

**O‘ZBEKISTON RESPUBLIKASI OLIY VA O‘RTA MAXSUS TA’LIM
VAZIRLIGI**

TOSHKENT DAVLAT IQTISODIYOT UNIVERSITETI

Sobirov A.A., Dadabaeva R.A., Xashimova D.P.

ALGORITMLASHTIRISH VA DASTURLASH ASOSLARI

*O‘zbekiston Respublikasi Oliy va o‘rta maxsus ta’lim vazirligi huzuridagi
Muvofiqlashtiruvchi kengash tomonidan 5330200 - Informatika va axborot
texnologiyalari (iqtisodiyot) va 5232600 - Biznes informatika bakalavriat ta’lim
yo‘nalishlarida tahsil olayotgan talabalar uchun o‘quv qo‘llanma sifatida tavsiya
etilgan*

TOSHKENT – IQTISODIYOT – 2019

UO‘K: 347.214.21(075)

KBK 67.404.1

K 98

Yo-18

Sobirov A.A., Dadabayeva R.A., Xashimova D.P. Algoritm lashtirish va dasturlash asoslari. O‘quv qo‘llanma. - T.: «IQTISODIYOT», 2019. - 132 bet.

Mazkur o‘quv qo‘llanmada iqtisodiy masalalarni yechishda algoritm lashtirish va dasturlashning asosiy tamoyillari va namunaviy yo‘llari ochib berilgan. Qo‘llanmada hisoblash algoritm larning qabul qilingan tasnifi yoritilgan, turli amaliy masalalar uchun algoritm va dasturlarni tuzish misollari keltirilgan bo‘lib, u 5330200 - Informatika va axborot texnologiyalari (iqtisodiyot) va 5232600 - Biznes informatika bakalavriat ta‘lim yo‘nalishlarining “Algoritm lashtirish va dasturlash asoslari” fanidan o‘quv qo‘llanma sifatida tavsiya etiladi.

В данном учебном пособии рассматриваются основные принципы и типовые приемы алгоритмизации и программирования при решении экономических задач. В пособии описываются принятая классификация вычислительных алгоритмов, приводятся примеры составления алгоритмов и программ для различных прикладных задач, и рекомендуется по предмету «Основы алгоритмизации и программирования» направлений образования бакалавриата 5330200 - «Информатика и информационные технологии (экономика)» и 5232600- Бизнес информатика, в качестве учебного пособия.

This textbook describes the basic principles and typical techniques of algorithmization and programming in solving economic problems. The textbook describes the accepted classification of computational algorithms, provides examples of compiling algorithms and programs for various applied problems, and is recommended in the subject "Fundamentals of Algorithmization and Programming" in the bachelor's degrees 5330200 - "Computer Science and Information Technology (Economics)" and 5232600 - Business Computer Science, in as a training manual.

ISBN: 978-9943-05-691-6

UO‘K: 347.214.21(075)

KBK 67.404.1

K 98

Yo-18

© “IQTISODIYOT” nashriyoti, 2019

© Sobirov A.A., Dadabaeva R.A.Xashimova D.P., 2019

MUNDARIJA

KIRISH	13
1-bob. FANGA KIRISH	16
1.1. «Algoritm lashtirish va dasturlash asoslari» fanini ta`rifi va mohiyati, o`qitish maqsadi	16
1.2. Algoritm tushunchasi va xususiyatlari	17
1.3 Masalalarni kompyuterda yechish bosqichlari. Masala qo`yilishi	20
1.4 Masalani matematik ifodalash. Sonli usulni tanlash	20
1.5 Hisoblash jarayonini algoritm lashtirish	21
1.6 Dastur tuzish. Dasturni tahrirlash	23
1.7 Masalani shaxsiy kompyuterda yechish va natijalarni tahlil qilish	25
2-bob. ALGORITMLARNI AKS ETTIRISH USULLARI	27
2.1. Algoritm larni aks ettirish usullari. Algoritmni oddiy tilda ifodalash	27
2.2. Algoritmni blok-chizmalar ko`rinishida ifodalash	28
2.3. Algoritmni maxsus (algoritmik) tilda yozish	33
3- bob. HISOBLASH JARAYONLARINING TURLARI. CHIZIQLI HISOBLASH JARAYONLARI	35
3.1. Hisoblash jarayonlarining turlari	35
3.2. Chiziqli hisoblash jarayoni	37
3.3. Chiziqli hisoblash jarayonlariga misollar	38
4- bob. TARMOQLANGAN HISOBLASH JARAYONLARI	41
4.1. Tarmoqlangan hisoblash jarayoni	41
4.2. Oddiy tarmoqlangan hisoblash jarayonining algoritmi	42
4.3. Murakkab tarmoqlangan hisoblash jarayonining algoritmi	43
5- bob. TAKRORLANUVCHI HISOBLASH JARAYONLAR	50

5.1.	Takrorlanuvchi (siklik) hisoblash jarayonlari	50
5.2	Vektor elementlari qiymatini yig'indisini topish masalasining blok-chizmasi	53
5.3.	Aniq berilgan son asosida oddiy sikllarni tashkil etish blok-chizmalari	55
5.4.	Takrorlanishlar soni aniq bo'lmagan sikllar	61
5.5	Matritsalar (ikki o'lchamli massivlar) tushunchasi	63
5.6	Matritsaning elementlar yig'indisini topish algoritmini tuzish	64
5.7	Kvadrat matritsa elementlarini qayta ishlashning variantlari	66
6- bob.	MA'LUMOTLARNI SARALASH ALGORIMLARI	70
6.1.	Ma'lumotlarni saralashning mohiyati	70
6.2.	Oddiy tanlov saralash metodi	71
6.3.	Oddiy almashuv saralash metodi	73
6.4.	Qo'shni elementlar juftliklarini izchil tartibga solinishi	75
7- bob.	ALGORITMIK DASTURLASH TILLARINING UMUMIY TA'RIFI	78
7.1.	Dasturlash tillari va ularning tasniflanisi	78
7.2.	Paskal tilining umumiy ta'rifi	80
7.3.	Paskal algoritmik tilining alifbosi	81
7.4.	Paskal tilida ma'lumotlar turlari	82
7.5.	Paskal algoritmik tilida ifodalar	83
7.6	Paskal tilida yozilgan dasturning tuzilishi	84
8- bob.	DASTURLASH TILINING SODDA OPERATORLARI	88
8.1.	Paskal tilidagi operatorlar	88
8.2.	O'zlashtirish operatori	89
8.3.	Ma'lumotlarni kiritish va chiqarish operatorlari	89
9- bob.	TARMOQLANGAN HISOBLASH JARAYONLARNI DASTURLASH	93
9.1	Tarmoqlanuvchi hisoblash jarayonlarini dasturlash	93

9.2	Shartsiz o‘tish operatori	93
9.3.	Shartli o‘tish operatori	94
9.4.	Tanlash operatori	96
10- bob.	SODDA TAKRORLANUVCHI HISOBLASH JARAYONLARNI DASTURLASH	100
10.1.	Takrorlanuvchi jarayonlarini tashkil qilish	100
10.2.	Sharti avval tekshiriladigan takrorlanish jarayoni	101
10.3.	Sharti keyin tekshiriladigan takrorlanish jarayoni	103
10.4.	Parametrli takrorlanish jarayoni	105
11- bob.	MURAKKAB TAKRORLANUVCHI HISOBLASH JARAYONLARNI DASTURLASH	107
11.1.	Murakkab takrorlanuvchi hisoblash jarayonining tushunchasi	107
11.2.	Ma’lumotlar massivi haqida tushuncha	109
11.3.	Massiv elementlarini saralash bo‘yicha dasturlarni tuzish	112
11.4.	Massivda eng kichik elementni izlash algoritmining dasturi	115
12- bob.	PROTSEDURA VA FUNKSIYALAR	117
12.1.	Paskal tilida protsedura konstruksiyasi	117
12.2.	Funksiya qism dasturi haqida	119
12.3.	Protsedura va funksiya qism dasturiga murojaat qilish	120
12.4.	Protsedura va funksiya qism dasturlarining parametrlari	121
12.5	Ayrim standart protsedura va funksiyalar	122
	GLOSSARIY	124
	FOYDALANILGAN ADABIYOTLAR RO`YXATI	129

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	13
Глава-1. ВВЕДЕНИЕ В ПРЕДМЕТ	16
1.1. Характеристика и сущность, цель изучения дисциплины «Основы алгоритмизации и программирования»	16
1.2. Понятие и свойства алгоритма	17
1.3. Этапы решения задач на компьютере. Постановка задачи	20
1.4. Математическое отображение задачи. Выбор числового метода	20
1.5. Алгоритмизация вычислительного процесса	21
1.6. Разработка программы. Редактирование программы	23
1.7. Решение задачи на персональном компьютере и анализ результатов	25
Глава-2. СПОСОБЫ ОТОБРАЖЕНИЯ АЛГОРИТМОВ	27
2.1. Способы отображения алгоритмов. Запись алгоритма на естественном языке	27
2.2. Описание алгоритмов с помощью блок-схем	28
2.3. Запись алгоритма на специальном (алгоритмическом) языке	33
Глава-3. ТИПЫ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ. ЛИНЕЙНЫЙ ВЫЧИСЛИТЕЛЬНЫЙ ПРОЦЕСС	35
3.1. Типы вычислительных процессов	35
3.2. Линейный вычислительный процесс	37
3.3. Примеры линейных вычислительных процессов	38
Глава-4. РАЗВЕТВЛЁННЫЙ ВЫЧИСЛИТЕЛЬНЫЙ ПРОЦЕСС	41
4.1. Разветвлённый вычислительный процесс	41
4.2. Алгоритм простого разветвлённого вычислительного процесса	42
4.3. Алгоритм сложного разветвлённого вычислительного процесса	43
Глава-5. ПОВТОРЯЮЩИЙСЯ ВЫЧИСЛИТЕЛЬНЫЙ ПРОЦЕСС	50

5.1.	Повторяющиеся (циклические) вычислительные процессы	50
5.2	Блок-схема задачи нахождения суммы значений элементов вектора	53
5.3.	Блок-схемы организации простых циклов для точно заданных данных	55
5.4.	Циклы с неизвестным числом повторений	61
5.5	Понятие матрицы (двухмерного массива)	63
5.6	Составление алгоритма нахождения суммы элементов матрицы	64
5.7	Варианты обработки элементов квадратной матрицы	66
Глава-6.	АЛГОРИТМЫ СОРТИРОВКИ ДАННЫХ	70
6.1.	Сущность сортировки данных	70
6.2.	Метод сортировки простым выбором	71
6.3.	Метод сортировки простого обмена	73
6.4.	Последовательное упорядочение пар смежных элементов	75
Глава-7.	ОБЩАЯ ХАРАКТЕРИСТИКА АЛГОРИТМИЧЕСКИХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ	78
7.1.	Языки программирования и их классификация	78
7.2.	Общая характеристика языка Паскаль	80
7.3.	Алфавит языка Паскаль	81
7.4.	Типы данных в языке Паскаль	82
7.5.	Выражения в алгоритмическом языке Паскаль	83
7.6	Структура программы, написанной на языке Паскаль	84
Глава-8.	ПРОСТЫЕ ОПЕРАТОРЫ ЯЗЫКА ПРОГРАММИРОВАНИЯ	88
8.1.	Операторы языка Паскаль	88
8.2.	Оператор присваивания	89
8.3.	Операторы ввода и вывода данных	89
Глава-9.	ПРОГРАММИРОВАНИЕ РАЗВЕТВЛЁННЫХ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ	93

9.1	Программирование разветвлённых вычислительных процессов	93
9.2	Оператор безусловного перехода	93
9.3.	Оператор условного перехода	94
9.4.	Оператор выбора	96
Глава-10.	ПРОГРАММИРОВАНИЕ ПРОСТЫХ ПОВТОРЯЮЩИХСЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ	100
10.1.	Организация повторяющихся процессов	100
10.2.	Повторяющиеся процессы с предварительной проверкой условия	101
10.3.	Повторяющиеся процессы с последующей проверкой условия	103
10.4.	Повторяющиеся процессы с параметром	105
Глава-11.	ПРОГРАММИРОВАНИЕ СЛОЖНЫХ ПОВТОРЯЮЩИХСЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ	107
11.1.	Понятие сложного повторяющегося вычислительного процесса	107
11.2.	Понятие массива данных	109
11.3.	Разработка программ сортировки элементов массива	112
11.4.	Программа алгоритма поиска самого наименьшего элемента в массиве	115
Глава-12.	ПРОЦЕДУРЫ И ФУНКЦИИ	117
12.1.	Конструкция процедуры в языке Паскаль	117
12.2.	Подпрограмма функция	119
12.3.	Вызов процедуры и функции	120
12.4.	Параметры подпрограмм процедура и функция	121
12.5	Примеры стандартных процедур и функций	122
	ГЛОССАРИЙ	124
	СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	129

CONTENT

INTRODUCTION	13
Chapter -1. INTRODUCTION TO THE SUBJECT	16
1.1. Description and essence, the purpose of studying the discipline "Fundamentals of Algorithmization and Programming"	16
1.2. The concept and properties of the algorithm	17
1.3. Stages of solving problems on the computer. Formulation of the problem	20
1.4. The mathematical representation of the problem. Numerical method selection	20
1.5. Computational process algorithmization	21
1.6. Program development. Editing a program	23
1.7. Solving a problem on a personal computer and analyzing the results	25
Chapter -2. ALGORITHM DESCRIPTION METHODS	27
2.1. Algorithm description methods. Natural language algorithm recording	27
2.2. Description of algorithms using flowcharts	28
2.3. Writing an algorithm in a special (algorithmic) language	33
Chapter-3. TYPES OF COMPUTING PROCESSES. LINEAR COMPUTING PROCESS	35
3.1. Types of computing processes	35
3.2. Linear computing process	37
3.3. Linear computing process examples	38
Chapter-4. BRANCHED COMPUTER PROCESS	41
4.1. Branched computer process	41
4.2. Simple branched computing process algorithm	42
4.3. Complex branched computing process algorithm	43
Chapter-5. REPEATED COMPUTER PROCESS	50

5.1.	Repeated (cyclic) computing processes	50
5.2	The flowchart of the problem of finding the sum of the values of the elements of the vector	53
5.3.	Flowcharts for organizing simple loops for precisely defined data	55
5.4.	Cycles with an unknown number of repetitions	61
5.5	The concept of a matrix (two-dimensional array)	63
5.6	Drawing up an algorithm for finding the sum of matrix elements	64
5.7	Processing options for square matrix elements	66
Chapter-6.	DATA SORT ALGORITHMS	70
6.1.	The essence of data sorting	70
6.2.	Simple selection sorting method	71
6.3.	Simple exchange sort method	73
6.4.	Sequential ordering of pairs of adjacent elements	75
Chapter-7.	GENERAL CHARACTERISTICS OF ALGORITHMIC PROGRAMMING LANGUAGES	78
7.1.	Programming languages and their classification	78
7.2.	General characteristics of the Pascal language	80
7.3.	Pascal language Alphabet	81
7.4.	Pascal language data types	82
7.5.	Expressions in the algorithmic language Pascal	83
7.6	The structure of a program written in Pascal	84
Chapter-8.	SIMPLE PROGRAMMING LANGUAGE OPERATORS	88
8.1.	Pascal language operators	88
8.2.	Assignment operator	89
8.3.	Data input and output operators	89
Chapter-9.	PROGRAMMING BRANCHED COMPUTER PROCESSES	93
9.1	Programming branched computer processes	93

9.2	Unconditional jump operator	93
9.3.	Conditional jump operator	94
9.4.	Select statement	96
Chapter-10.	PROGRAMMING SIMPLE REPEATED COMPUTER PROCESSES	100
10.1.	Organization of repetitive processes	100
10.2.	Repeating processes with pre-condition checking	101
10.3.	Repeating processes followed by condition checking	103
10.4.	Repeating processes with parameter	105
Chapter-11.	PROGRAMMING DIFFICULT REPEATED COMPUTER PROCESSES	107
11.1.	The concept of a complex repetitive computing process	107
11.2.	Concept of data array	109
11.3.	Development of programs for sorting array elements	112
11.4.	The algorithm for finding the smallest element in an array	115
Chapter-12.	PROCEDURES AND FUNCTIONS	117
12.1.	Pascal procedure design	117
12.2.	Subroutine function	119
12.3.	Call procedure and function	120
12.4.	Subroutine parameters procedure and function	121
12.5	Examples of standard procedures and functions	122
	GLOSSARY	124
	BIBLIOGRAPHY	129

KIRISH

Axborot-kommunikatsiya texnologiyalari rivojlanishining zamonaviy jahon darajasi shundayki, respublikada jahon axborot makonining infratuzilmalari va milliy axborot-hisoblash tarmog‘i integratsiyasiga mos keluvchi milliy tizimni yaratish milliy iqtisodiyot, boshqarish, fan va ta‘lim samaradorligining muhim omili bo‘lmoqda. Bu muammolar ancha murakkab va ayni paytda respublikamiz uchun dolzarbdir. Hozirda olib borilayotgan iqtisodiy, tuzilmaviy va boshqa o‘zgarishlarni amalga oshirish natijalari respublikada axborotlashtirish bilan bog‘liq muammolarning qanday va qaysi muddatlarda hal etishga ham bog‘liqdir.

