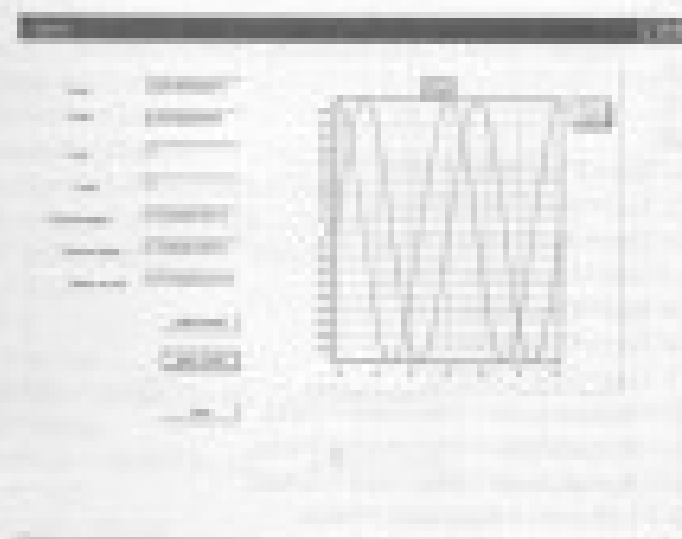


20.43-rasm. Dastur natijada hosil bo'lgan rasml

20.44- va 20.45-rasmdagi esa dastur va ushbu funktsiyalarining grafiklari shunday ko'rinishda bo'ladi.



20.44-rasm. Dastur natijada hosil bo'lgan rasml



20.45-rasm. Dastur natijada hosil bo'lgan rasml

Double kxh;

WinTitle="vcl.h";

Synopsis Submap;

WinTitle="math.h";

WinTitle="tmath.h";

```
.....  
Synopsis packageversion_info  
Synopsis version "1.0.0"  
TForm1 *Form1;  
Real Xmin, Xmax, Ymin, Ymax, Hx, Hy, k, Pl;  
.....  
__fastcall TForm1::TForm1(TComponent* Owner)  
: TForm(Owner)  
{  
}  
.....
```

void __fastcall TForm1::FormCreate(TObject *Sender)

{Xmin=2*M_PI;

Xmax=2*M_PI;

Ymin=1;

Ymax=5;

Hx=0.1;

Hy=0.1;

k=0.001;

Edit1->Text=FloatToStr(Xmin);

Edit2->Text=FloatToStr(Xmax);

Edit3->Text=FloatToStr(Ymin);

Edit4->Text=FloatToStr(Ymax);

Edit5->Text=FloatToStr(Hx);

Edit6->Text=FloatToStr(Hy);

Edit7->Text=FloatToStr(k);

Chart1->BottomAxis->Automatic=False;

Chart1->BottomAxis->Minimum=Xmin;

Chart1->BottomAxis->Maximum=Xmax;

Chart1->LeftAxis->Automatic=False;

Chart1->LeftAxis->Minimum=Ymin;

Chart1->LeftAxis->Maximum=Ymax;

Chart1->BottomAxis->Increment=Hx;

Chart1->LeftAxis->Increment=Hy;

};

void __fastcall TForm1::Button1Click(TObject *Sender)

{

Xmin=StrToFloat(Edit1->Text);

Xmax=StrToFloat(Edit2->Text);

Ymin=StrToFloat(Edit3->Text);

Ymax=StrToFloat(Edit4->Text);

Hx=StrToFloat(Edit5->Text);

Hy=StrToFloat(Edit6->Text);

Chart1->BottomAxis->Minimum=Xmin;

Chart1->BottomAxis->Maximum=Xmax;

Chart1->LeftAxis->Minimum=Ymin;

Chart1->LeftAxis->Maximum=Ymax;

Chart1->BottomAxis->Increment=Hx;

Chart1->LeftAxis->Increment=Hy;

};

void __fastcall TForm1::Button2Click(TObject *Sender)

{Plot x, y1, y2;

Series1->Chart1;

Series2->Chart1;

Xmin=StrToFloat(Edit1->Text);

Xmax=StrToFloat(Edit2->Text);

k=StrToFloat(Edit7->Text);

n=Xmin;

do {

Ymin(n);

Series1->AddXY(n, y1, " ", ofIntColor);

y2=exp(n);

Series2->AddXY(n, y2, " ", ofIntColor);

n+=1;

while(n < Xmax);

};


```

    MSKey+=12;
}
if(Label6->Caption="Edit">Tombol4(Label7->Caption="Edit")>Click
+ Isap;
if(1<Label6->Caption="Tampilkan hasil")>Isap;
Label6->Caption="Tampilkan hasil" + Isap;
Label7->Caption="Masukkan hasil" + Isap;
Tombol4(Isap);
}

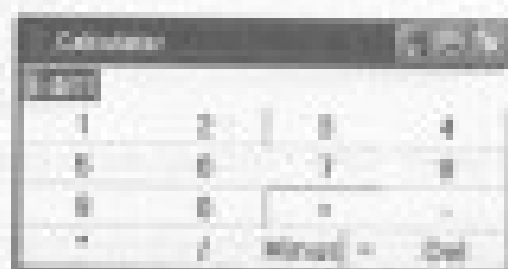
```

1. Kalkulator dasar

Baru akan membuat *GUI* aplikasi yang di dalamnya terdapat *GUI* kursor. Sehingga aplikasi akan memiliki tampilan otomatis (dalam kalkulator) yang akan digunakan. Kita akan membuat *GUI* kursor dengan *Python* C++ Builder. Kita akan membuat *GUI* kursor, dan kita akan membuat *GUI* kursor yang akan digunakan.

Membuat *GUI*

Membuat *GUI*



20.10.2019. Screenshot aplikasi kalkulator.

Untuk membuat *GUI* kursor.

Kita akan membuat *GUI* kursor yang akan digunakan.

Kita akan membuat *GUI* kursor yang akan digunakan.