Boshqaruv masalalarini yechishda zamonaviy axborot texnologiyalarining o‘rni nihoyatda ahamiyatga ega.

O‘zbekiston Respublikasi Prezidenti Shavkat Miromonovich Mirziyoyev o‘zining Oliy Majlisga Murojaatnomasida ta‘kidlaganidek, “Iqtisodiyotning barcha sohalarini raqamli texnologiyalar asosida yangilashni nazarda tutadigan Raqamli iqtisodiyot milliy konsepsiyasini ishlab chiqishimiz kerak. Shu asosda “Raqamli O‘zbekiston – 2030” dasturini hayotga tatbiq etishimiz zarur”¹.

Shuningdek, O‘zbekiston Respublikasi Prezidenti Shavkat Miromonovich Mirziyoyev o‘zining «Axborot texnologiyalari va kommunikatsiyalari sohasini yanada takomillashtirish chora-tadbirlari to‘g‘risida» 2018- yil 19- fevraldagi PF-5349-sonli Farmonida ta‘kidlaganidek, “Mamlakatda zamonaviy axborot texnologiyalari va kommunikatsiyalarini rivojlantirish, elektron davlat xizmatlarini ko‘rsatishning yaxlit tizimini yaratish, davlat organlarining aholi bilan muloqot qilishining yangi mexanizmlarini joriy etish yuzasidan izchil ish olib borilmoqda”².

Shuning uchun milliy iqtisodiyotning turli sohalari uchun dasturiy vositalarni ishlab chiqish ko‘nikmasiga ega mutaxassislarni tayyorlash katta ahamiyatga ega.

Dasturlashni o‘rganish va amaliy o‘zlashtirish jarayoni uch qismdan iborat:

¹ O‘zbekiston Respublikasi Prezidenti Shavkat Mirziyoyevning Oliy Majlisga Murojaatnomasi. //Xalq so‘zi, 2018-y., 29- dekabr.

² O‘zbekiston Respublikasi Prezidentining “Axborot texnologiyalari va kommunikatsiyalari sohasini yanada takomillashtirish chora-tadbirlari to‘g‘risida”gi 2018- yil 19- fevraldagi PF-5349-sonli Farmoni.

- algoritmlarni qurish usullarini oʻrganish;
- dasturlash koʻnikmasini shakllantirish;
- aniq dasturlash tilini oʻrganish va amaliy oʻzlashtirish.³

“Algoritmlashtirish va dasturlash asoslari” fani boʻyicha oʻquv qoʻllanmaning 1-7 boblari birinchi qismning masalalarini yechishga bagʻishlangan. Qoʻllanmaning 8-12 boblari talabalarda birlamchi dasturlash koʻnikmalarini shakllantirishga qaratilgan.

Ushbu qoʻllanmaning birinchi bobida «Algoritmlashtirish va dasturlash asoslari» fanini taʼrifi va mohiyati, oʻqitish maqsadi, algoritm tushunchasi va xususiyatlari hamda masalalarni kompyuterda yechish bosqichlari keltirilgan.

Ikkinchi bobda algoritmlarni aks ettirish usullari yoritilgan, xususan algoritmni oddiy tilda ifodalash, algoritmni blok-chizmalar koʻrinishida ifodalash va algoritmni maxsus (algoritmik) tilda yozish.

Uchinchi bobda hisoblash jarayonlarining turlari, chiziqli hisoblash jarayoni algoritmlari bayon etilgan va unga misollar keltirilgan.

4-bobda tarmoqlangan hisoblash jarayonlari, xususan oddiy va murakkab tarmoqlangan hisoblash jarayonlarining algoritmlari yoritilgan va ularga misollar keltirilgan.

Beshinchi bobda takrorlanuvchi hisoblash jarayonlar algoritmlari bayon etilgan va ularga misollar keltirilgan.

Oltinchi bobda maʼlumotlarni saralash algoritm(lari) turlari, xususan oddiy tanlov saralash metodi, oddiy almashuv saralash metodi, qoʻshni elementlar juftliklarini izchil tartibga solinishi metodi va ularga misollar keltirilgan.

Ettinchi bobda algoritmik dasturlash tillari va ularni tasniflashi, Paskal tilining umumiy taʼrifi, Paskal algoritmik tilining alifbosi, Paskal tilida maʼlumotlar turlari, Paskal algoritmik tilida ifodalar hamda Paskal tilida yozilgan dasturning tuzilishi keltirilgan.

³ Семакин И.Г., Шестаков А.П. Основы алгоритмизации и программирования Издательство: Академия. 2016 г. - С. 6.

Sakkizinchi bobda dasturlash tilining sodda operatorlari, xususan o'zlashtirish operatori, ma'lumotlarni kiritish va chiqarish operatorlarining formati, qo'llanish tartibi misollar bilan bayon etilgan.

To'qqizinchi bobda tarmoqlangan hisoblash jarayonlarni shartsiz o'tish, shartli o'tish va tanlash operatorlari yordamida dasturlash qoidalari va tartibi yoritilgan. Operatorlarni qo'llash bo'yicha misollar keltirilgan.

O'ninchi bobda sodda takrorlanuvchi hisoblash jarayonlarni, xususan sharti avval tekshiriladigan takrorlanish jarayoni, sharti keyin tekshiriladigan takrorlanish jarayoni va parametrli takrorlanish jarayoni dasturlash qoidalari va tartibi yoritilgan va ularga misollar keltirilgan.

O'n birinchi bobda murakkab takrorlanuvchi hisoblash jarayonlarni dasturlash jarayoni ko'rib chiqilgan, xususan bir o'lchamli va ko'p o'lchamli massivlardan foydalangan holda dasturlarni tuzish hamda massiv elementlarini saralash bo'yicha dasturlarni tuzish qoidalari va tartibi yoritilgan va ularga misollar keltirilgan.

O'n ikkinchi bobda Paskal tilida protsedura konstruksiyasi, Paskalda funksiya tushunchasi, protsedura va funksiya qism dasturiga murojaat qilish, protsedura va funksiya qism dasturlarining parametrlari va ayrim standart protsedura va funktsiyalar keltirilgan.

1-bob. FANGA KIRISH

1.1. «Algoritmlashtirish va dasturlash asoslari» fanini ta`rifi va mohiyati, o`qitish maqsadi.

1.2. Algoritm tushunchasi va xususiyatlari.

1.3. Masalalarni kompyuterda yechish bosqichlari. Masala qo`yilishi.

1.4. Masalani matematik ifodalash. Sonli usulni tanlash.

1.5. Hisoblash jarayonini algoritmlashtirish.

1.6. Dastur tuzish. Dasturni tahrirlash.

1.7. Masalani shaxsiy kompyuterda yechish va natijalarni tahlil qilish.

1.1. «Algoritmlashtirish va dasturlash asoslari» fanini ta`rifi va mohiyati, o`qitish maqsadi

Insoniyat tarixining ko`p asrlik tajribasi ezgu g`oyalar, sog`lom mafkura va zamonaviy bilimlardan mahrum har qanday jamiyat tanazzulga yuz tutishini ko`rsatgan. Shuning uchun bizning mamlakatimiz ham o`z oldiga ozod va obod Vatan, demokratik jamiyat barpo etish, erkin va farovon davlat qurish, rivojlangan davlatlar qatorida borish maqsadini qo`ygan.

O`sib kelayotgan yoshlarni kelajak jamiyatning faol quruvchilari bo`lishi uchun ularni fan va texnikaning eng ilg`or va zamonaviy yutuqlari bilan qurollantirish hamda olgan bilimlarini amaliyotda qo`llashga o`rgatish talab qilinadi. Bu holat ayniqsa, kompyuter bilan bog`liq iqtisodiy, siyosiy va amaliy masalalarni yechishda yaqqol o`z aksini topadi.

Demak, yoshlarni zamonaviy kompyuterlar bilan ishlash, milliy iqtisodiyotning turli masalalarini yechishga mo`ljallangan dasturiy ta`minot bilan muloqot qilishga o`rgatishdan tashqari, yangi masalalar uchun dasturiy ta`minot ishlab chiqish yo`l-yo`llari bilan ham tanishtirish lozim bo`ladi.

Shuning uchun iqtisodiy masalalarni yechish uchun dasturiy vositalarni ishlab chiqish ko`nikmasiga ega mutaxassislarni tayyorlash katta ahamiyatga ega.

“Algoritmlashtirish va dasturlash asoslari” fanini o‘qitishdan asosiy maqsadi bo‘lib talabalarni iqtisodiy masalalarni yechish jarayonining algoritmlarini ishlab chiqishning metod va qoidalari bo‘yicha nazariy bilimlarni berish hamda turli hisoblash jarayonlarining algoritmlarini tuzish va ularni amalga oshirish dasturlarini ishlab chiqishning amaliy ko‘nikmalarini shakllantirish hisoblanadi.

Fanni o‘qitishning asosiy vazifalari quyidagilardan iboratdir:

- mavjud standartlar asosida hisoblash jarayonlarining grafik chizmalarining ishlab chiqish qoidalari bilan tanishtirish;
- turli hil hisoblash jarayonlarining algoritmlarini tuzish ko‘nikmalarini shakllantirish;
- algoritmik dasturlash tillarining asosiy nazariy asoslari tanishtirish;
- algoritmik dasturlash tillaridan foydalangan holda dasturlarni ishlab chiqishning amaliy ko‘nikmalarni rivojlantirish.

1.2. Algoritm tushunchasi va xususiyatlari

Qo‘yilgan masalani u yoki bu turini yechishning algoritmlarini shakllantirish va ishlab chiqish eng ma‘suliyatli hamda muhim bosqichlardan hisoblanadi, chunki bu bosqichda keyinchalik shaxsiy kompyuterda bajarilishi kerak bo‘lgan amallarning ketma-ketligi oldindan belgilab olinadi. Algoritmda yo‘lga qo‘yilgan xatoliklar hisoblash jarayonini noto‘g‘ri bajarilishiga olib keladi, ya‘ni noto‘g‘ri natijalarni beradi.

Algoritm – bu masalani yechish usullarini izohlashdir, yoki boshqacha qilib aytganda, kutilayotgan natijalarni shaxsiy kompyuter tomonidan olish uchun bajarilayotgan hisoblash jarayolarining ketma-ketliklaridir.

“Axborot-kommunikatsiya texnologiyalari” izohli lug‘atida quyidagi ta‘riflar keltirilgan:

1. Vazifani bajarishga qaratilgan aniq belgilangan qoidalarining tartiblangan chekli to‘plami.
2. Dastlabki ma‘lumotlarni ohirgi natijaga o‘tkazuvchi hisoblash jarayoni orqali masala yechimini aniq ko‘rsatuvchi amallar mazmuni va ketma-ketligi.

Algoritm atamasi oʻrta asrlarda yashab ijod etgan buyuk oʻzbek matematigi Al-Xorazmiy nomidan kelib chiqqan. U oʻzi kashf etgan oʻnli sanoq tizimida IX asrning 825 yilidayoq toʻrt arifmetika amallarini bajarish qoidalarini bergan. Arifmetika amallarini bajarish jarayoni esa alxorazm deb atalgan. Bu atama 1747 yildan boshlab algorismus, 1950 yilga kelib algoritm deb ham ataldi. Kompyuterlar paydo boʻlishi bilan algoritm atamasi hozirgi maʼnosi bilan axborot-kommunikatsiyalar texnologiyalari sohasida eng asosiy atamalardan biri boʻlib qoldi.

Semakin I.G. va Shestakov A.P. algoritmning quyidagi tarifini berganlar: algoritm – bu qandaydir bajaruvchini boshqarish buyruqlarining ketma-ketligi.⁴

Trofimov V.V. va Pavlovskaya T.A. esa algoritmni hisoblash jarayoniga beriladigan va birlamchi maʼlumotlarni kerakli natijaga oʻzgartirish jarayonini aniq belgilovchi oddiy elementar xarakterlar izchilligini oʻzida aks ettiruvchi aniq koʻrsatma⁵ sifatida belgilaganlar.

Algoritm quyidagi **xususiyatlarga** egadir:

1. Aniqliligi, yaʼni turli hisobga olinmagan harakatlarga yoʻl qoʻymaslik va uni bajaruvchilar uchun tushunarli boʻladi. Ushbu xususiyatiga koʻra algoritmni bajarish jarayoni oʻz-oʻzidan bajariladigan xarakterga ega.

2. Natijaviyligi, yaʼni maʼlum bir oddiy bosqichlarni bosib oʻtgandan soʻng kutilayotgan natijani olishga erishishdan iborat.

3. Ommaviyligi, yaʼni bitta masalani yechish uchun emas balki shunga oʻxshash turdosh masalalar sinfini hal etish uchun xizmat qiladi.

4. Diskretlik, yaʼni algoritmni bir nechta elementar bosqichlarga ajratish imkonini mavjudligi.

5. Tushunarlik - algoritmni bajarish uchun qulay boʻlgan matn koʻrinishida tasvirlash.

6. Tugallanganligi - ketma ket bajariluvchi harakatlar soni cheksiz emas, ularni sanash mumkin

⁴ Семакин И.Г., Шестаков А.П. Основы алгоритмизации и программирования Издательство: Академия. 2016 г. - С. 6.

⁵ Трофимов В.В., Павловская Т.А. Алгоритмизация и программирование. Учебник для академического бакалавриата. Издательство: Москва – ЮРАЙТ. 2018 г. – С. 8.

Masalaning algoritmini ishlab chiqish jarayoni quyidagi **bosqichlardan** iborat:

- ma'lumotlarga qayta ishlov berish bosqichlarni belgilab olish;
- ma'lumotlarga qayta ishlov berish bosqichlari ketma-ketligini ular o'rtasidagi mantiqiy aloqalarni o'rnatish asosida aniqlash;
- har bir bosqich mazmunini izohlash.

Algoritm tuzishda quyidagilarga amal qilinsa, qo'yilgan masalani natijasini tez va to'g'ri olish mumkin:

1. Qo'yilgan masalani to'g'ri o'qish va tushunib olish, masalani qo'ygan shaxsning asosiy maqsadini bilish;
2. Ishga daxldor qiyinchiliklarni aniq ko'rish va ortiqcha narsalarni barchasini yo'qota bilish;
3. Nazariyani qo'llash mumkin bo'lgan barcha hollarni aniqlash va uni mustaqil qo'llash yoki lozim bo'lsa, maslahat olish uchun mutaxassisga murojaat qilish;
4. Qo'yilgan masalani bir-biriga bog'liq bo'lmagan tushunarli bo'laklarga ajrata olish va ular orasida bog'liklikni tushunish;
5. Dasturlashga va dastur resurslariga ketgan xarajatlarga nisbatan taqdim etilayotgan yechimni afzalligini baholash va foydalanuvchining talabini to'la qondira bilish;
6. Qo'yilgan masalani qoniqarli yechimini olishda har bir bo'lak yechimlari to'plamini bir butun holga keltirish;
7. Masala yechimini sodda va aniq tushunarli tilda bayon eta olish;
Bu til tabiiy yoki sun'iy bo'lishining ahamiyati yo'q;
8. Masalani kompyuterdan foydalanib yechish jarayonida muvaffaqiyatsizlikka uchraganda o'zini qo'lga ola bilish va boshqa yechish yo'lini qidirish.

1.3. Masalalarni kompyuterda yechish bosqichlari. Masala qo'yilishi

Turli xildagi masalalarni shaxsiy kompyuterlarda (SHK) yechishgacha tayyorlash va uni to'g'ridan-to'g'ri SHK yechishlar o'z tarkibiga quyidagi **bosqichlarni** qamrab oladi:

- masalaning qo'yilishi;

- masalani matematik ifodalash, sonli usulni tanlash;
- hisoblash jarayonini algoritmlashtirish;
- ma'lumotlarni tayyorlash va tarkibini aniqlash;
- kompyuter dasturini tuzish;
- dasturni sozlash;
- masalani shaxsiy kompyuterda yechish va natijalarini tahlil qilish.⁶

Masalaning qo'yilishi. Turli xildagi masalalarni yechishning dastlabki bosqichi bo'lib uning qanday qo'yilishi hisoblanadi. Masalaning qo'yilishi deganda, avvalom bor uni to'g'ri va ravon ifoda qilmoq, hamda uni to'g'ri yechish uchun qo'yilayotgan hamma shartlarini aniqlash tushuniladi. Undan tashqari masalani yechish uchun kerak bo'lgan birlamchi axborotlar ham aniqlanish kerakdir va hisoblanayotgan kattaliklar ro'yhati va ularni taqdim etish shakli ko'rsatilishi lozim. Qo'yilgan maqsadni amalga oshirish uchun kerakli ma'lumotlar tarkibi (strukturasi), tuzilishi, ifodalanishi aniqlangan bo'lib, ular orasidagi bog'lanishlar aniq ifodalangan bo'lsa, "masala qo'yilgan" deb aytiladi.

1.4. Masalani matematik ifodalash. Sonli usulni tanlash

Masalani matematik ifodalash. Sonli usulni tanlash. Bu bosqichda masalani yechish uchun kerakli va yetarli bo'lgan dastlabki ma'lumotlarning tarkibi, tavsifi, turi, tuzilishi hisobga olingan xolda matematik terminlarda ifodalanadi hamda masalani yechishning matematik modeli yaratiladi.

Buning uchun har xil (sohasiga qarab) matematik apparat ishlatilishi mumkin.

Masalan iqtisod soxasidagi mutaxassislar - chiziqli dasturlash, dinamik dasturlash, stoxastik dasturlash, bashorat (prognoz) qilish bilan bog'liq masalalarni yechish matematik apparatini bilishlari kerak.

Mutaxassis o'z soxasini har tomonlama yaxshi o'rgangan va amaliy jixatdan puxta o'zlashtirgan va qo'llaniladigan har xil matematik apparatning barcha imkoniyatlarini to'liq tushunib yetgan va amaliyotga qo'llay oladigan bo'lishi kerak.

⁶ Семакин И.Г., Шестаков А.П. Основы алгоритмизации и программирования Издательство: Академия. 2016 г. - С. 6.

Bu bosqichda 2 ta asosiy savolga javob topish kerak:

1. Masalani ifodalash uchun qanday matematik strukturalar maqsadga muvofiq keladi?

2. Yechilgan o'xshash masalalar bormi?

Tanlangan matematik struktura (apparat)da masalaning elementida ob'ektlari to'la ifodalanishi zarur.

Sonli matematik usullar hisoblash jarayolarini elementar operatsiyalarga ajratishga yordam beradi, ya'ni hohlagan masalani yechishni operatsiyalarni bajarish ketma-ketligiga olib keladi. Ularning tarkibiga arifmetik operatsiyalar($<$, $\leq$, $>$, $\geq$, $=$) va harakatlar kirib, hisoblash jarayonini tarmoqlashtirish uchun xizmat qiladi. Aniq bir masalani yechish uchun sonli usullarning barchasining orasidan shundayini tanlash kerakki, u qo'yilayotgan masalani yaxshi darajada yechishni ta'minlab berishi lozim.

Ma'lum bir sonli usulni tanlashda turli shartlar qatori quyidagi omillar bo'lishi mumkinki, ya'ni hisoblashlar natijasining aniqliligi, SHK masalani yechilish vaqti, talab qilinayotgan xotira hajmi va boshqalar.

1.5. Hisoblash jarayonini algoritmlashtirish

Algoritmlashtirish jarayoni masalani yechish algoritmini tuzishni ta'minlaydi va birlamchi axborotni natijaviyga aylantirish bosqichlar ketma-ketligini aniqlash va ifoda etish uchun xizmat qiladi.

Masalani yechish algoritmi kompyuterning imkoniyatlarini, yechish aniqligini hamda masalani kompyuterda yechish vaqtini va qiymatini hisobga olgan xolda yaratilsa maqsadga muvofiq kelgan bo'lar edi.

Masalani yechish algoritmi kutilayotgan natijani olish maqsadida dastlabki ma'lumotlar va oraliq natijalar ustidan bajarilayotgan elementar operatsiyalar tartibini aniq belgilab berishi kerakdir.

Algoritm masalani yechish usuli hisoblanmaydi, balki tanlangan sonli usulni amalga oshiradi. 2-3 bosqichlar bir-biri bilan jips, mustahkam bog'langan. Ya'ni yaratilgan uslubni har xil usullar bilan amalga oshirish mumkin, shu sababdan

masalani yechish uslubi va algoritmining bir nechta variantlari bo'lishi mumkin va keraklisi tanlab olinadi.

Murakkab masalaning algoritmini yaratishda qadamma-qadam oydinlashtirish uslubidan foydalangan ma'qul, har bir qadamda algoritmning tarkibi sodda va tushunarli bo'lib qolishiga erishmoq kerak.

Masalani algoritmlash jarayonida, algoritmning ba'zi bo'laklarini, lavhalarini, mantiqan alohida qismlarini ifodalashda tipik algoritmlar va amaliyotda tekshirilgan algoritmlardan iloji boricha ko'p foydalangan ma'qul.

Algoritmlashda modullik prinsipidan foydalanish algoritmni o'qishda va dasturlashda qulayliklar yaratadi. Oxir oqibatda masalani yechish algoritmi ishchi holatga keltiriladi, ya'ni algoritm grafik ko'rinishda biror algoritmik til vositasida ifodalash darajasiga keltiriladi. Agar bitta sonli usulni amalga oshirish uchun bir nechta algoritmlar mavjud bo'lsa, u holda ular orasidan shaxsiy kompyuterdan eng samarali foydalanish uchun xizmat qiladigan usuli tanlab olinadi (tarkibida kam operatsiyalar bori kam xotira talab qiladi).

Dasturning samarador va unumdorligi, masalani yechish algoritmining qanchalik puxta tashkil qilinganligiga bog'liq.

Ma'lumotlarni tayyorlash va tarkibini aniqlash. Ma'lumotlarni tasvirlash usulini tanlash algoritmning bajarilishi bilan chambarchas bog'langan. Shu sababdan ma'lumotni tasvirlashning shunday turini, usulini tanlash kerakki, masalani echish jarayoni sodda va tushunarli bo'lsin.

Ma'lumotlar oddiy o'zgaruvchilar ko'rinishida (bu xol juda kam uchraydi), massiv ko'rinishida, alohida ma'lumot fayllari (ketma-ket o'qiladigan yoki bevosita o'qiladigan) ko'rinishida axborot tashuvchida joylashgan bo'lishi mumkin.

1.6. Dastur tuzish. Dasturni tahrirlash

Dastur tuzish. Dasturlash – bu shaxsiy kompyuter dasturlash tilida algoritmni tasvirlash jarayonidir. Agar dastur kompyuter tilida (ya'ni kompyuter komandalarida) tuzilsa, u holda dastur to'g'ridan-to'g'ri shaxsiy kompyuterda amalga oshiriladi.

Dastur simvollar tilida yozilsa, u holda dastur matni translyator orqali kompyuter tiliga o'g'iriladi va undan keyin u shaxsiy kompyuterda yechiladi.

Dasturlash jarayonida quyidagi takliflar inobatga olinsa xatolarni tuzatish jarayoni engillashadi.

1. Dastur umumiy bo'lishi kerak, ya'ni ma'lumotlarni aniq biror turiga bog'liq bo'lmasligi kerak, massivning chegara parametrlarini tekshirmoq lozim. Massiv elementlarining soni 0 yoki 1 bo'lib qolish, yoki yuqori chegarasidan oshib ketish xolati.

2. O'zgarmas kattalik xamda o'zgaruvchi kattalik ko'rinishida ishlatish. (Biror o'zgarmas kattalikni boshqasi bilan almashtirish zarurati bo'lib qolsa, dastur matnini chaqirib o'zgartirish kerak - bu noqulay xolat EXE, COM fayllarida aslo mumkin emas).

Dasturda kiritiladigan ma'lumotlarni nazorat qilish qismi bo'lishi kerak.

3. Dasturdagi arifmetik amallarni kamaytirish va dasturning ishlashini tezlatish uchun:

- darajaga oshirish amallari ko'paytirish amali bilan almashtirilgani ma'qul;
- bir xil ma'lumot bilan hisoblanayotgan arifmetik (algebraik) ifodalarni bir marta hisoblab qiymatini biror o'zgaruvchida saqlab ishlatish.
- takrorlashlarni tashkil qilishda takrorlanishning chegarasini tekshirish uchun ifodalardan emas balki oddiy o'zgaruvchilardan foydalanish.
- takroriy hisoblashlar tarkibida uchraydigan va takrorlanish davomida qiymatini o'zgartirmaydigan ifodalarni takrorlanishdan tashqarida hisoblash.

4. Dasturning xar bir bo'lagi, moduli qismiga tushuntirishlar yozilgan bo'lishi kerak. Dasturdagi tushuntirishlar, masalani echish ketma-ketligini ifodalovchi mantiqiy ketma-ketlikdan iborat bo'lmog'i kerak.

Dasturdagi modullar, qismlar aniq ko'rsatilgan bo'lishi kerak. Takrorlanish boshi va takrorlanish oxiri aloxida qatorda turgani ma'qul.

Dasturni sozlash. Dasturdagi xatoliklarni aniqlash va ularni to'g'rilash jarayoni sozlash deyiladi. Dasturlarda sintaksis va mantiqiy xatolar bo'lishi mumkin. Translyator tomonidan dasturni sintaksis nazorat qilish natijasida dasturlash tili

tomonidan ularni tuzishda ruxsat etilmagan turli konstruktsiyalar aniqlanadi. Sintaksis xatoliklarning mavjudligi dasturning to'g'ri bajarilishiga yo'l qo'ymaydi. Shunda ham dasturning bajarilishi yakunlansa, u holda olingan natijalar shubha uyg'otadi.

Sintaktik xatoliklarni tuzatgandan so'ng dasturchi kompyuterda olingan natijalar asosida dasturning to'g'ri ishlayotganini aniqlash kerak bo'ladi. Agar olingan natijalar kutilishi kerak bo'lgan natijalar bilan mos kelsa, u holda dasturni mukammalroq qilib yana bir bor tekshirib olish lozim. Agar natijalar noto'g'ri bo'lib chiqsa, u holda biz judayam yomon bo'lgan mantiqiy xatoliklarga duch kelgan bo'lamiz.

Mantiqiy xatoliklarga asosan dastur mazmunini loyihalashtirayotganda yo'l qo'yiladi. Masalani mohiyatini tahlil qilish, algoritmini tuzish yoki dasturni o'zini dasturlash tilida yozish vaqtida dasturchi tomonidan uning ma'nosi bo'yicha xatolikka yo'l qo'yilishi mumkindir. Shuning uchun ham dasturlashni dastlabki bosqichiga qaytib qo'yilgan masalaning ma'no-moxi-yatini chuqurroq o'rganishga to'g'ri keladi. Vujudga kelgan holat algoritmda e'tiborga olinganmi? Hisoblovchi (schetchik) sikldan to'g'ri foydalanilayaptimi? Hisoblovchilar uchun to'g'ri birlamchi va natijaviy qiymatlar ko'rsatilganmi? Birlamchi ma'lumotlar dasturlash tilida mumkin bo'lgan formatga egamimi yoki yo'qmi? Hisoblashlar to'g'ri asoslanganmi? Ushbu savollarni shu bosqichda hal qilishga to'g'ri keladi. Mantiqiy xatoliklarning borligi bosib o'tilgan bosqichlarning barchasi yoki ayrimlarini qaytadan tekshirib chiqishni talab etadi.

Dasturning to'g'ri ishlashi va yo'l qo'yilgan xatoliklarni aniqlash uchun testlardan foydalaniladi. Test - maxsus tayyorlangan dastlabki malumotlar bo'lib, ular ustida amallar bajarish bilan masalaning echimi-natija olinadi. Test tayyorlash juda murakkab ish bo'lib, qo'lda hisob-kitob ishlarini bajarishni talab qiladi xamda dasturning xamma qismlarini, bo'laklarini, modullarini tekshirish shart.

Dasturning xatosini tuzatish bo'yicha yo'l - yo'riqlar:

1. Maxsus tayyorlangan ma'lumotlar asosida dasturni qo'lda echib chiqish (imkoni bo'lsa) yoki mantiqan alohida bo'lgan bo'laklarini, modullarini qo'lda hisoblash.

2. Dasturni va uning bo‘laklarini, modullarini test yordamida tekshirish.

3. Dasturning kerakli joylariga bosib chiqarish buyrug‘ini qo‘yish (tuzatishlardan keyin olib tashlanadi).

4. Dasturning xatolarini tuzatishda, muloqot rejimida bajarilganda (STOP) to‘xtash buyrug‘idan foydalanish.

5. Dasturlash tilini va amal bajaruvchi tizimi (AT)ning maxsus xatolarni tuzatish imkoniyatlaridan foydalanish.

6. Xatolarni tuzatish jarayonida kam xajmdagi ma'lumotlar bilan ishlashni tashkil qilish.

Yo‘l qo‘yilgan xatolikning mazmunidan kelib chiqqan holda ularni tuzatish barcha dasturni to‘liq yoki ayrim qismini masala algoritmgiga kerak bo‘lgan o‘zgartirishlarni kiritgandan keyin qaytadan tuzishni talab etadi. Bu jarayon katta kuch va kompyuter vaqtini ishlatishni talab etadi, shuning uchun ham dasturni yozishning dastlabki bosqichlaridan e‘tibor yuqori darajada bo‘lsa uni oldini olish mumkin bo‘ladi.

1.7. Masalani shaxsiy kompyuterda yechish va natijalarni taxlil qilish

Kompyuter xatolari tuzatilib tayyorlangan dastlabki ma'lumotlardan foydalangan xolda masalaning yechimini (yechimlarini) avtomatik tarzda hisoblaydi. Agar natijalar masalaning yechimi uchun yaroqli deb topilsa masalani yechish tugallangan hisoblanadi, aks xolda yuqoridagi bosqichlar qaytadan ko‘rib chiqiladi. Masalani yechish natijasida olingan sonlar yoki sonlar massivi, matnlar yoki matn ko‘rinishidagi massivlar xar taraflama izoxlanib, tushuntiriladi. Dasturdan foydalanish uchun ko‘rgazma yozish quyidagilarni o‘z ichiga oladi:

- Dastur ishlashi uchun ma'lumotlarni tayyorlash usuli, tuzilishi aniq belgilangan;
- Dasturning ishlashi uchun kompyuterni sozlash yo‘llari;
- Dasturni ishga tushirish va ishlash paytida bo‘ladigan savol-javoblar;
- Dasturni ishlash jarayonida kelib chiqadigan xar xil xolatlarni bartaraf qilish yo‘llari aniq va puxta tushunarli qilib yozilgan bo‘lishi kerak.

Tayanch iboralar

algoritm, algoritmning xususiyatlari, aniqliligi, natijaviyligi, ommaviyligi, diskretlilik, masalaning qo'yilishi, masalani matematik ifodalash, sonli usulni tanlash, algoritmlashtirish, sozlash, dasturlash, test.

Nazorat va muhokama uchun savollar

1. Algoritm deganda siz nimani tushunasiz? Algoritmga misolar keltiring.
2. Algoritm qanday xususiyatlarga ega?
3. Masalaning algoritmini ishlab chiqish jarayoni qanday bosqichlardan iborat?
4. Masalalarni kompyuterda yechish jarayoni qanday bosqichlardan iborat?
5. Masalaning qo'yilishi deganda nimani tushunasiz?
6. Sonli usulni tanlash qanday amalga oshiriladi?
7. Hisoblash jarayonlarini algoritmlashtirishni tushuntirib bering.
8. Ma'lumotlarni tayyorlash va tarkibini aniqlash (tanlash) deganda siz nimani tushunasiz?
9. Dasturlash deganda nimani tushunasiz?
10. Dasturni sozlash bosqichida qanday ishlar amalga oshiriladi?

2- bob. ALGORITMLARNI AKS ETTIRISH USULLARI

2.1. Algoritmni aks ettirish usullari. Algoritmni oddiy tilda ifodalash.

2.2. Algoritmni blok-chizmalar ko‘rinishida ifodalash.

2.3. Algoritmni maxsus (algoritmik) tilda yozish.

2.1. Algoritmni aks ettirish usullari. Algoritmni oddiy tilda ifodalash

Algoritmni ishlab chiqishda uni bir necha xil usul bilan ifodalab bersa bo‘ladi.

Shulardan uchta keng tarqalgan. Bular:

1. Algoritmni oddiy tilda ifodalash;
2. Algoritmni blok-chizmalar ko‘rinishida ifodalash;
3. Algoritmni maxsus (algoritmik) tilda yozish.

Algoritmni ifodalashning eng keng tarqalgan shakli - oddiy tilda so‘zlar bilan bayon qilishdir. Bu nafaqat hisoblash algoritmlarida, balki hayotiy, turmushdagi "algoritm"larga ham tegishlidir.

Masalan, biror bir taom yoki qandolat mahsulotini tayyorlashning retsepti ham oddiy tilda tavsiflangan algoritmdir. Shaharlararo telefon - avtomat orqali aloqa o‘rnatishning o‘ziga xos algoritmidan foydalanasiz. Do‘kondan yangi kir yuvish mashinasi yoki magnitofon sotib olinsa, ishni foydalanishning algoritmi bilan tanishishdan boshlaymiz.

Masalani kompyuterda yechishda ham, ko‘pincha matematika tilini ham o‘z ichiga olgan tabiiy tildan foydalanish mumkin. Algoritmning bunday tildagi yozuvi izlanayotgan natijaga olib keladigan amallar ketma-ketligi ko‘rinishida bo‘lib, odam tomonidan bir ma'noli idrok etilishi kerak. So‘zlar bilan ifodalangan har bir amal “algoritmning qadami” deb ataladi. Qadamlar tartib nomeriga ega bo‘ladi.

Algoritm ketma-ket, qadam-ba qadam bajarilishi kerak. Agar algoritm matnida "N sonli qadamga o‘tilsin" deb yozilgan bo‘lsa, bu algoritmning bajarilishi ko‘rsatilgan N-qadamdan davom etishini bildiradi.

Algoritmni oddiy tilda ifodalash kompyuterga kiritish uchun yaramaydi. Buning uchun algoritmni kompyuter tilida shunday bayon qilish kerakki, masalan

kompyuterda yechish jarayonida bu algoritmi ishini avtomatik boshqarib turadigan bo'lsin.