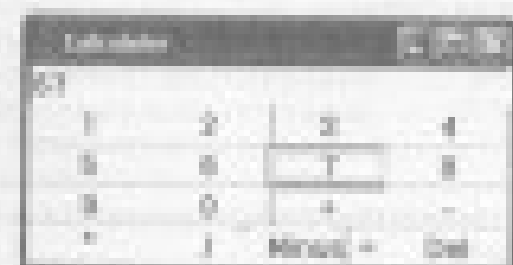
20.10.2019. Screenshot aplikasi kalkulator.

Kita akan membuat *GUI* kursor.



20.10.2019. Screenshot aplikasi kalkulator.

Kita akan membuat *GUI* kursor.



20.10.2019. Screenshot aplikasi kalkulator.

Calculator			
78			
1	2	3	4
5	6	7	8
9	0	+	-
*	/	Memory	Del

20.4.2000. Dastur loyihalashning natijasi (qarang rasviro)

Bunda kalkulyatorning rasviro bo'lishi. Rasviro topmasi uchun rasviro bo'lishi topmasi bo'lishi rasviro bo'lishi bo'lishi. Arifmetik amallar rasviro bo'lishi amallar topmasi uchun rasviro bo'lishi topmasi bo'lishi rasviro bo'lishi. Del topmasi kalkulyatorning rasviro bo'lishi rasviro bo'lishi.

Dastur kod

```
#include <stdio.h>
#pragma hdrstop
```

```
#include <stdlib.h>
```

```
#pragma pack(push, 1)
```

```
#pragma warning (d4151)
```

```
TCalculator *Calculator;
```

```
double a, b, c=0;
```

```
int main (int argc,
```

```
char **argv)
```

```
{
    Calculator *Calculator = (Calculator *) malloc(sizeof(Calculator));
```

```
    if (Calculator == NULL)
```

```
{
```

```
    return 1;
```

```
void __fastcall TCalculator::FormCreate(TObject *Sender)
```

```
{
```

```
    // TODO: Add your initialization code here
```

```
}
```

```
void __fastcall TCalculator::FormClose(TObject *Sender)
```

```
{
    if (Edit1->Text.Length > 0) ShowMessage ("Xatirangizni saqlashni istaysizmi?");
```

```
    if (Edit1->Text.Length > 0)
```

```
    {
        ShowMessage ("Xatirangizni saqlashni istaysizmi?");
```

```
        if (Edit1->Text.Length > 0)
```

```
        {
            ShowMessage ("Xatirangizni saqlashni istaysizmi?");
```

```
            if (Edit1->Text.Length > 0)
```

```
            {
                ShowMessage ("Xatirangizni saqlashni istaysizmi?");
```

```
            }
        }
    }
}
```

```
}
```

```
void __fastcall TCalculator::Click(TObject *Sender)
```

```
{
    if (Edit1->Text.Length > 0)
```

```
    {
        ShowMessage ("Xatirangizni saqlashni istaysizmi?");
```

```
    }
}
```

```
}
```

```
}
```

```
void __fastcall TCalculator::FormClose(TObject *Sender)
```

```
{
    if (Edit1->Text.Length > 0) ShowMessage ("Xatirangizni saqlashni istaysizmi?");
```

```
    if (Edit1->Text.Length > 0)
```

```
    {
        ShowMessage ("Xatirangizni saqlashni istaysizmi?");
```

```
        if (Edit1->Text.Length > 0)
```

```
        {
            ShowMessage ("Xatirangizni saqlashni istaysizmi?");
```

```
            if (Edit1->Text.Length > 0)
```

```
            {
                ShowMessage ("Xatirangizni saqlashni istaysizmi?");
```

```
                if (Edit1->Text.Length > 0)
```

```
                {
                    ShowMessage ("Xatirangizni saqlashni istaysizmi?");
```

```
                    if (Edit1->Text.Length > 0)
```

```
                    {
                        ShowMessage ("Xatirangizni saqlashni istaysizmi?");
```

```
                    }
                }
            }
        }
    }
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```

j = 0; Edit->Text=>Text=nil)
//-----
void __fastcall TCalculator::EditClick(TObject *Sender)
{j = 0; Edit->Text=>Text=nil}
//-----
void __fastcall TCalculator::DivClick(TObject *Sender)
{j = 0; Edit->Text=>Text=nil}
//-----
void __fastcall TCalculator::ModClick(TObject *Sender)
{j = 0; Edit->Text=>Text=nil}
//-----
void __fastcall TCalculator::DivClick(TObject *Sender)
{j = 0; Edit->Text=>Text=nil}
//-----
void __fastcall TCalculator::SubClick(TObject *Sender)
{j = 0; Edit->Text=>Text=nil}
//-----
void __fastcall TCalculator::TriggedClick(TObject *Sender)
{j = 0; Edit->Text=>Text=nil}
//-----
void __fastcall TCalculator::MulClick(TObject *Sender)
{j = 0; Edit->Text=>Text=nil}
//-----
void __fastcall TCalculator::DelClick(TObject *Sender)
{if Edit->Text.Length > 0
ShowMessage('Edit: max length von Edit')
else Edit->Clear()}
//-----
void __fastcall TCalculator::MinuClick(TObject *Sender)
{if Edit->Text.Length > 0
ShowMessage('Edit: max length von Edit')
else {j:=1; i=Edit->Text; Edit->Clear(); i=StrToFloat(i, 0)}
//-----
void __fastcall TCalculator::AddClick(TObject *Sender)

```

```

}
if Edit->Text.Length > 0
ShowMessage('Edit: max length von Edit')
else {j:=1; i=Edit->Text; Edit->Clear(); i=StrToFloat(i, 0)}
//-----
void __fastcall TCalculator::DivClick(TObject *Sender)
{if Edit->Text.Length > 0
ShowMessage('Edit: max length von Edit')
else {j:=1; i=Edit->Text; Edit->Clear(); i=StrToFloat(i, 0)}
//-----

void __fastcall TCalculator::EditKeyPress(TObject *Sender, char
&Key)
{if (Key == 0) &Key := #0
ShowMessage('von Edit'); Key:=0; i:=i+(Key - CharToOrd('#'))
//-----
void __fastcall TCalculator::ButtonClick(TObject *Sender)
{j:=0; i=Edit->Text;}
//-----

```

1. Schritt: Kalkulator erstellt.
 Maske fertig.
 Designing overlay.



25.17-Kalkulator (Befehlsfeld fertig)

Bozda Edit maydoniga son kiritiladi. Sonlar tugmasichalar ustida o'zgaruvchi tugmalarini bosish orqali amalga oshiriladi. Axborotli amallar mas aniqrovi shu amallar tugmalar ustida o'zgaruvchi tugmalarini bosish orqali kiritiladi. Clear tugmasi Edit maydonini tozalash uchun xizmat qiladi.