Kompyuter tushunadigan shaklda yozilgan algoritmi masalani yechish dasturidir.

Algoritmi oddiy tilda yozishda to'rt xil amaldan: hisoblash, N- qadamga o'tish, shartni tekshirish, hisoblashning oxiri, shuningdek kiritish va chiqarish amallaridan foydalanilgan maqul. Bular ichida eng ko'p foydalaniladigani hisoblash amalidir.

2.2. Algoritmi blok-chizmalar ko'rinishida ifodalash

Nisbatan murakkab masalalarni yechishda algoritmdan muayyan kompyuter tilidagi dasturga o'tish juda qiyin. Bunday bevosita o'tishda algoritmi alohida qismlari orasidagi bog'lanish yo'qoladi, algoritmi tarkibining asosiy va muhim bo'lmagan qismlarini farqlash qiyin bo'lib qoladi.

Bunday sharoitda keyinchalik aniqlash va to'g'rilash ancha vaqt talab qiladigan xatolarga osongina yo'l qo'yish mumkin.

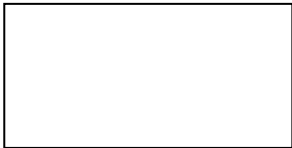
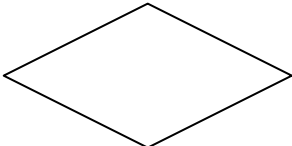
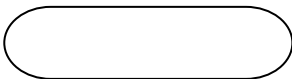
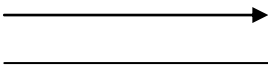
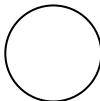
Odatda algoritmi bir necha marta ishlab chiqiladi, ba'zan xatolarni to'g'rilash, algoritmi tarkibini aniqlashtirish va tekshirish uchun bir necha marta orqaga qaytishga to'g'ri keladi. Algoritmi ishlab chiqishning birinchi bosqichida algoritmi yozishning eng qulay usuli - algoritmi grafik ko'rinishda ifodalashdir.

Algoritmi bosqichlar ketma-ketligini grafik jihatdan tasvirlab berish blok-chizmalar deb ataladi.

Blok-chizmalarning har bir qismi ma'lum bir shakl ko'rinishdagi geometrik figuralar bilan tasvirlanadi. Blok-chizma ANSI simvollar deb nomlanuvchi (Amerika milliy standartlar instituti) standart figuralaridan foydalanadi. Har bir figura bitta bosqichni ko'rsatadi va blok deb nomlanadi.

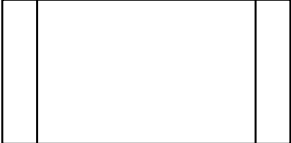
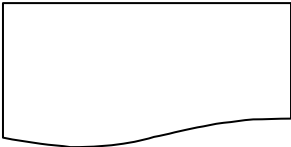
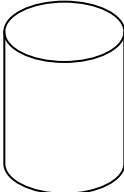
Algoritmlashtirishda ko'p qo'llaniladigan **bloklar** 2.1-jadvalda keltirilgan.

Algoritmlashtirishda ko`p qo`llaniladigan bloklar va ularning ta`rifi⁷

	- jarayon (bitta yoki bir guruh operatsiyalarni bajarish)
	- qaror qabul qilish (ma'lum bir shartlar asosida algoritm yoki dasturni bajarilish yo`nalishini tanlash)
	- modifikatsiya (dastur, buyruqlarni o`zgartiruvchi operatsiyalarni bajarish)
	- boshlash, tugatish. Ma'lumotlarni qayta ishlash jarayonini boshlash, tugatish va to`xtatib turish uchun xizmat qiladi.
	- kiritish, chiqarish (ma'lumotlarni qayta ishlash uchun kerakli bo`lgan shaklga keltirish(kiritish) yoki qayta ishlash natijalarini ro`yhatga olish (chiqarish)).
	- oqimlar chizig'i. Simvollar orasidagi aloqalarning ketma-ketligini tasvirlab beradi.
	- ulovchi. O`zaro bog`laydigan simvollar asosida oqimlar chizig'i o`rtasida uzilgan aloqalarni ko`rsatadi.

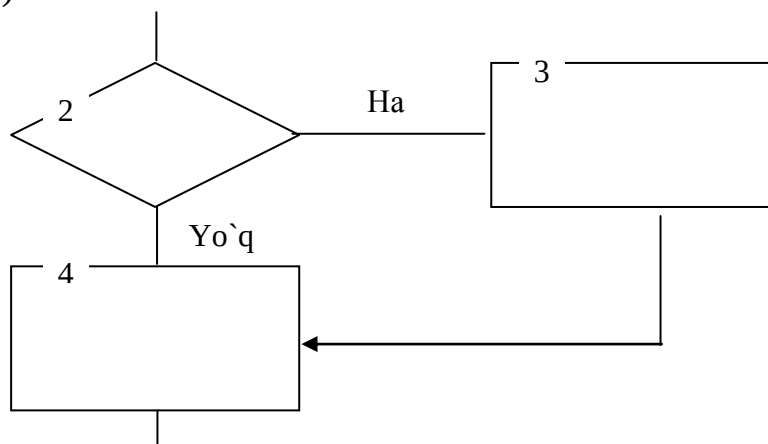
⁷ R.Dadabayeva, Sh.Nasridinova, N.Shoaxmedova, L.Ibragimova, Sh.Ermatov. Axborot-kommunikatsion texnologiyalar va tizimlar.-T.: "Sano-standart" nashriyoti. O`quv qo`llanma. 2017. B. 255

2.1-jadval (davomi)

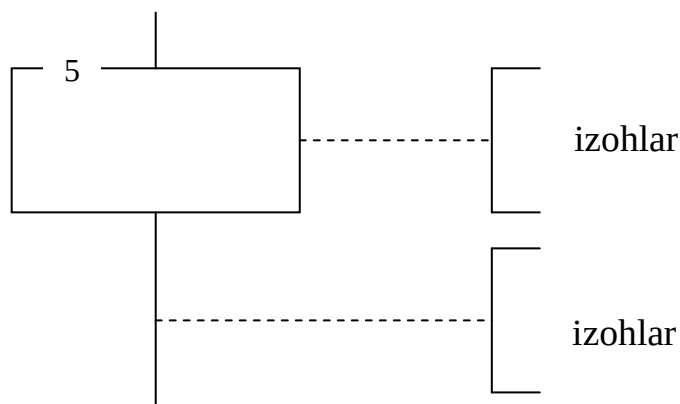
	- oldindan aniqlangan jarayon.
	- xujjat.
	- magnitli disk.
	- betlarni o`zaro bog`lovchi.

Oqimlar chizig'ining normal yo`nalishi bo`lib yuqoridan quyiga va chapdan o`ngga hisoblanadi hamda strelkalar(yo`naltiradigan) bilan belgilanmaydi. Boshqa barcha holatlarda strelkalar bo`lishi kerakdir.

Bloklar raqami chiziqlar tasvirini uzgan holda blokning chap yuqori qismiga qo`yiladi. Bloklar harflar bilan yoki harf-raqam ko`rinishida belgilanishi ham mumkin (2.1-rasm).

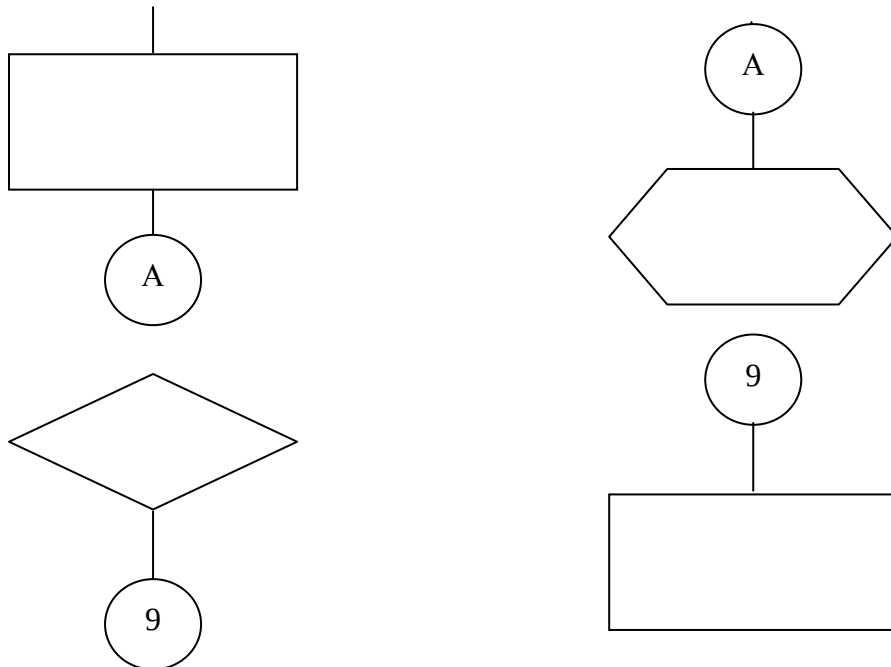
2.1-rasm. Blok-chizmani ko`rinishidan lavha⁸

Agar blokning mazmunini ifoda etuvchi soʻzlar blokka sigʻmasa, u holda «izohlar»dan foydalaniladi. Izohlar chizmalarning boʻsh joyida keltiriladi va izoh berilayotgan blok bilan chiziq orqali yoki shtrixli chiziq bilan ulanadi (2.2-rasm).



2.2-rasm. Blok-chizmaga izohlar qoʻyish⁹

Uzoq joylashgan bloklarning bir-biri bilan boʻlgan oqimlar chizigʻi uzilishi mumkin. Bunday hollarda uzilish tugagan va boshlangan joyda harf, raqam yoki harf-raqam koʻrinishdagi «Ulovchi» belgisi qoʻyiladi (3.3-rasm).

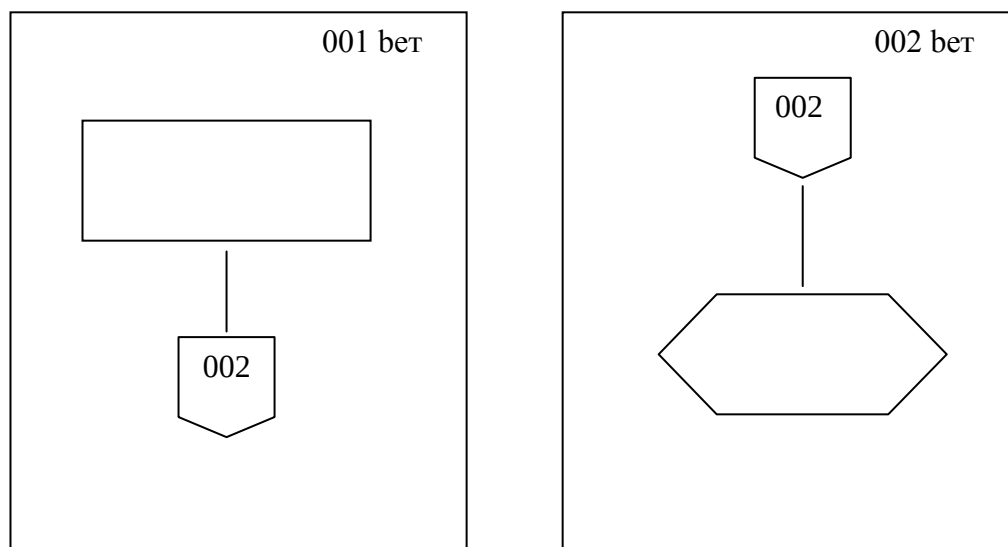


2.3-rasm. Blok-chizmaga «Ulovchi» belgisini qoʻyilishi¹⁰

⁹ R.Dadabayeva, Sh.Nasridinova, N.Shoaxmedova, L.Ibragimova, Sh.Ermatov. Axborot-kommunikatsion texnologiyalar va tizimlar.-T.: “Sano-standart” nashriyoti. Oʻquv qoʻllanma. 2017. B. 256

¹⁰ R.Dadabayeva, Sh.Nasridinova, N.Shoaxmedova, L.Ibragimova, Sh.Ermatov. Axborot-kommunikatsion texnologiyalar va tizimlar.-T.: “Sano-standart” nashriyoti. Oʻquv qoʻllanma. 2017. B. 257

«Ulovchi»ning nomi bo`lib qabul qilib oluvchi blokning nomeri (identifikator) xizmat qiladi. Agar oqimlar chizig`ining o`zaro bog`lanishi ulovchilari turli betlarda bo`lsa, u holda betlararo bog`lovchi qo`llanilib, uning ichiga bet va blokning nomerlari keltiriladi (3.4-rasm).



2.4-rasm. Blok-chizmaga betlararo bog`lovchi belgisini qo`yilishi¹¹

Blok-chizmalarning asosiy afzalligini, algoritmni tasvirlashning yaqqolligi bilan izohlash hisoblanadi.

Masalani yechishning algoritmlarini blok-chizmalar asosida ifoda etib ularning yaqqolligini ta`minlash borasida quyidagi **texnik qoidalarni** inobatga olish kerak bo`ladi:

- a) bloklarni gorizontaal va vertikal holatda joylashtirish kerak;
- b) bloklar orasidagi masofani shunday olish kerakki, ularni ulab turuvchi chizig`lar etarli darajada kalta bo`lsin;
- v) bloklarning geometrik o`lchami hisoblash jarayonini tavsiflashning hajmiga to`g`ri kelishi kerak. Shu bilan birgalikda blokning uzunligi uning balandligiga nisbatan  $v=1,5 \cdot a$  ko`rinishda bo`lishi kerak;

¹¹ R.Dadabayeva, Sh.Nasridinova, N.Shoaxmedova, L.Ibragimova, Sh.Ermatov. Axborot-kommunikatsion texnologiyalar va tizimlar.-T.: “Sano-standart” nashriyoti. O`quv qo`llanma. 2017. B. 257

g) blok-chizmalarda bloklarning joylashishi va guruhlarga ajratilishi shunday bo'lishi kerakki, ularni ko'rib axborotlarni yaxshi qabul qilish ta'minlanishi lozim.

Algoritmni qurish qoidalari GOST 19. 002-80 da (xalqaro standart ISO 2636-73 ga mos keladi) qat'iy belgilab berilgan. GOST 19. 003 -80 (ISO 1028-73ga mos) algoritm va dasturlar tuzimlarida qo'llaniladigan simvollar ro'yxatini, bu simvollarning shakli va o'lchamlarini, shuningdek ular bilan tasvirlanadigan funksiyalarni (amallarni) belgilaydi.

2.3. Algoritmni maxsus tilda ifodalash

Bu usulda algoritmni ifodalash uchun “dasturlash tillari” deb ataluvchi suniy tillar qo'llaniladi. Buning uchun ishlab chiqilgan algoritm shu tillar yordamida bir ma'noli va kompyuter tushuna oladigan ko'rinishda tavsiflanishi zarur.

Uning tarkibida cheklangan sondagi sintaksis konstruksiyalar to'plami bor bo'lib, u bilan algoritm yaratuvchi tanish bo'lishi kerak. Ana shu konstruksiyalardan foydalanib buyruq va ko'rsatmalar formal ifodalarga o'tkaziladi.

Dasturlash tillarida hisoblash jarayonlari ayrim avtonom bosqichlari mazmunini izohlash operatorlar yordamida amalga oshiriladi.

Operator – kompyuter dasturi bajarilayotganda ma'lum bir harakatlarni amalga oshirish bo'yicha ko'rsatma beradi.

Zamonaviy dasturlash tillari kompyuterning ichki kompyuter tilidan keskin farq qiladi va kompyuter bevosita ana shu tilda ishlay olmaydi. Buning uchun dasturlash tilidan mashina tushunadigan tilga tarjima qiluvchi maxsus dastur - translyatordan foydalaniladi.

Dasturni translyatsiya qilish va bajarish jarayonlari vaqtlarga ajraladi.

Avval barcha dastur translyatsiya qilinib, so'ngra bajarish uslubida ishlaydigan translyatorlar “kompilyatorlar” deb ataladi. Dastlabki tilning har bir operatorini o'zgartirish va bajarishni ketma-ket amalga oshiriladigan translyatorlar “interpretatorlar” deb ataladi.

Dasturlashning ixtiyoriy tili belgilar majmuini va algoritmlarni yozish uchun ushbu belgilarni qo'llash qoidalarini o'z ichiga oladi.

Dasturlash tillari bir biridan alifbosi, sintaksisi va semantikasi bilan ajralib turadi.

Alifbo - tilda qo'llaniladigan ko'plab turli ramziy belgilar (harflar, raqamlar, maxsus belgilar)dir.

Tilning sintaksisi jumlar tuzishda belgilarning bog'lanish qoidalarini belgilaydi, semantikasi esa ushbu jumlalarining mazmuniy izohini belgilaydi.

Tayanch iboralar

oddiy til, blok-chizma, dasturlash til, jarayon, qaror qabul qilish, modifikatsiya, kiritish, chiqarish, oqimlar chizig'i, ulovchi, operator, translyator

Nazorat va muhokama uchun savollar

1. Algoritmni aks ettirish usullarini sanab bering.
2. Algoritmni oddiy tilda ifodalashga misollar keltiring.
3. Algoritmning blok-chizmasi nima uchun xizmat qiladi?
4. Algoritm blok-chizmasining asosiy bloklarini sanab bering va ularga tarif bering.
5. Algoritmni maxsus tilda ifodalash qanday amalga oshiriladi?
6. Operatorlar nima uchun xizmat qiladi?

3- bob. HISOBLASH JARAYONLARINING TURLARI. CHIZIQLI HISOBLASH JARAYONLARI

3.1. Hisoblash jarayonlarining turlari.

3.2. Chiziqli hisoblash jarayoni.

3.3. Chiziqli hisoblash jarayonlariga misollar.

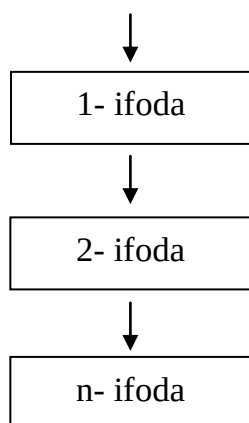
3.1. Hisoblash jarayonlarining turlari

Masala yechimining algoritmi ishlab chiqilayotgan davrda asosan uch xil turdagi algoritmlardan foydalanib, murakkab ko‘rinishdagi algoritmlar yaratiladi.

Algoritmning asosiy turlariga chizig‘li, tarmoqlangan va takrorlanadigan ko‘rinishlari kiradi.

Murakkab masalalarning yechimini olish algoritmlari yuqoridagi turlarining barchasini o‘z ichiga olishi mumkin.

Chiziqli turdagi algoritmlarda bloklar biri ketidan boshqasi joylashgan bo‘lib, berilgan tartibda bajariladi (3.1-rasm). Bunday bajarilish tartibi “tabiiy tartib” deb ham yuritiladi.

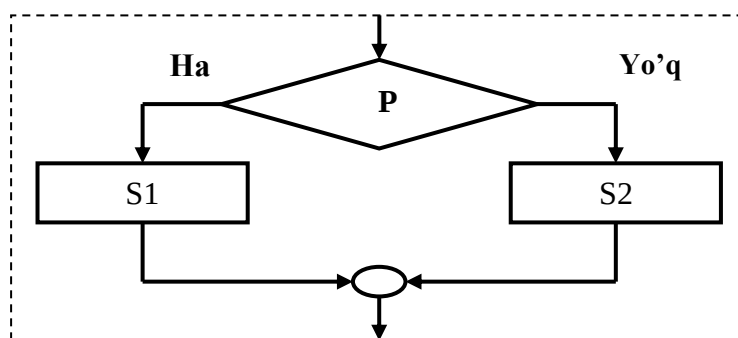


3.1-rasm. Chiziqli hisoblash jarayoni¹²

¹² Alimov R.X., Xayitmatov O‘.T., Fayzullayev S.X., Majidov R.R., Sattarova M. , Akramov A.A. Algoritm asoslari va algoritmik tillar. O‘quv qo‘llanma. TDIU, 2007. B. 30

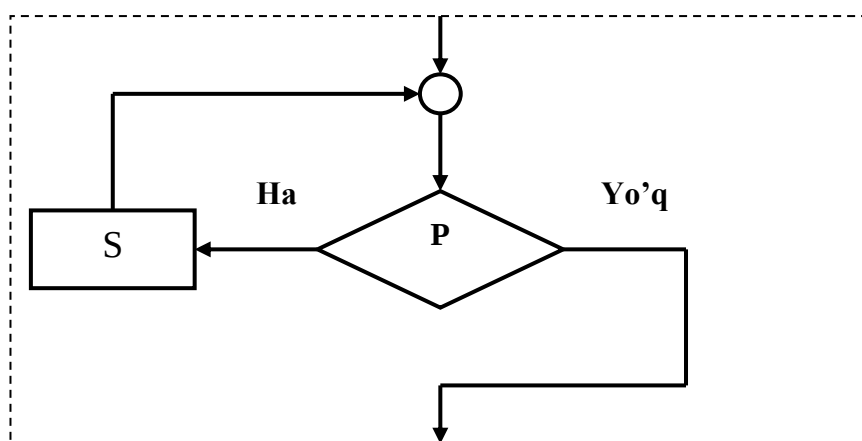
Ko‘p hollarda biron bir oraliq natijaga bog‘liq ravishda hisoblashlar yoki u yoki boshqa ifodaga ko‘ra amalga oshirilishi mumkin yani birorta mantiqiy shartni bajarilishiga bog‘lik holda hisoblash jarayoni u yoki bu tarmoq bo‘yicha amalga oshirilishi mumkin. Bunday tuzilishdagi hisoblash jarayonining algoritmi “tarmoqlanuvchi turdagi algoritm” deb ataladi (3.2-rasm).

Ko‘pgina hollarda masalalarning yechimini olishda bitta matematik bog‘lanishga ko‘ra unga kiruvchi kattaliklarni turli qiymatlariga mos keladigan qiymatlarini ko‘p martalab hisoblash to‘g‘ri keladi.



3.2-rasm. Tarmoqlangan hisoblash jarayoni¹³

Hisoblash jarayonining bunday ko‘p martalab takrorlanadigan qismi “takrorlanishlar” deb ataladi. Takrorlanishlarni o‘z ichiga olgan algoritmlar “takrorlanuvchi turdagi algoritmlar” deb ataladi (3.3-rasm).



3.3-rasm. Takrorlanuvchi hisoblash jarayoni¹⁴

¹³ Alimov R.X., Xayitmatov O‘.T., Fayzullayv S.X., Majidov R.R., Sattarova M. , Akramov A.A. Algoritm asoslari va algoritmik tillar. O‘quv qo‘llanma. TDIU, 2007. B. 31

¹⁴ Alimov R.X., Xayitmatov O‘.T., Fayzullayev S.X., Majidov R.R., Sattarova M. , Akramov A.A. Algoritm asoslari va algoritmik tillar. O‘quv qo‘llanma. TDIU, 2007. B. 31

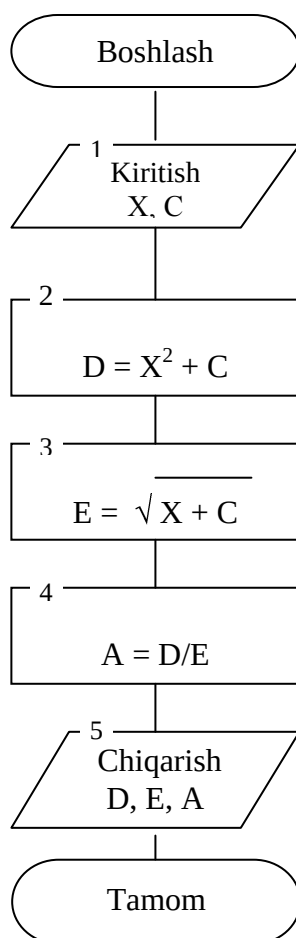
Takrorlanuvchi turdagi algoritmni yozish va chizish oʻlchamlarini sezilarli darajada qisqartirish takrorlanadigan qismlarni ixcham ifodalash imkonini beradi.

3.2. Chiziqli hisoblash jarayoni

Hisoblash jarayonlarining shunday turiga chiziqli deb aytiladiki, unda hisoblashning barcha bosqichlari chiziqli ketma-ketlik koʻrinishida bajariladi. Bunda hisoblashlarning yoʻnalishi birlamchi maʼlumot yoki oraliq natijalarga bogʻliq boʻlmaydi.

Misol. Quyidagi ifodani hisoblashning blok-chizmasini tuzing (3.4-rasm).

$$A = \frac{X^2 + C}{\sqrt{X + C}}$$



3.4-rasm. Yuqorida keltirilgan misolni yechish algoritmi

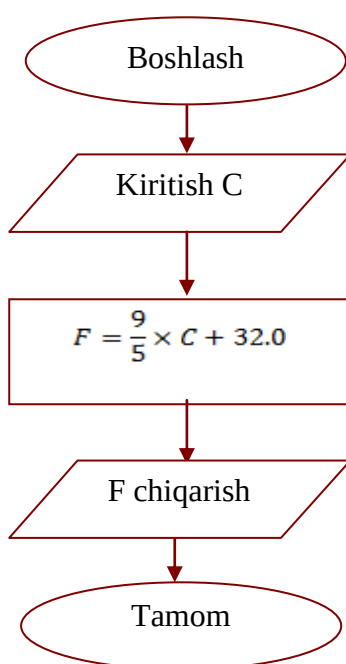
Birinchi blokda oʻzgaruvchilar X, S kiritish koʻrsatilgan, chunki ular birlamchi

ma'lumotlar bo'lib xizmat qiladi. 2,3,4 bloklarda hisoblashlar bajarilgandan so'ng D,E,A qiymatlari olinib, 5 blokning bajarilish jarayonida chiqariladi.

3.3. Chiziqli hisoblash jarayonlariga misollar

Masala 1. Selsiy bo'yicha temperaturani Farengeyt bo'yicha gradusga $F = \frac{9}{5} * C + 32,0$ formulasi yordamida o'tkazish algoritmini tuzing.

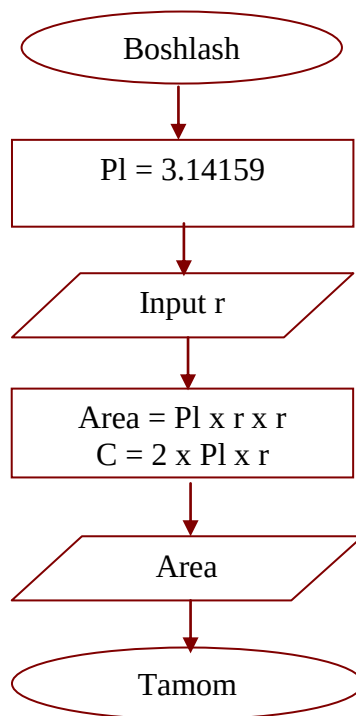
Masala yechimining algoritmini blok-chizmasi 3.5-rasmda keltirilgan.



3.5- rasm. Selsiy bo'yicha temperaturani Farengeyt gradusga o'tkazish algoritmining blok-chizmasi

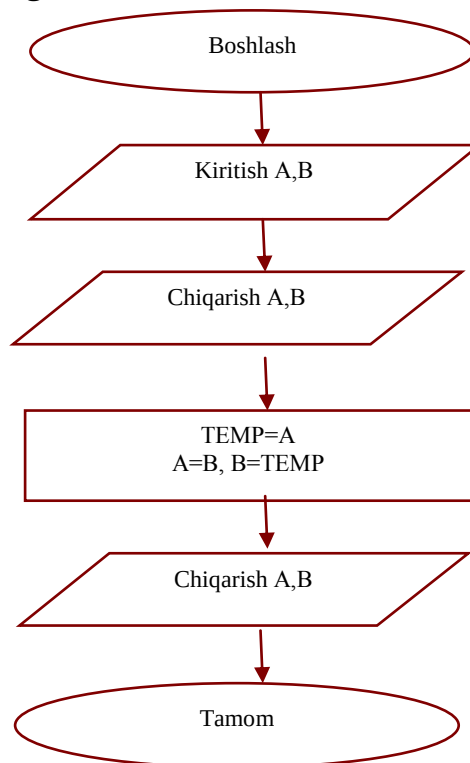
Masala 2. R radiusga ega aylananing uzinasini va yuzini topish blok chizmasini tuzing.

Masala yechimining algoritmini blok-chizmasi 3.6-rasmda keltirilgan.



3.6-rasm. R radiusga ega aylananing uzinasini va yuzini topish blok chizmasi

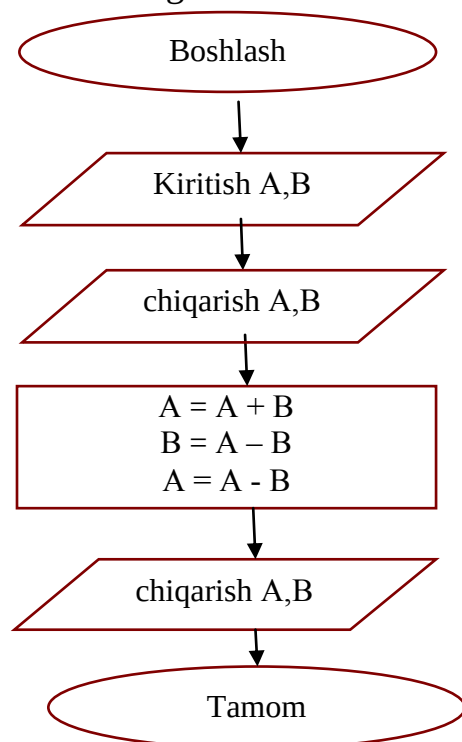
Masala 3. Ikkita sonning vaqtinchalik o'zgaruvchidan foydalanib joyini almashtirish algoritmini tuzing.



3.7- rasm. Ikkita sonning vaqtinchalik o'zgaruvchidan foydalanib joyini almashtirish blok chizmasi

Masala yechimining algoritmini blok-chizmasi 3.7-rasmda keltirilgan.

Masala 4. Ikkita sonning vaqtinchalik o'zgaruvchidan foydalanmay joyini almashtirish algoritmini tuzing.



3.8-rasm. Ikkita sonning vaqtinchalik o'zgaruvchidan foydalanmay joyini almashtirish blok-chizmasi

Masala yechimining algoritmini blok-chizmasi 3.8-rasmda keltirilgan.

Tayanch iboralar

chiziqli hisoblash jarayoni, tarmoqlanuvchi turdagi algoritm, takrorlanuvchi turdagi algoritmlar, birlamchi ma'lumot, oraliq natija, ifoda, o'zgaruvchi, takrorlanishlar.

Nazorat va muhokama uchun savollar

1. Hisoblash jarayonlarining qanday turlarini bilasiz? Ularga ta'rif bering.
2. Chiziqli hisoblash jarayoni boshqa hisoblash jarayonlaridan nima bilan farq qiladi?
3. Chiziqli hisoblash jarayonida qanday bloklardan foydalaniladi?
4. Qaysi bloklardan chiziqli hisoblash jarayoni iborat bo'lishi mumkin emas?
5. Xarakatlarning qaysi tartibi tabiiy hisoblanadi?

4- bob. TARMOQLANGAN HISOBLASH JARAYONLARI

4.1. Tarmoqlangan hisoblash jarayoni.

4.2. Oddiy tarmoqlangan hisoblash jarayonining algoritmi.

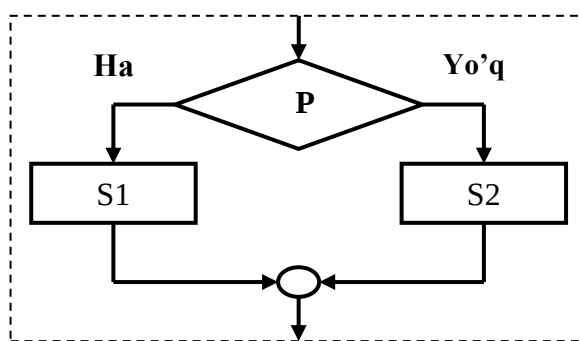
4.3. Murakkab tarmoqlangan hisoblash jarayonining algoritmi.

4.1. Tarmoqlangan hisoblash jarayoni

Hisoblash jarayonlarining shundayi tarmoqlangan deb ataladiki, unda u birlamchi yoki oraliq ma'lumotlar xususiyatidan kelib chiqqan holda bir yoki bir necha yo'nalish bo'yicha bajarilishi mumkin bo'ladi. Bunda har bir yo'nalish hisoblash jarayonining tarmog'i hisoblanadi. U yoki bu tarmoqning tanlanishi mantiqiy shartlarning bajarilishini tekshirish asosida ta'minlanadi. Ushbu algoritmlarda xarakatlarnin tabiiy amalga oshirish tartibi busiladi. So'zlar bilan tarmoqlanish quyidagicha bayon etiladi:

**AGAR shart bajarilsa (P), 1-xarakat (S1) amalga oshiriladi, AKS HOLDA
2- xarakat (S2) amalga oshiriladi**

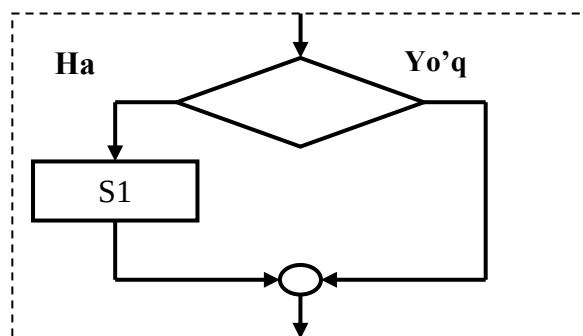
Agar ikkala xarakat ishtirok etsa unda to'liq alternativa to'risida gap ketadi (4.1-rasm).



4.1. To'liq alternativa¹⁵

¹⁵ Основы алгоритмизации : учеб.-метод. пособие для студ. оч. и заоч. обуч. технич. специальностей / В.И. Логинов, Л.Н. Шемагина. – Н. Новгород : Изд-во ФГБОУ ВПО «ВГАВТ», 2010. С. 25

Agar 2-xarakatning o‘rniga “n-bandga o‘ting” degan ko‘rsatma berilgan bo‘lsa, unda bu yozuv shakli to‘liq emas alternativa deb ataladi (4.2-rasm).