Header kod:

```

#include <iostream>
#include <math.h>
#include <string.h>
#include <stdlib.h>
#include <conio.h>
#pragma hdrstop

#include "Unit.h"
//-----
#pragma pack(push, 1)
#pragma warning ("off")
#ifdef _MSC_VER
using a, b, c;
#endif
class Calc
{public: int ans;
void Plus();
void Minus();
void Div();
void Add();
void Tong();
};
Calc c;
void Calc::Plus()
{
int n;
n=Form1->Edit1->Text.Length();
if(n==0)

```

```

ShowMessage("Edit maydoniga son kiriting");
else
{ans=1, n=Form1->Edit1->Text; n=atoi(n.c_str());
Form1->Edit1->Clear();
}

void Calc::Minus()
{
int n;
n=Form1->Edit1->Text.Length();
if(n==0)
ShowMessage("Edit maydoniga son kiriting");
else
{ans=2, n=Form1->Edit1->Text; n=atoi(n.c_str());
}

void Calc::Add()
{
int n;
n=Form1->Edit1->Text.Length();
if(n==0)
ShowMessage("Edit maydoniga son kiriting");
else
{ans=3, n=Form1->Edit1->Text; n=atoi(n.c_str());
}
}

__fastcall TForm1::TForm1(TComponent* Owner)
: TForm1(Owner)
{
}

void Calc::Div()
{

```

```

int n;
n = Form1->Edit1->Text.Length();
if(n == 0)
    ShowMessage("Edit may be empty, not clicking");
else
    (n % 4 == 0 ? Form1->Edit1->Text = n * 4 : Form1->Edit1->Text = n * 4);
Form1->Edit1->Clear();
}
}

```

void CalcLength()

```

{
    if(Form1->Edit1->Text.Length() == 0)
        ShowMessage("Edit may be empty, not clicking");
    else
    {
        int n = 0;
        for(int i = 0; i < Form1->Edit1->Text.Length(); i++)
        {
            n++;
        }
        if(n % 4 == 0)
            Form1->Edit1->Text = Form1->Edit1->Text * 4;
        else
            Form1->Edit1->Text = Form1->Edit1->Text * 4;
        Form1->Edit1->Clear();
    }
}

```

```

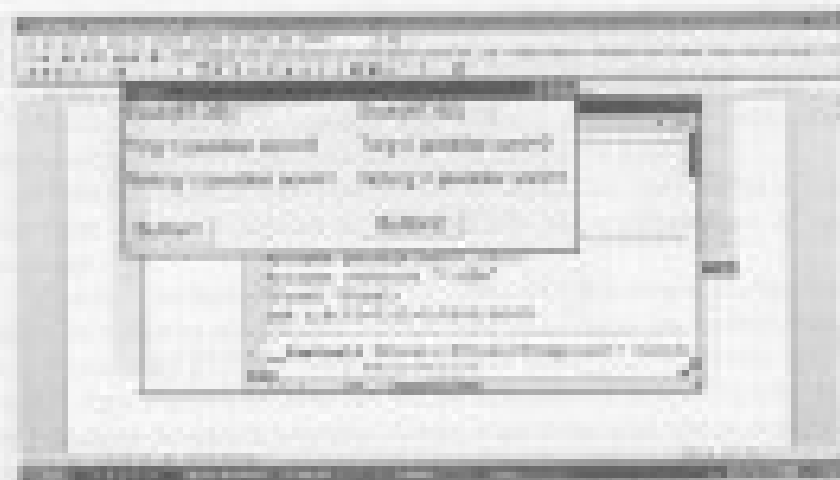
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
void __fastcall TForm1::Button1Click(TObject *Sender)
{
}

```

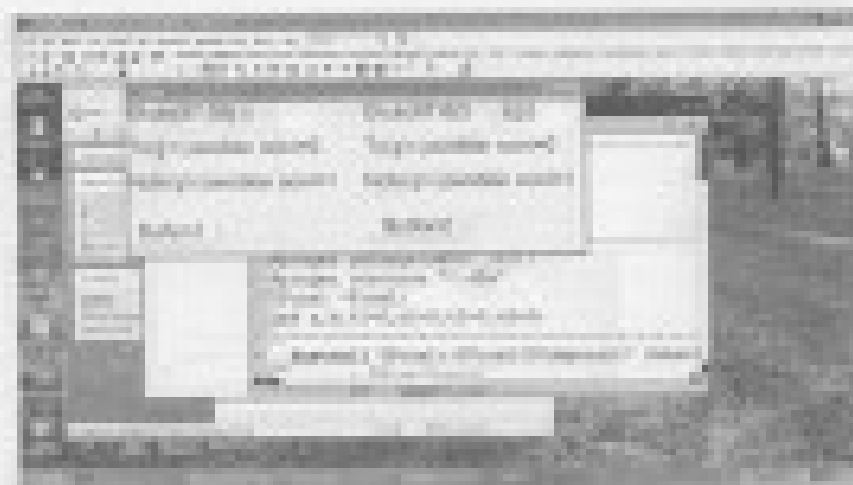
```

{
    if(n == 0)
        ShowMessage("Edit may be empty, not clicking");
    else
    {
        int n = 0;
        for(int i = 0; i < Form1->Edit1->Text.Length(); i++)
        {
            n++;
        }
        if(n % 4 == 0)
            Form1->Edit1->Text = Form1->Edit1->Text * 4;
        else
            Form1->Edit1->Text = Form1->Edit1->Text * 4;
        Form1->Edit1->Clear();
    }
}

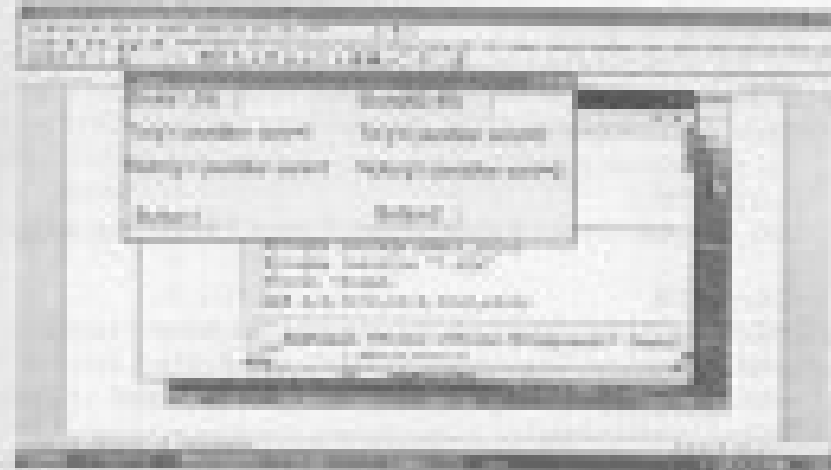
```