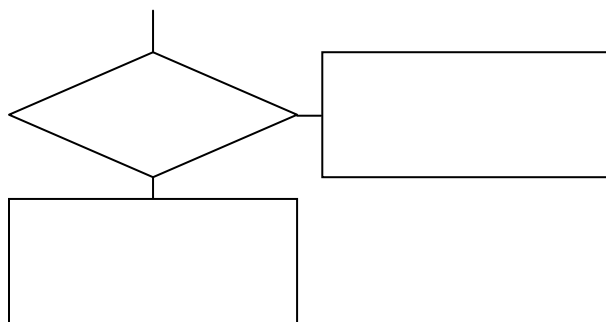


4.2. To‘liq emas alternativa¹⁶

P (shart) – “Ha” yoki “Yo‘q” qiymatini qabul qilishi mumkin bo‘lgan mantiqiy ifoda. Agar shart bajarilsa, xarakat (S1) amalga oshiriladi, aks holda xarakat bajarilmaydi.

4.2. Oddiy tarmoqlangan hisoblash jarayonining algoritmi

Tarmoqlangan hisoblash jarayonining algoritmlari oddiy, blokning ikkala tarmog‘i ham mantiqiy ifodalarni hisoblashni talab etmaydi, bo‘lishi mumkin (4.3-rasm).



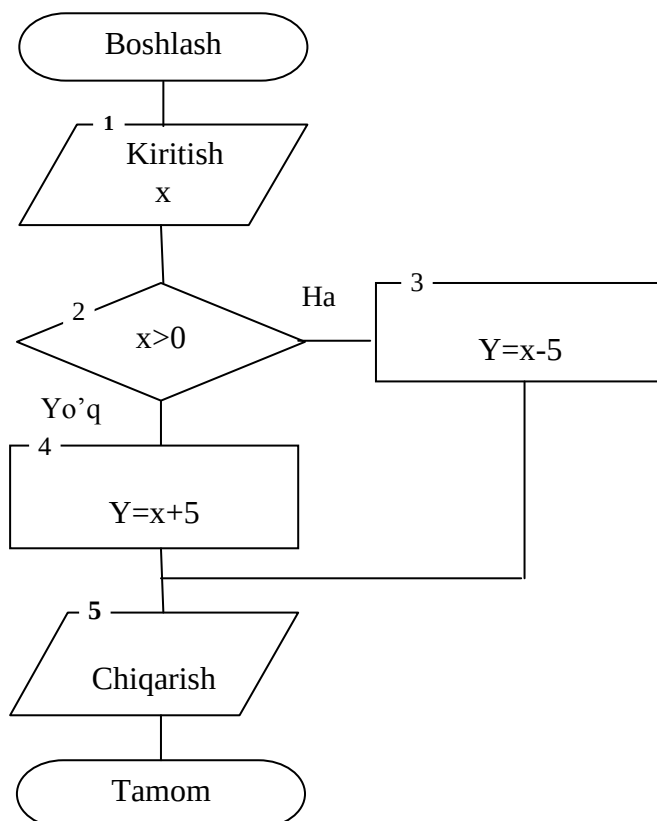
4.3-rasm. Oddiy tarmoqlangan hisoblash jarayonining algoritmi¹⁷

¹⁶ Основы алгоритмизации : учеб.-метод. пособие для студ. оч. и заоч. обуч. технич. специальностей / В.И. Логинов, Л.Н. Шемагина. – Н. Новгород : Изд-во ФГОУ ВПО «ВГАВТ», 2010. С. 26

¹⁷ R.Dadabayeva, Sh.Nasridinova, N.Shoaxmedova, L.Ibragimova, Sh.Ermatov. Axborot-kommunikatsion texnologiyalar va tizimlar.-T.: “Sano-standart” nashriyoti. O‘quv qo‘llanma. 2017. B.258

Masala 1. Quyidagi ifodani hisoblashning blok-chizmasini tuzing (4.4-rasm).

$$y = \begin{cases} x > 0 \text{ bo'lganda, } x+5, \\ x \leq 0 \text{ bo'lganda, } x-5. \end{cases}$$



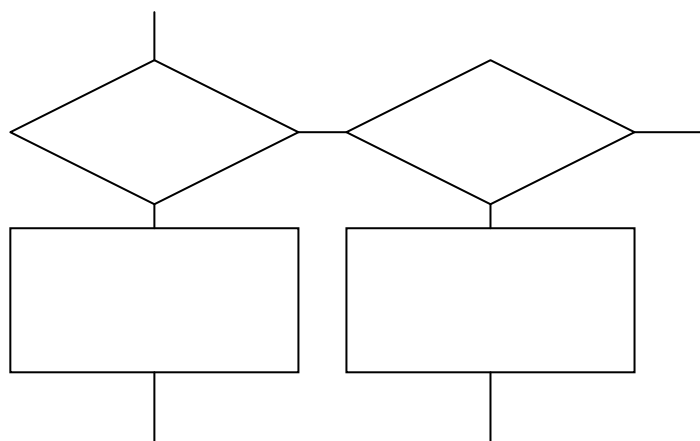
4.4-rasm.1-masalani yechish algoritmining blok-chizmasi

4.4-rasmda keltirilgan blok-chizmada hisoblash jarayonining tarmoqlangan algoritmi keltirilgan. Agar $x > 0$ sharti bajarilsa, u holda 2,4,5 bloklarning bajarilish jarayoni bo‘ladi, aks holda esa 2,3,5 bloklar bajariladi. Shunday qilib, masalaning algoritmidan kelib chiqib ikkita blokdan bittasi yoki 3-blok, yoki 4-blok bajarilishi mumkin.

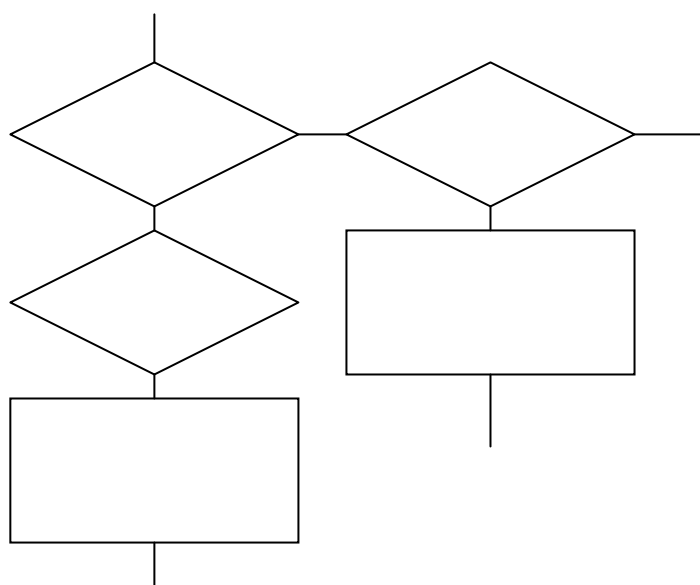
4.3. Murakkab tarmoqlangan hisoblash jarayonining algoritmi

Murakkab tarmoqlangan hisoblash jarayonida bitta yoki ikkala tarmoqlarda ham mantiqiy ifodalarni hisoblash bloklari bo‘ladi (4.5-rasm).

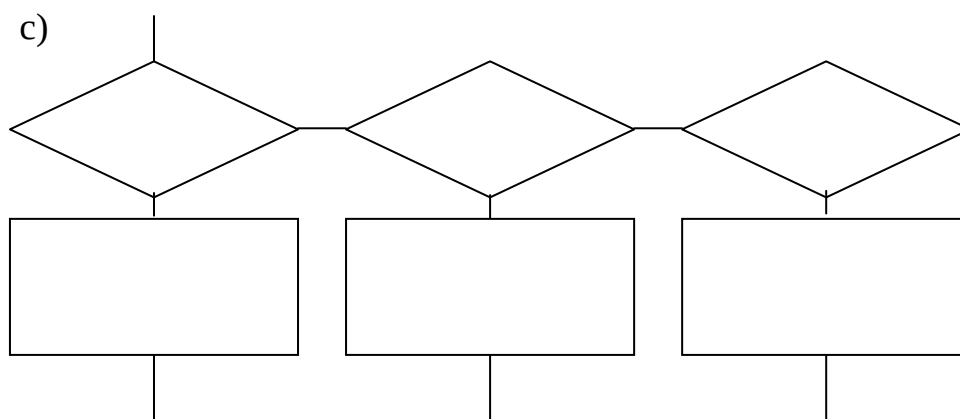
a)



b)



c)

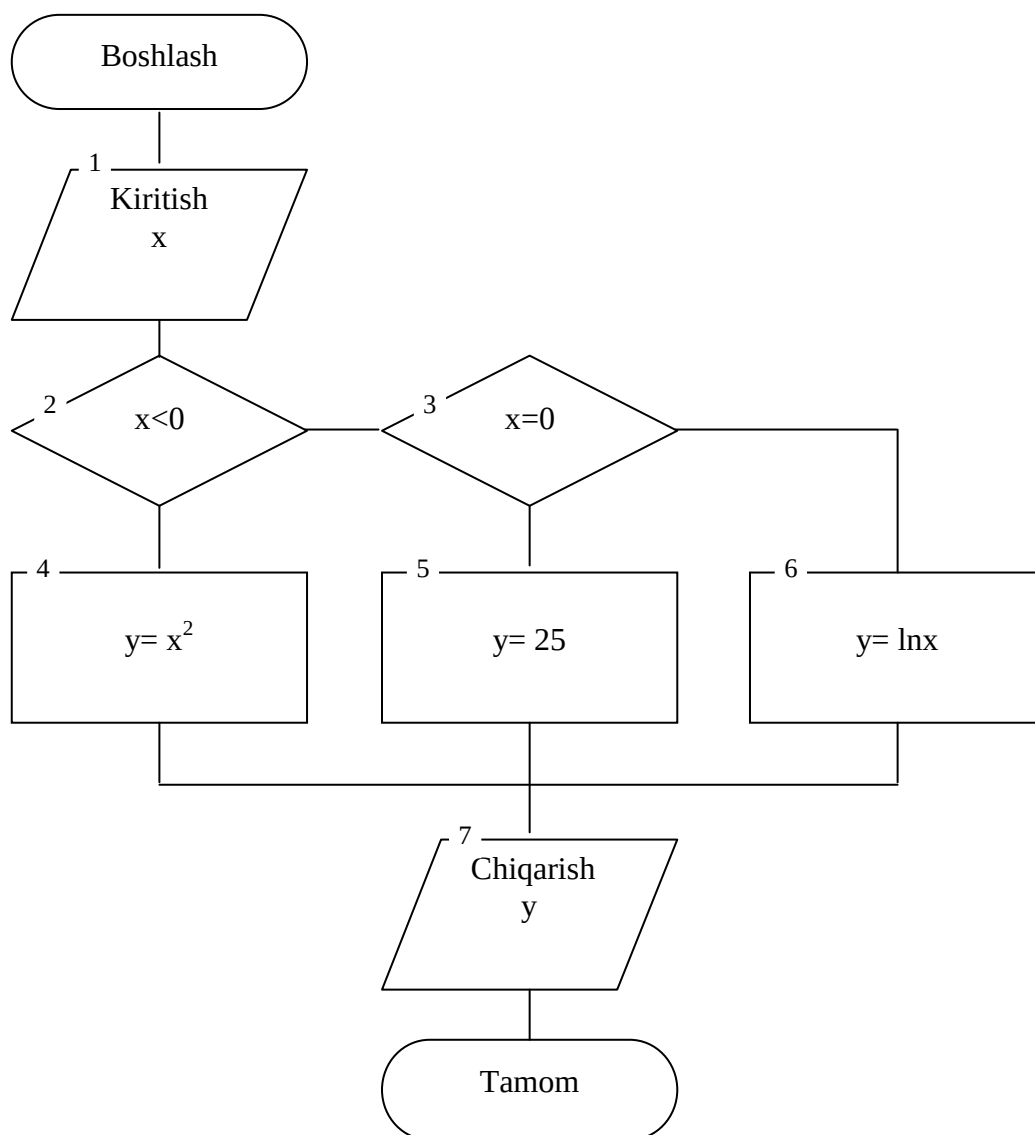


4.5-rasm. Murakkab tarmoqlangan hisoblash jarayonining algoritmlari¹⁸

¹⁸ R.Dadabayeva, Sh.Nasridinova, N.Shoaxmedova, L.Ibragimova, Sh.Ermatov. Axborot-kommunikatsion texnologiyalar va tizimlar.-T.: “Sano-standart” nashriyoti. O`quv qo`llanma. 2017.B. 260

Masala 2. Quyidagi ifodani hisoblashning blok-chizmasini tuzing (4.6-rasm).

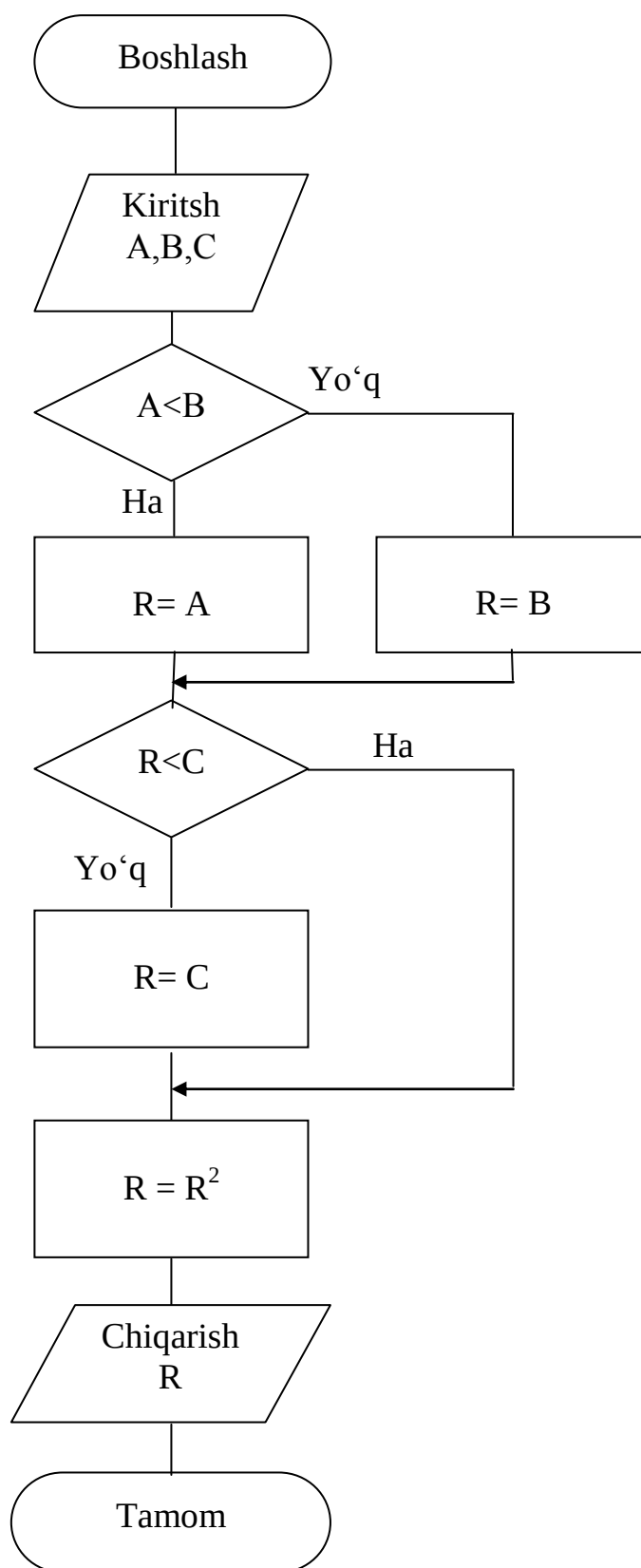
$$y = \begin{cases} x < 0 \text{ bo'lganda, } x^2, \\ x = 0 \text{ bo'lganda, } 25, \\ x > 0 \text{ bo'lganda, } \ln x. \end{cases}$$



4.6-rasm. Masala yechimining blok-chizmasi

4.6-rasmda murakkab tarmoqlangan jarayoni algoritmining blok-chizmasi keltirilgan. Bu erda tarmoqlanish uchta yo‘nalish bo‘yicha bo‘lishi mumkin: 2,4,7; 2,3,5,7; 2,3,6.

Masala 3. Berilgan uchta A, V, S sonlarning eng kichigi kvadratini topish algoritmining blok-chizmasini tuzing. Natijani R harfi bilan belgilaymiz (4.7-rasm).

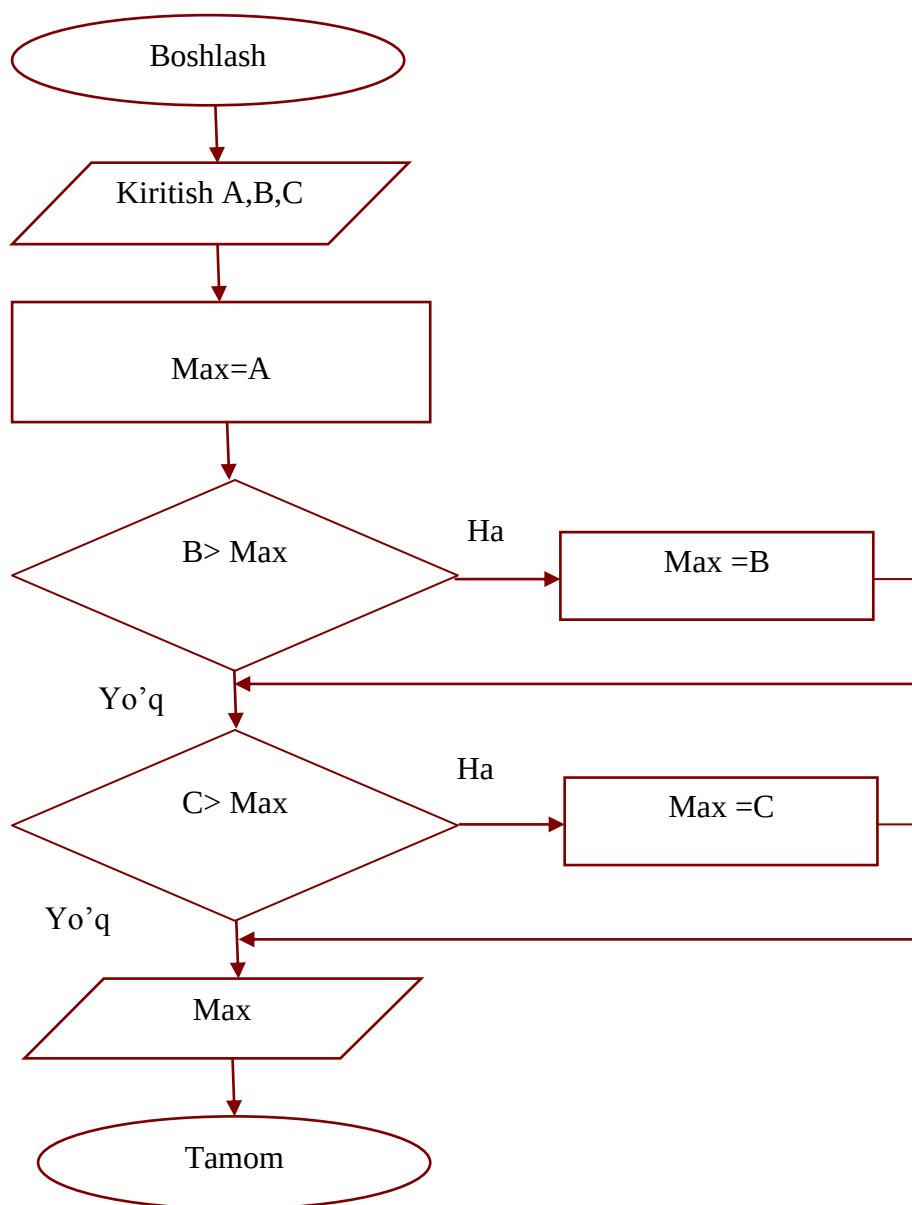


4.7-rasm. Masala yechimi algoritmining blok-chizmasi

Masala 4. Ucha son ichidan eng kattasini topish tuzing.

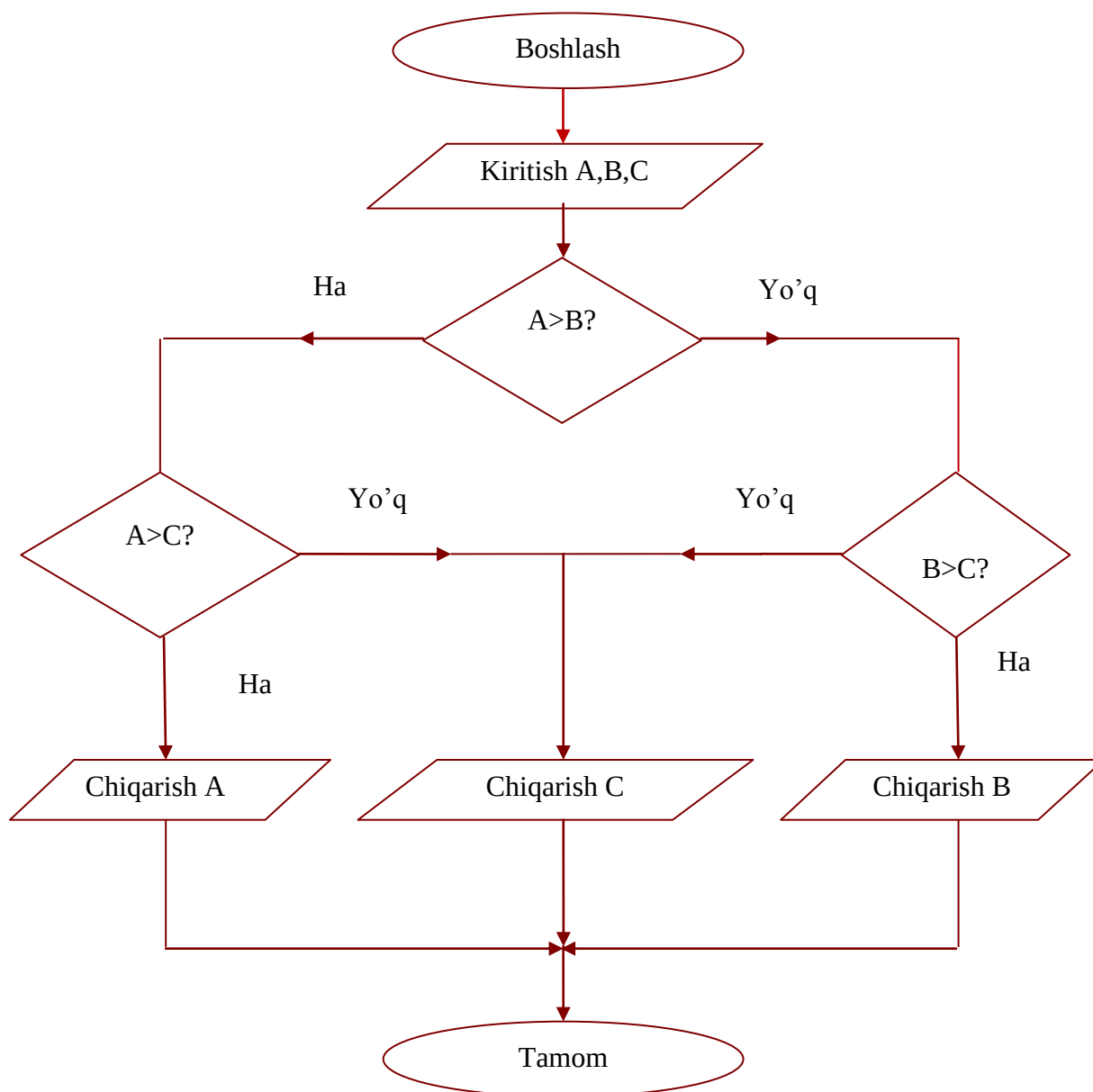
Masala yechimining algoritmini 2 hil usulda yechamiz.

Masala yechimining algoritmining blok-chizmasining 1 varianti 4.8. rasmda keltirilgan.



4.8-rasm. Uchta son ichidan eng kattasini toppish algoritmining blok-chizmasi (1-variant)

4.9-rasmda keltirilgan blok-chizmada uchta son ichidan eng kattasini topish 2-varianti aks etirilgan.



**4.9-rasm. Uchta son ichidan eng kattasini toppish algoritmining blok-chizmasi
(2-variant)**

Tayanch iboralar

tarmoqlangan hisoblash jarayoni, to‘liq alternativa, to‘liq emas alternativa, shart, mantiqiy ifoda, oddiy tarmoqlangan hisoblash jarayoni, murakkab tarmoqlangan hisoblash jarayoni, blok-chizma.

Nazorat va muhokama uchun savollar

1. Tarmoqlangan hisoblash jarayoni boshqa hisoblash jarayonlaridan nima bilan farq qiladi?
2. Tarmoqlangan hisoblash jarayonida qanday bloklardan foydalaniladi?
3. To'liq va to'liq emas tarmoqlangan algoritmlarini farqi va o'xshashlik tomonlari nimadan iborat?
4. Oddiy va murakkab tarmoqlangan hisoblash jarayonining farqi nimadan iborat?
5. Murakkab tarmoqlangan hisoblash jarayoniga misol keltiring.

5- bob. TAKRORLANUVCHI HISOBLASH JARAYONLAR

5.1. Takrorlanuvchi (siklik) hisoblash jarayonlari.

5.2. Vektor elementlari qiymatini yig'indisini topish masalasining blok-chizmasi.

5.3. Aniq berilgan son asosida oddiy sikllarni tashkil etish blok-chizmalari.

5.4. Takrorlanishlar soni aniq bo'lmagan sikllar.

5.5. Matritsalar (ikki o'lchamli massivlar) tushunchasi.

5.6. Matritsaning elementlar yig'indisini topish algoritmini tuzish.

5.7. Kvadrat matritsa elementlarini qayta ishlashning variantlari.

5.1. Takrorlanuvchi (siklik) hisoblash jarayonlari

Hisoblash bosqichlarini ko'p marta qaytarishi mumkin bo'lgan hisoblash jarayonlari takrorlanuvchi deyiladi. Sikllar shaxsiy kompyuterda masalalarni algoritmlashtirish va dasturlashtirishning asosini tashkil qiladi.

Hisoblash jarayonlarini ko'p marta qaytariladigan qismi **siklning tanasi** deyiladi.

Takrorlanuvchi algoritmlar quyidagilarga ajraladi:

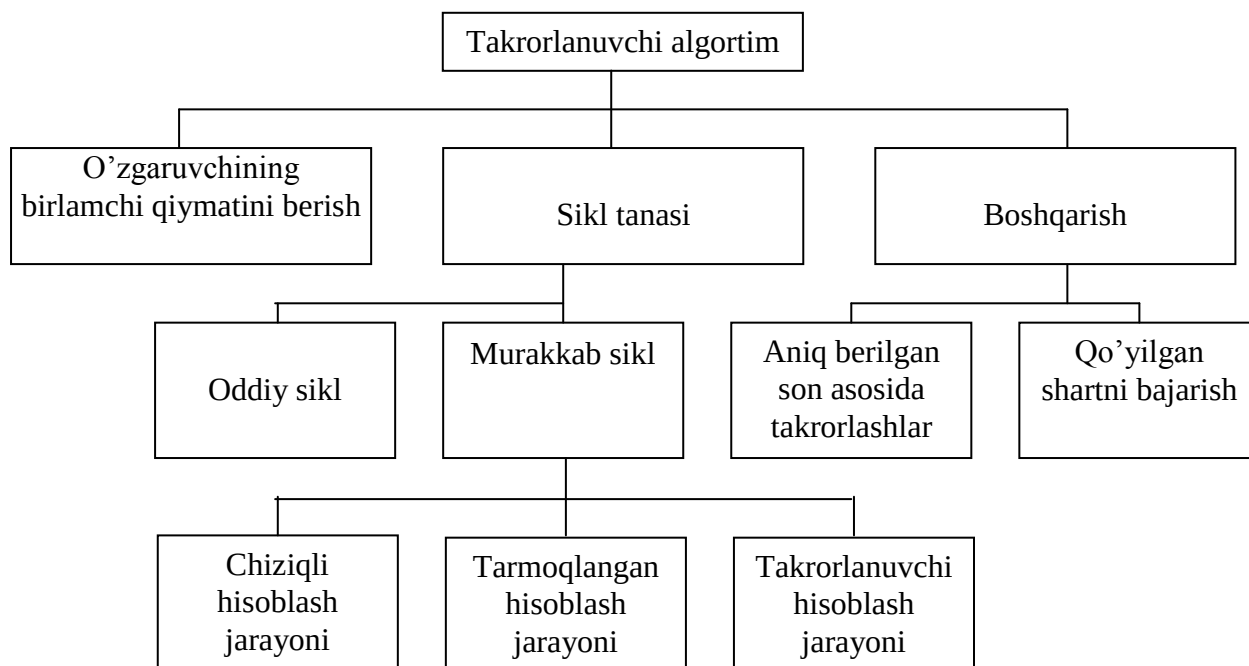
- 1) sikl tanasining algoritmining turi bo'yicha;
- 2) takrorlanuvchi jarayonni boshqarish xususiyatiga ko'ra.

Siklning tanasi algoritmi chiziqli, tarmoqlangan va takrorlanuvchi bo'lishi mumkin. Algoritmning turiga ko'ra siklning tanasidagi takrorlanuvchi algoritmlar oddiy va murakkab bo'lishi mumkin.

Agar siklning tanasi chiziqli yoki tarmoqlangan hisoblash jarayonlaridan iborat bo'lsa unda u **oddiy** takrorlanuvchi algoritm deyiladi.

Agar siklning tanasi takrorlanuvchi hisoblash jarayonlaridan iborat bo'lsa unda u **murakkab** takrorlanuvchi algoritm deyiladi.

Takrorlanuvchi algoritmning tasniflanish chizmasi 5.1- rasmda keltirilgan.



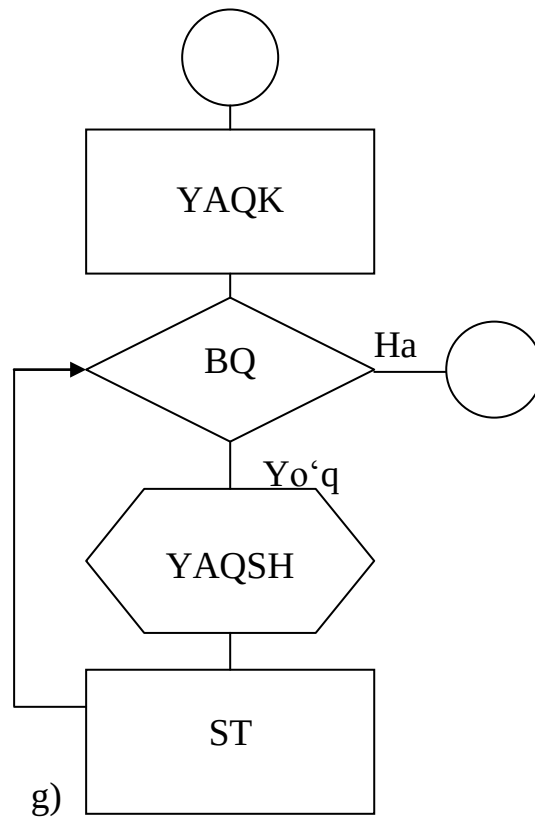
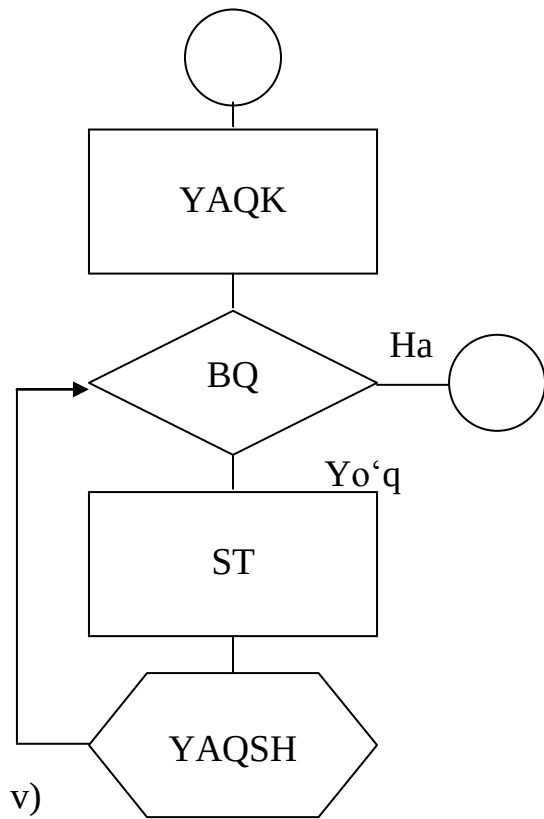
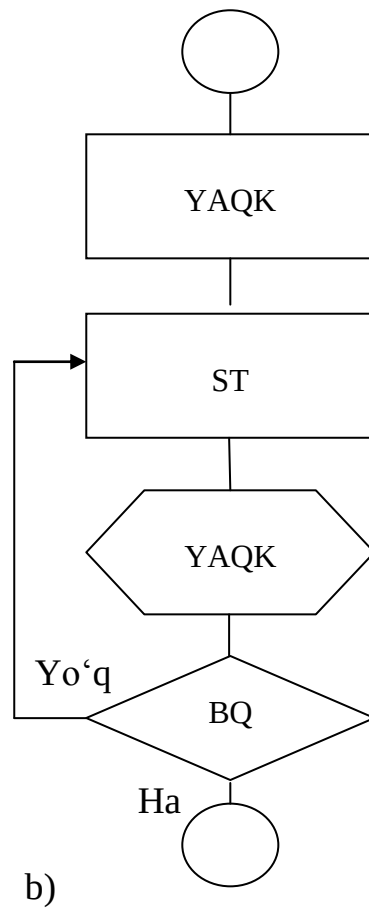
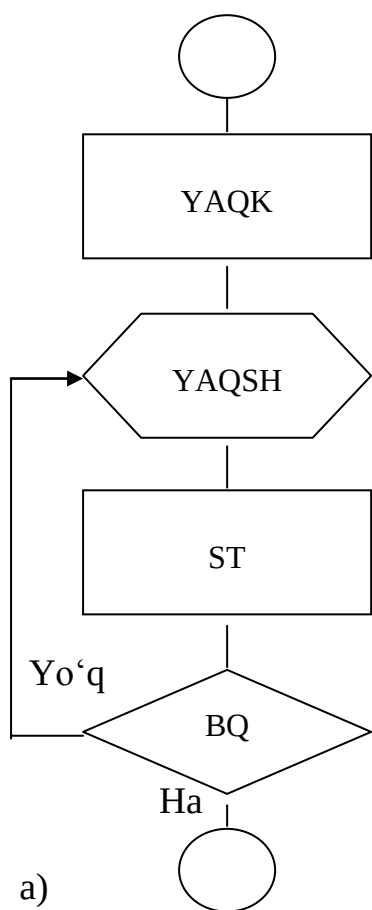
5.1-rasm. Takrorlanuvchi algoritmnining tasniflanishi¹⁹

Boshqaruvchi qismning xususiyatiga ko'ra siklning qaytarilishlar soni aniq bo'lgan bo'lishi va aniq bo'lmagan bo'lishi ham mumkin. Aniq bo'lgandan siklning necha marta qaytarilishi ko'rsatiladi, masalan 20, 50, 100. Aniq bo'lmaganda ma'lum bir natural son orqali ifodalanadi, masalan N, M, K.