J2SE IDE: Find Usages (showing usages) dialog



J2SE IDE: Show Usages (showing usages) dialog



J2SE IDE: Show Usages (showing usages) dialog

class

```

class

```

```

class

```

```

class

```

```

class

```

```

class

```

```

class

```

```

class

```

```

class

```

```

class

```

```

class

```

```

class

```

```

class

```

```

class

```

```

class

```

```

class

```

```

class

```

```

class

```

```

class

```

```

class

```

```

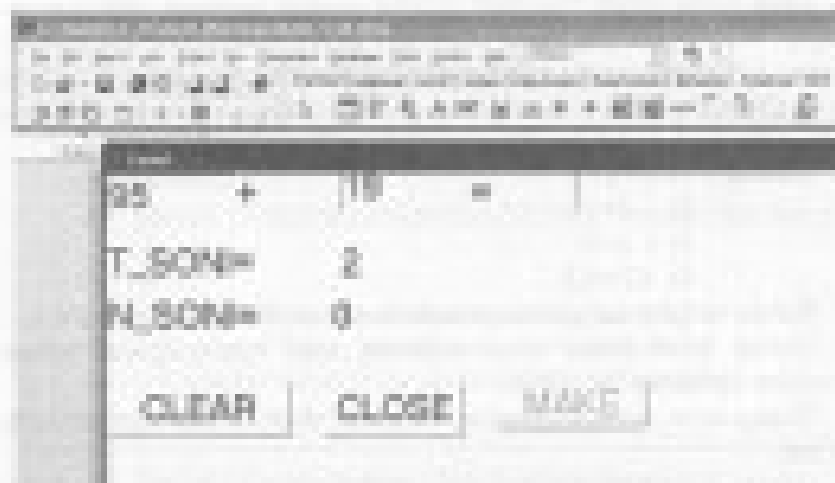
class

```

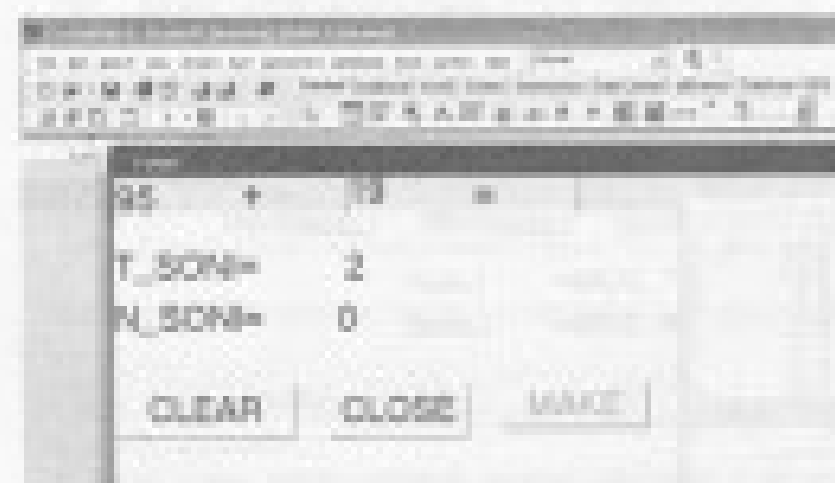
```

class

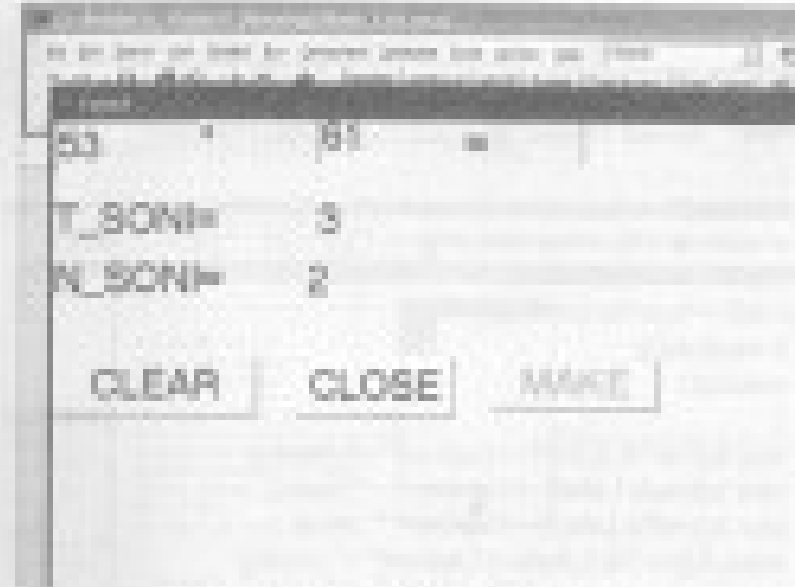
```

2042-figure: Simple Input/Output widget (before)



2042-figure: Simple Input/Output widget (after)



2042-figure: Simple Input/Output widget (before)

Header file

```

#ifndef SIMPLEIO_H
#define SIMPLEIO_H

#include <QWidget>
#include <QPushButton>

#include "ui_simpleio.h"

namespace packagestart_name {
namespace namespace ""_dnn {
    class SimpleIO : public QWidget {
        Q_OBJECT
    public:
        SimpleIO(QWidget *parent = 0);
        ~SimpleIO();
        void setValues(int a, int b, int c, int d);
        void show();
    private:
        void __init__(QWidget *parent = 0);
        void __cleanup__();
    };
}
}

#endif

```

```

}

void __fastcall TForm1.Button1Click(TObject *Sender)
{
  Edit1->Clear();
  n=random(100); b=random(100);
  Edit1->Text=IntToStr(n);
  Edit2->Text=IntToStr(b);
  n=random(1);
  switch(n)
  {
    case 0: n:=1; b:=Label5->Caption="1"; break;
    case 1: n:=1; b:=Label5->Caption="2"; break;
    case 2: n:=1; b:=Label5->Caption="3"; break;
    case 3: n:=1; b:=Label5->Caption="4"; break;
  }
}

void __fastcall TForm1.Button2Click(TObject *Sender, char &Key)
{
  Button2->Enabled=false;
  if Key=='+')
  {
    n:=n+IntToStr(Edit1->Text);
    n:=n+1;
    Label1->Caption=IntToStr(n);
    Label2->Caption=IntToStr(b);
    Button2Click(Sender);
  }
}

void __fastcall TForm1.Button3Click(TObject *Sender)
{
  Edit1->Clear();
  Edit2->Clear();
  Edit3->Clear();
}

```

```

void __fastcall TForm1.FormCreate(TObject *Sender)
{
  Randomize(); Label5->Caption="1";
}

```

4. CPU test option disabled inside
 Dimer interface via built (50.68-50.72-ram).

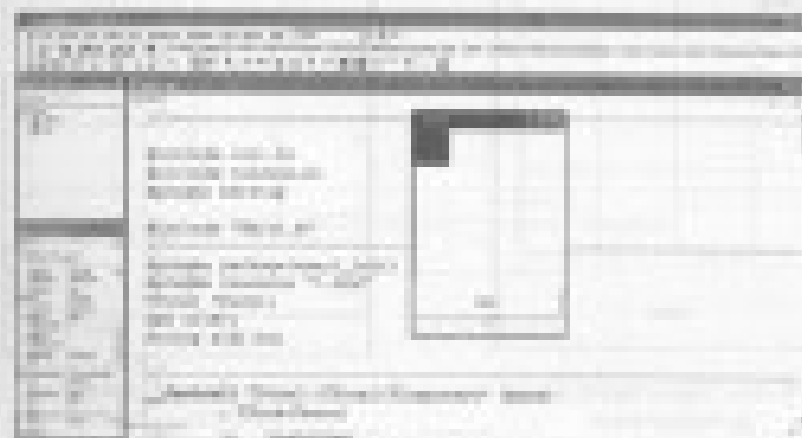


Fig. 4. Dimer

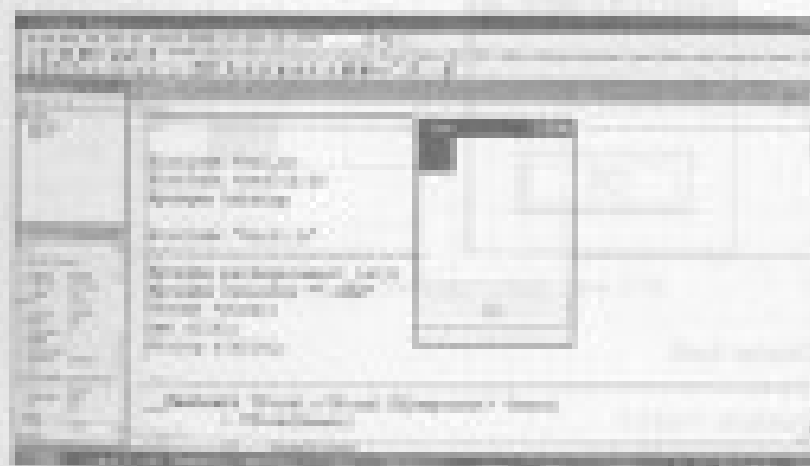


Fig. 5. Dimer (50.68-50.72-ram) - Dimer test results


```
void __fastcall TForm1::SGPaintCell(TObject *Sender, int ACol, int ARow)
```

```
{
    local ACatIndex;
```

```
if (ACol > ColOfACat[ARow + 1] or ACatIndex = 0)
    SG := ColOfACat[ARow + 1] - SG := ColOfACat[ARow];
```

```
SG := ColOfACat[ARow] + 1;
SG := ColOfACat[ARow] + 1;
SG := ColOfACat[ARow] + 1;
```

```
if (ARow = 0)
```

```
if (SG > ColOfACat[ARow] or SG = 0)
```

```
SG := ColOfACat[ARow] + 1; SG := ColOfACat[ARow];
```

```
SG := ColOfACat[ARow] + 1;
```

```
if (ACol = 0)
```

```
if (SG > ColOfACat[ACol] or SG = 0)
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1;
```

```
if (SG > ColOfACat[ACol] or SG = 0)
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1;
```

```
if (SG > ColOfACat[ACol] or SG = 0)
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1;
```

```
if (SG > ColOfACat[ACol] or SG = 0)
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1;
```

```
if (SG > ColOfACat[ACol] or SG = 0)
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1;
```

```
if (SG > ColOfACat[ACol] or SG = 0)
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1;
```

```
if (SG > ColOfACat[ACol] or SG = 0)
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1;
```

```
if (SG > ColOfACat[ACol] or SG = 0)
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1;
```

```
if (SG > ColOfACat[ACol] or SG = 0)
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1;
```

```
if (SG > ColOfACat[ACol] or SG = 0)
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1;
```

```
if (SG > ColOfACat[ACol] or SG = 0)
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
if (ACol > 0) (SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

```
SG := ColOfACat[ACol] + 1; SG := ColOfACat[ACol];
```

20-look for the number

1. Interfacing data/binding of the
2. QED for visual components
3. Visual components defined
4. Interfacing with system
5. Data local variables
6. Object-oriented system
7. Richard C. --- Richard C. --- Interfaced
8. Many using the language of the
9. Designing system
10. Project language of the
11. New form language of the
12. Custom language of the
13. Help (Richard C. --- Interfaced)
14. Designing system
15. Interface (IT) system of the
16. Components defined of the
17. Language of the system of the
18. Designing system of the

18. Kuchkari mantiqiy tizim.
19. Kuch mantiqiy tizim.
20. Virtual va mavjud komponentlar.
21. Uchi 1.6 hamma shartlari.
22. Komponentlarga propertiyalar yaratish.
23. Komponenta modulyati.
24. TTab va TTabal komponentlarning o'zlashdigi va ishlari.
25. StackBox va RadioButton komponentlarning o'zlashdigi va ishlari.
26. TGroupBox va TRadioGroup komponentlarning o'zlashdigi va ishlari.
27. TPanel va TImagePanel komponentlarning o'zlashdigi va ishlari.
28. TImage va TImage komponentlarning o'zlashdigi va ishlari.
29. TImage va TImage komponentlarning o'zlashdigi va ishlari.
30. Image yaratish tizimi.
31. OnClick roqam.
32. TPageControl komponentlarning modulyati.
33. TTab va TTabal komponentlarning o'zlashdigi va ishlari.
34. TTabal va TImageList komponentlarning o'zlashdigi va ishlari.
35. TGroupBox va TRadioGroup komponentlarning o'zlashdigi va ishlari.
36. TImage komponentlarning ishlatilishi ishlari.
37. TImage komponentlarning ishlatilishi ishlari.
38. OnClick roqam.
39. OnClick roqam.
40. TTab va TTabal komponentlarning o'zlashdigi va ishlari.
41. TImageList va TImageList komponentlarning o'zlashdigi va ishlari.
42. TImage komponentlarning modulyati.
43. TImage komponentlarning ishlatilishi.
44. TImageList va TImageList komponentlarning ishlatilishi ishlari.
45. TImage komponentlarning ishlatilishi ishlari.
46. Options yaratish.
47. Picture object yaratish.
48. Picture va Picture object yaratish.
49. TImageList va TImageList komponentlarning o'zlashdigi va ishlari.
50. TImage komponentlarning ishlatilishi.
51. TImage komponentlarning ishlatilishi va ishlari.

52. TImageList va TImageList komponentlarning ishlatilishi va ishlari.
53. TImage komponentlarning ishlatilishi ishlari.
54. Options yaratish.
55. Picture object yaratish.
56. OnClick roqam.
57. OnClick roqam.
58. OnClick roqam.
59. TImageList va TImageList komponentlarning o'zlashdigi va ishlari.
60. TImageList va TImageList komponentlarning o'zlashdigi va ishlari.
61. TImage komponentlarning ishlatilishi ishlari.
62. Options yaratish.
63. Picture object yaratish.

FOYDALANILGAN ADABIYOTLAR

Asosiy adabiyotlar

1. Кривоноз Е., Ротен Д. Язык программирования С#. М.: Финансы и статистика, 1995.
2. Льюс Д. С++C++. Справочник. М.: Наука, 1997.
3. Грейс Бил. Объектно-ориентированный анализ и проектирование С примерами приложений на С++. Новый дизайн, 2004г., 500.
4. Грейс Б. Объектно-ориентированные методы. Принципы и практика. Москва, 2004, 879 с.
5. Пашков ГС. Объектно-ориентированное программирование. Учебник. МЕТУ им. Баумана, 2003, 120 с.
6. Андрица Н.А. Основы программирования на языке С#. С++. Учебный курс. М.: 2002.
7. Шмидский И.Е. Программирование на языке С++. Справочник. Учебник для бакалавров. М.: Дашкунта, 2004, 160 с.
8. Справочник Е. Язык программирования С++. Третье издание. М.: Наука, 1999.
9. Дэй Джон. Объектно-ориентированное программирование на С++. Второе издание. – М.: Наука, 1999.

Qo'shimcha adabiyotlar

1. Кривоноз А.Е. Язык С++ Петер, 2003, 234 с.
2. Майерс Г. Наилучшее эффективное использование С++. 25 правил реализации. ДМК-Пресс, 2000, 304 с.
3. Мейсманов Д.В. Справочник по Visual C++, СПб, 2004.
4. Льюис Дев. С+++. Инновационная парадигма, СПб: Петер, 1999.

5. Льюис Д. Новый стандарт языка С++. 10 минут на урок. Петер, Москва, 2004, 234 с.
6. Шмидский Г. Справочник С++. Второе издание, СПб: БИТ, 1998.
7. Шмидский Д.В. Язык С++ – М.: Финансы и статистика, 1999.
8. Фейсман Г. Объектно-ориентированное программирование на С++ 4.5. – Киев: Дашкунта, 1996.
9. Шмидский Г. Теория и практика С++, СПб: БИТ, 1996.

MUNDRIA

1000

FORMAL C-TILING STRUCTURE AND FASTENERS

LOG TO 15:50 (1970 AM)

1.1. Aichele vs. Kienast'scher Ansatz	1
1.2. Otagaki-Müller	4
1.3. Kienast-Keller	8
1.4. Amsler	11
1.5. Burke (ohne Modell)	16
1.6. C. Elie de Beaune (Modell)	18
1.7. Mathematisches Modell von Elie de Beaune	19
1.8.1. Aichele vs. Kienast'scher Ansatz	24
1.8.2. Aichele vs. Kienast'scher Ansatz	25

3.4.4. **CONCENTRATION OF PARTICLES** 36

2.0. Operatorlar turlari	26
2.1. Unari operatorlari	27
2.2. Siki operatorlari	30
2.3. Oraliq operatorlari	33
2.4. Foydalanuvchi funktsiyalari	33
2.5. Paketiqlar	34
2.6. Nazariyati	44

2. Ask for your name	45
3. Ask for your name for help	46

Book 1049592-AR-VA-STATE-AR-17

1.1. Bio etichetare produselor	47
1.2. Certificarea produselor	48
1.3. Pierdere calitatii produselor	58
1.4. Noile metode de etichetare	62
1.5. Etichetare si sigilare	64
Etichetare si sigilare manuala	66
Etichetare si sigilare mecanizata	67

ACKNOWLEDGMENTS

4.1. Struktur va'zif	64
4.2. Struktur va'zif mas'uliyati	71
4.3. Struktur mas'uliyati	74
4.4. Abstrakt tushuncha mas'uliyati	76
4.5. Strukturning boshqa strukturadan tashkil topishi	81
4.6. Strukturalar	83
a) Abstrakt tushuncha mas'uliyati	87
b) Abstrakt tushuncha mas'uliyati	88

1. *Journal of Interpersonal Violence* 25(12):2391-2406, 2010. © 2010 Sage Publications

1.1. Karmachshlar	80
1.2. Karmachshlarning amaliyoti	80
1.3. Karmachshlar va amaliyot	84
1.4. O'qish amaliyoti	86
1.5. Fikrlash amaliyoti	88
1.6. Tashkilot amaliyoti	88
1.7. Tashkilot amaliyoti	88
1.8. Tashkilot amaliyoti	88
1.9. Tashkilot amaliyoti	88
1.10. Tashkilot amaliyoti	88
1.11. Tashkilot amaliyoti	88
1.12. Tashkilot amaliyoti	88
1.13. Tashkilot amaliyoti	88
1.14. Tashkilot amaliyoti	88
1.15. Tashkilot amaliyoti	88
1.16. Tashkilot amaliyoti	88
1.17. Tashkilot amaliyoti	88
1.18. Tashkilot amaliyoti	88
1.19. Tashkilot amaliyoti	88
1.20. Tashkilot amaliyoti	88
1.21. Tashkilot amaliyoti	88
1.22. Tashkilot amaliyoti	88
1.23. Tashkilot amaliyoti	88
1.24. Tashkilot amaliyoti	88
1.25. Tashkilot amaliyoti	88
1.26. Tashkilot amaliyoti	88
1.27. Tashkilot amaliyoti	88
1.28. Tashkilot amaliyoti	88
1.29. Tashkilot amaliyoti	88
1.30. Tashkilot amaliyoti	88
1.31. Tashkilot amaliyoti	88
1.32. Tashkilot amaliyoti	88
1.33. Tashkilot amaliyoti	88
1.34. Tashkilot amaliyoti	88
1.35. Tashkilot amaliyoti	88
1.36. Tashkilot amaliyoti	88
1.37. Tashkilot amaliyoti	88
1.38. Tashkilot amaliyoti	88
1.39. Tashkilot amaliyoti	88
1.40. Tashkilot amaliyoti	88
1.41. Tashkilot amaliyoti	88
1.42. Tashkilot amaliyoti	88
1.43. Tashkilot amaliyoti	88
1.44. Tashkilot amaliyoti	88
1.45. Tashkilot amaliyoti	88
1.46. Tashkilot amaliyoti	88
1.47. Tashkilot amaliyoti	88
1.48. Tashkilot amaliyoti	88
1.49. Tashkilot amaliyoti	88
1.50. Tashkilot amaliyoti	88
1.51. Tashkilot amaliyoti	88
1.52. Tashkilot amaliyoti	88
1.53. Tashkilot amaliyoti	88
1.54. Tashkilot amaliyoti	88
1.55. Tashkilot amaliyoti	88
1.56. Tashkilot amaliyoti	88
1.57. Tashkilot amaliyoti	88
1.58. Tashkilot amaliyoti	88
1.59. Tashkilot amaliyoti	88
1.60. Tashkilot amaliyoti	88
1.61. Tashkilot amaliyoti	88
1.62. Tashkilot amaliyoti	88
1.63. Tashkilot amaliyoti	88
1.64. Tashkilot amaliyoti	88
1.65. Tashkilot amaliyoti	88
1.66. Tashkilot amaliyoti	88
1.67. Tashkilot amaliyoti	88
1.68. Tashkilot amaliyoti	88
1.69. Tashkilot amaliyoti	88
1.70. Tashkilot amaliyoti	88
1.71. Tashkilot amaliyoti	88
1.72. Tashkilot amaliyoti	88
1.73. Tashkilot amaliyoti	88
1.74. Tashkilot amaliyoti	88
1.75. Tashkilot amaliyoti	88
1.76. Tashkilot amaliyoti	88
1.77. Tashkilot amaliyoti	88
1.78. Tashkilot amaliyoti	88
1.79. Tashkilot amaliyoti	88
1.80. Tashkilot amaliyoti	88
1.81. Tashkilot amaliyoti	88
1.82. Tashkilot amaliyoti	88
1.83. Tashkilot amaliyoti	88
1.84. Tashkilot amaliyoti	88
1.85. Tashkilot amaliyoti	88
1.86. Tashkilot amaliyoti	88
1.87. Tashkilot amaliyoti	88
1.88. Tashkilot amaliyoti	88
1.89. Tashkilot amaliyoti	88
1.90. Tashkilot amaliyoti	88
1.91. Tashkilot amaliyoti	88
1.92. Tashkilot amaliyoti	88
1.93. Tashkilot amaliyoti	88
1.94. Tashkilot amaliyoti	88
1.95. Tashkilot amaliyoti	88
1.96. Tashkilot amaliyoti	88
1.97. Tashkilot amaliyoti	88
1.98. Tashkilot amaliyoti	88
1.99. Tashkilot amaliyoti	88
2.00. Tashkilot amaliyoti	88

NAME: _____ PAGE: _____

6.1. Fəqarlar.....	108
6.2. Fəqarlar arasında münasibət qılmaq.....	110
6.3. Fəqarlar kəlliyorlu münasibət qılmaq.....	114
6.4. Qayı dərəcədəki kəlliyorlu və dərəcəli.....	115
6.4.1. Qayı dərəcəli kəlliyorlu.....	117
6.4.2. Qayı dərəcəli kəlliyorlu.....	118

Table 1. *Physocarpus opulifolius* L. var. *maximowiczii* (Rupr.) Maxim. 139

1.1. Propagator funktsiyasi	129
1.2. Makroliar	134
1.3. Korr. funksiyasi	136
1.4. Halk funktsiya parametrlari	140
1.5. Parametrlar va o'zgaruvchi bo'lgan funktsiyalar	145
1.5.1. O'zgaruvchi makroliar	145
1.5.2. O'zgaruvchi korr. funksiyalar	148

NAME: _____

B.1. Matematik Fonksiyonlar	148
B.2. Simetrik Fonksiyonlar	154
B.3. Sıra Fonksiyonlar	158
B.4. Sıra ve sıra elemanları Fonksiyonları	164
B.4.1. Sıra elemanları	164
B.4.2. Sıra elemanları	164

Page 10 of 10

0.1. Nazariy asoslar (shaxsiy hisobotlar)	270
0.2. Nazariy asoslar va usullar	270
0.3. Nazariy asoslar (shaxsiy hisobotlar)	274
0.4. Ishlar va tashkilot	277
0.5. Nazariy asoslar (shaxsiy hisobotlar)	280

8.5. Univariate Analysis	183
8.6. Bivariate Analysis	190
8.7. Multivariate Analysis	194

JOURNAL OF ENVIRONMENT & DEVELOPMENT 17(4) 379-396

10.1. Myriam	144
10.2. Universal sets: no number	191
10.3. Universal set	164
10.3.1. for vector spaces	190
10.3.2. for vector spaces for order	190

© 2004, E-44 TELUKA, ORYON, FLE DUSTISLAMA

THEORY OF THE MIND 101

11.1. Objektele matematicei generale sunt:	111
11.2. Obiectele matematice generale sunt: algebră, analiză și geometrie:	114
11.3. Inegalitatea triunghiului:	116
11.4. Formula inegalității lui...	118
11.5. Teorema:	120
11.6. Definiția:	122

[Page: THE ANCHOR] 19

11.1. Organismuslar va koronaviruslar	131
11.2. Anzilar	131
11.3. Dastur shakllantirilgan	134
11.4. Epizootlar	134
11.5. Psixologik va fiziologik	138
11.6. Mikrobyalar va viruslar	144
11.7. Funksionallik	144
11.8. Funksionallik	144
11.9. Funksionallik	144
11.10. Funksionallik	144
11.11. Funksionallik	144
11.12. Funksionallik	144
11.13. Funksionallik	144
11.14. Funksionallik	144
11.15. Funksionallik	144
11.16. Funksionallik	144
11.17. Funksionallik	144
11.18. Funksionallik	144
11.19. Funksionallik	144
11.20. Funksionallik	144
11.21. Funksionallik	144
11.22. Funksionallik	144
11.23. Funksionallik	144
11.24. Funksionallik	144
11.25. Funksionallik	144
11.26. Funksionallik	144
11.27. Funksionallik	144
11.28. Funksionallik	144
11.29. Funksionallik	144
11.30. Funksionallik	144
11.31. Funksionallik	144
11.32. Funksionallik	144
11.33. Funksionallik	144
11.34. Funksionallik	144
11.35. Funksionallik	144
11.36. Funksionallik	144
11.37. Funksionallik	144
11.38. Funksionallik	144
11.39. Funksionallik	144
11.40. Funksionallik	144
11.41. Funksionallik	144
11.42. Funksionallik	144
11.43. Funksionallik	144
11.44. Funksionallik	144
11.45. Funksionallik	144
11.46. Funksionallik	144
11.47. Funksionallik	144
11.48. Funksionallik	144
11.49. Funksionallik	144
11.50. Funksionallik	144
11.51. Funksionallik	144
11.52. Funksionallik	144
11.53. Funksionallik	144
11.54. Funksionallik	144
11.55. Funksionallik	144
11.56. Funksionallik	144
11.57. Funksionallik	144
11.58. Funksionallik	144
11.59. Funksionallik	144
11.60. Funksionallik	144
11.61. Funksionallik	144
11.62. Funksionallik	144
11.63. Funksionallik	144
11.64. Funksionallik	144
11.65. Funksionallik	144
11.66. Funksionallik	144
11.67. Funksionallik	144
11.68. Funksionallik	144
11.69. Funksionallik	144
11.70. Funksionallik	144
11.71. Funksionallik	144
11.72. Funksionallik	144
11.73. Funksionallik	144
11.74. Funksionallik	144
11.75. Funksionallik	144
11.76. Funksionallik	144
11.77. Funksionallik	144
11.78. Funksionallik	144
11.79. Funksionallik	144
11.80. Funksionallik	144
11.81. Funksionallik	144
11.82. Funksionallik	144
11.83. Funksionallik	144
11.84. Funksionallik	144
11.85. Funksionallik	144
11.86. Funksionallik	144
11.87. Funksionallik	144
11.88. Funksionallik	144
11.89. Funksionallik	144
11.90. Funksionallik	144
11.91. Funksionallik	144
11.92. Funksionallik	144
11.93. Funksionallik	144
11.94. Funksionallik	144
11.95. Funksionallik	144
11.96. Funksionallik	144
11.97. Funksionallik	144
11.98. Funksionallik	144
11.99. Funksionallik	144
12.00. Funksionallik	144

DR. A. NABROV, B. A. KURBULOV
M. R. DABDILDOV, Q. S. RAVILBAYEV

C VA C++ TILI

«O'zbekiston Respublikasi Oliy ta'lim vazirligi va Oliy ta'lim institutlari
Boshqaruvi, Dushanbe, 1999-yil, 10.10.1999. Boshqaruvi va Oliy ta'lim institutlari
kuchli bilan o'qitish va o'qitish uchun ishlatiladi»

«O'zbekiston Respublikasi

«O'zbekiston

2000

«O'zbekiston Respublikasi

«O'zbekiston Respublikasi

«O'zbekiston Respublikasi

«O'zbekiston Respublikasi

«O'zbekiston Respublikasi

«O'zbekiston Respublikasi

«O'zbekiston Respublikasi

«O'zbekiston Respublikasi

«O'zbekiston Respublikasi