Takrorlanuvchi algoritmnining asosiy qismlaridan bo'lib quyidagilar hisoblanadi: **siklning** tanasi (ST), boshqaruvchi qism (BQ), yangi qiymatlarni shakllantirish (YAQSH) va yangi qiymatlarni kiritish (YAQK). Ushbu qismlarning bajarilishi ketma-ketligi turlicha bo'lishi ham mumkin. Ularning ayrimlarini ko'rib chiqamiz.

Takrorlanuvchi jarayonlarni tashkil qilishning turli usullari mavjuddir. Quyidagi 5.2- rasmda ularning ayrimlaridan misollar keltirilgan.

¹⁹ R.Dadabayeva, Sh.Nasridinova, N.Shoaxmedova, L.Ibragimova, Sh.Ermatov. Axborot-kommunikatsion texnologiyalar va tizimlar.-T.: "Sano-standart" nashriyoti. O'quv qo'llanma. 2017. B. 262



5.2 – rasm. Takrorlanuvchi algoritmlarning asosiy qismlariga misollar²⁰

²⁰ R.Dadabayeva, Sh.Nasridinova, N.Shoaxmedova, L.Ibragimova, Sh.Ermatov. Axborot-kommunikatsion texnologiyalar va tizimlar.-T.: “Sano-standart” nashriyoti. O`quv qo`llanma. 2017. B. 263

5.2. Vektor elementlari qiymatini yig'indisini topish masalasining blok-chizmasi

Quyidagi formula asosida A vektor elementlari qiymatini yig'indisini topish masalasini ko'rib chiqamiz.

$$S = a_1 + a_2 + a_3 + \dots + a_{20} = \sum_{i=1}^{20} a_i$$

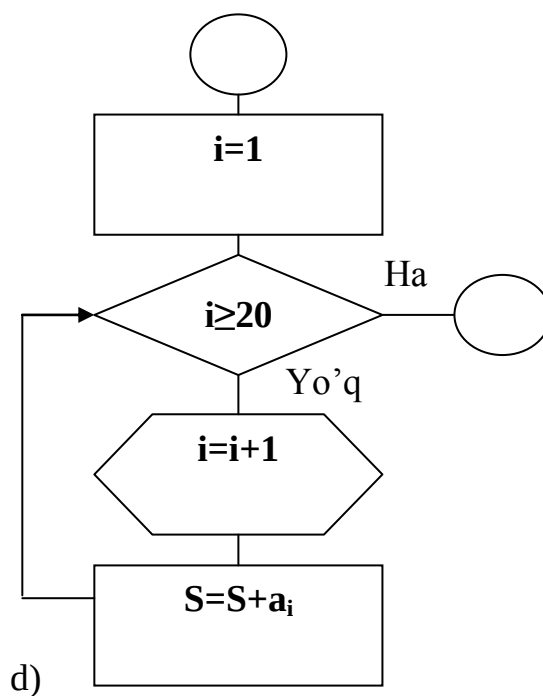
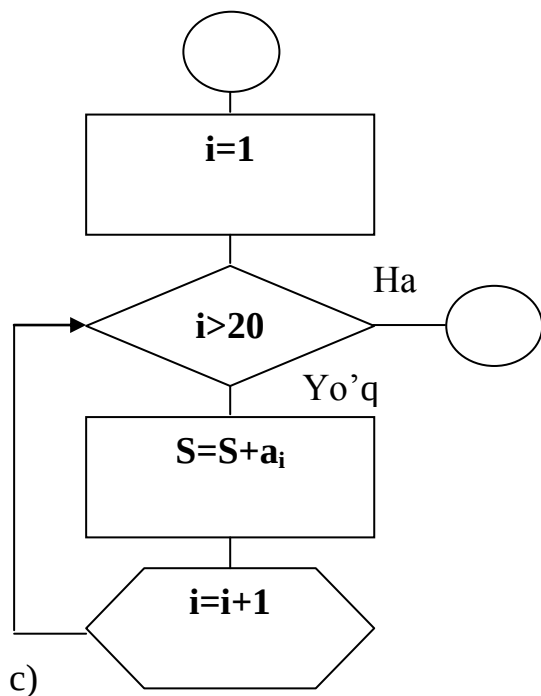
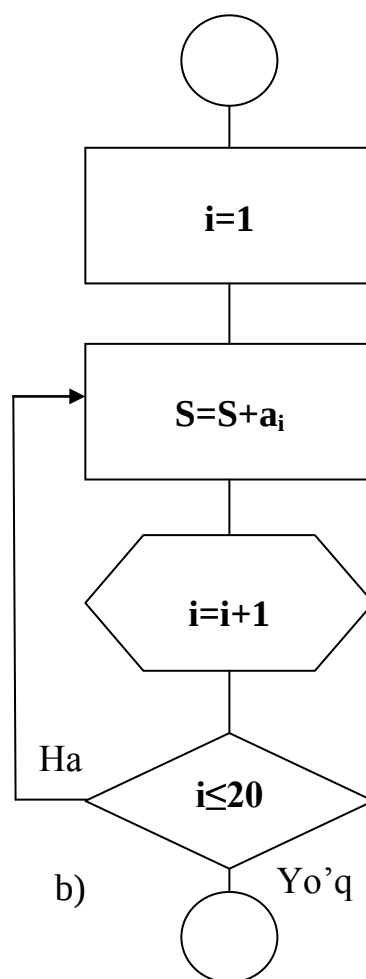
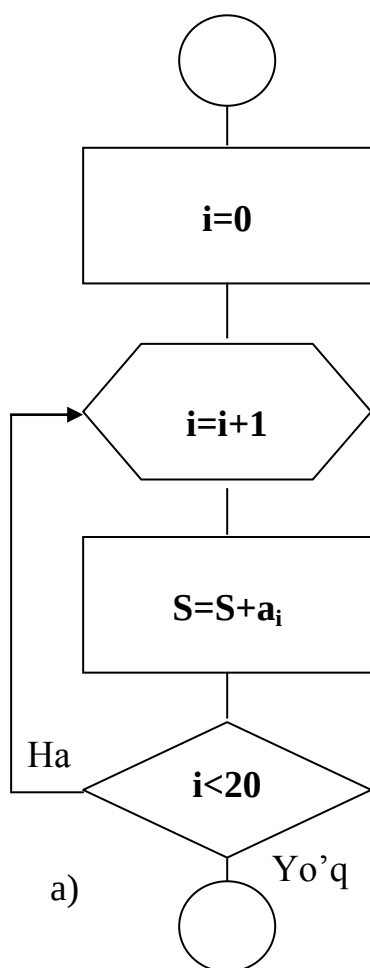
S-ni qiymatini topish uchun qo'shish operatsiyasini ko'pmarta (20-marta) takrorlash kerak bo'ladi. Ushbu operatsiyaning har bir bajarilishida oldingi natijaga vektorning keyingi yangi elementining qiymati qo'shib boriladi. Shunday qilib algoritmnining quyidagi qismi ko'pmarta bajariladi:

$$S = S + a_i$$

Bu erda i indeksning 1-dan 20-gacha 1-tadan qadam tashlab o'zgarib borishi bilan vektorning barcha elementlari ularning yig'indisini topish uchun ko'rib chiqiladi. Natijada takrorlanuvchi aylanishni o'zgaruvchan indeks i boshqaradi, shuning uchun ham uni siklning boshqaruvchi o'zgaruvchisi deyiladi yoki sikl parametridir.

Algoritmda siklning aylanishi bilan S-ning yig'indisini topish uchun u avvalom bor nolgacha tenglashtirilishi kerak. Aks holda olinishi kerak bo'lgan S bo'yicha natija noto'g'ri bo'lishi mumkin.

Quyidagi 5.3 rasmda yuqorida keltirilgan misolni yechishning ayrim blok-chizmalari keltirilgan.



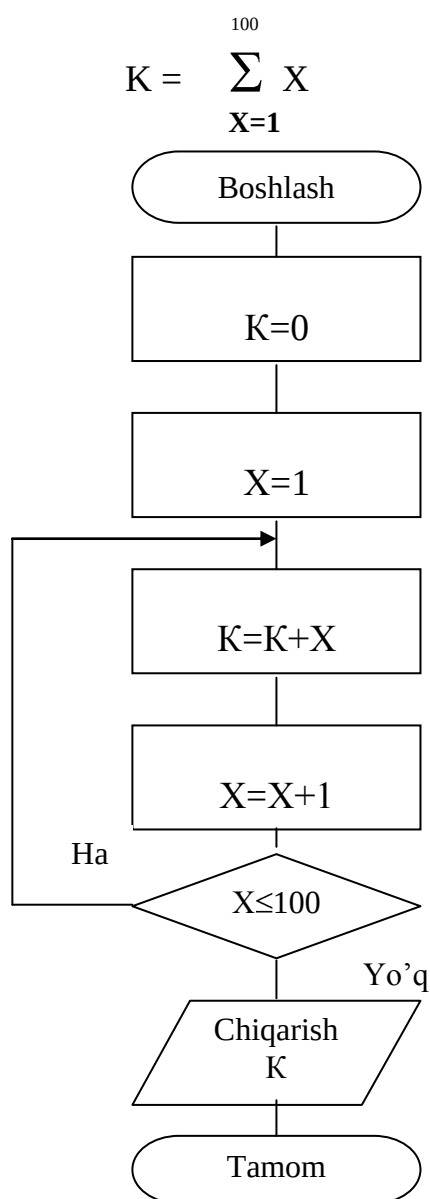
5.3 – rasm. Vektor elementlarining summasini aniqlash algoritmining blok-chizmasi²¹

²¹ R.Dadabayeva, Sh.Nasridinova, N.Shoaxmedova, L.Ibragimova, Sh.Ermatov. Axborot-kommunikatsion texnologiyalar va tizimlar.-T.: “Sano-standart” nashriyoti. O`quv qo`llanma. 2017.. B. 265

5.3. Aniq berilgan son asosida oddiy sikllarni tashkil etish blok-chizmalari

Aniq berilgan son asosida sikllarni tashkil qilishda sikl parametrining boshlang'ich va oxirgi qiymatlari, uning har bir takrorlanishidagi sikl parametrining o'zgarish qonunlari, siklning takrorlanish sonlari ko'rsatilishi kerak bo'ladi. Sikl tanasidagi birlamchi ma'lumotlar doimiy kattalik, oddiy o'zgaruvchan, indeksli o'zgaruvchan ko'rinishida bo'lishi mumkin. Agar sikl tanasidagi birlamchi ma'lumotlar doimiy kattalik ko'rinishida bo'lsa, sikl tanasi algoritmi shu kattalikka erishgancha aylanadi.

Masala 1. Ushbu ifodani hisoblash algoritmining blok-chizmasini tuzing.



5.4- rasm. 1-masalani yechish algoritmining blok-chizmasi

X-ning qiymati 1-dan 100-gacha o'zgaradi, ya'ni 1-100 gacha bo'lgan natural sonlarni jamlanish jarayoni amalga oshadi (5.4-rasm).

Bu yerda indeksli o'zgaruvchi ko'rilayotgan massiv elementi nomidir. Elementning massivdagi o'rnining indeksning qiymati bo'lmish elementlar ketma-ketligida joylashgan tartib raqami bilan aniqlanadi. Indeksning qiymati siklning takrorlanish jarayonida o'zgaradi. Sikl tanasining algoritmi massivdagi barcha elementlarga nisbatan qo'llaniladi.

Masala 2. Ushbu ifodani hisoblash algoritmining blok-chizmasini tuzing.

$$S = \sum_{i=1}^{50} a_i - b^2$$

Ushbu misolda (6.5-rasm) siklning parametri arifmetik progressiya qonuniga asosan 1-qadam tashlab o'zgarib boradi, uning boshlang'ich va yakuniy qiymati 1 va 50 tashkil qiladi. 3-blokda siklni tashkil qilish amalga oshadi, masalan i birlamchi qiymanga ega bo'ladi. 4-blokda esa hisoblashning takrorlanuvchi qismi beriladi. 5-chi blokda esa sikl parametri qabul qilingan qonun bo'yicha o'zgaradi. Bu blok siklning hisobchisi bo'ladi. 6-chi blokda parametrning hozirgi holatdagi qiymati uning yakuniy bo'lishi mumkin bo'lgan qiymati bilan solishtirish asosida siklning tugaganligi tekshiriladi. Agar ushbu holat saqlanib qolsa, u holda A vektorning barcha elementlari qiymati jamlanmaganligini bildiradi. Shuning uchun ham boshqarish jarayoni siklni takrorlanishiga berilib, unda S -da yig'ilgan vektor elementlari qiymati jamlamasiga vektorning navbatdagi elementi qiymati qo'shiladi. Agar $i \leq 50$ bajarilmasa, u holda sikldan chiqishga olib keladi. Bu A vektorning barcha 50-ta elementlari qiymati S -ga jamlanganligini ko'rsatadi.

Takrorlanuvchi jarayonlarni tashkil etishda masalani yechish algoritmini to'g'ri tuzilganligi katta ahamiyat kasb etadi. Algoritmida keltirilgan blok-chizmani to'g'ri bajarilishini tekshirish uchun undagi har bir blokning to'g'ri bajarilishini tekshirib chiqish kerakdir va ularning majmuasi bitta butun chizmani tashkil etadi. Bunday

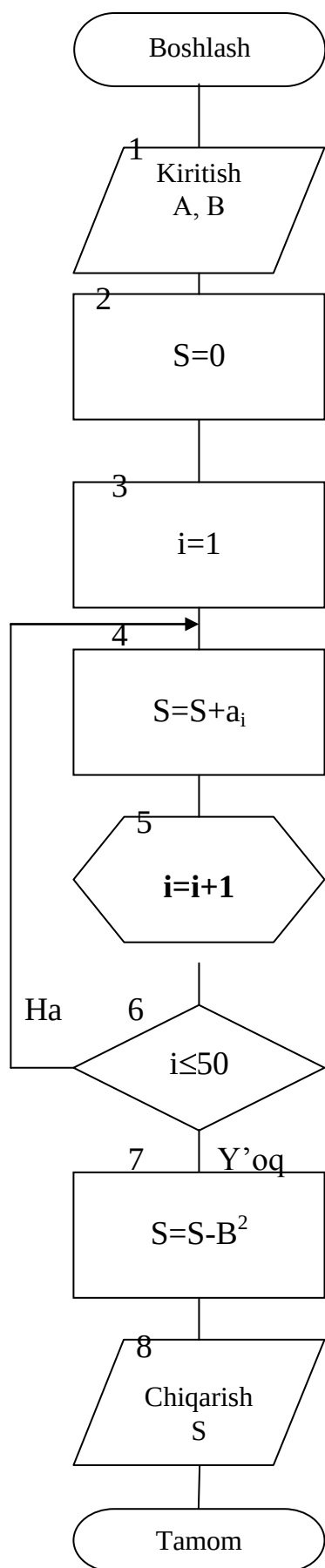
tahlillar natijasi masalani yechishning mantiqi bilan solishtiriladi. Agar mantiq jihatdan o‘zaro qarama-qarshiliklar bo‘lmasa, bunday blok-chizma asosida masalaning dasturini tuzish mumkin bo‘ladi. Algoritmning mantiqiy jihatdan to‘g‘ri ishlashini birorta misol asosida tekshirib ko‘rish mumkin, ya‘ni oltita 8,7,3,2,5,1 sonlarni summasini topish bilan tekshiramiz. Algoritm tahlillari barcha natijalarini 5.1-jadval ko‘rinishida keltiramiz.

5.1-jadval

2-masalani yechish algoritmining bajarilishi ketma-ketligi natijalari

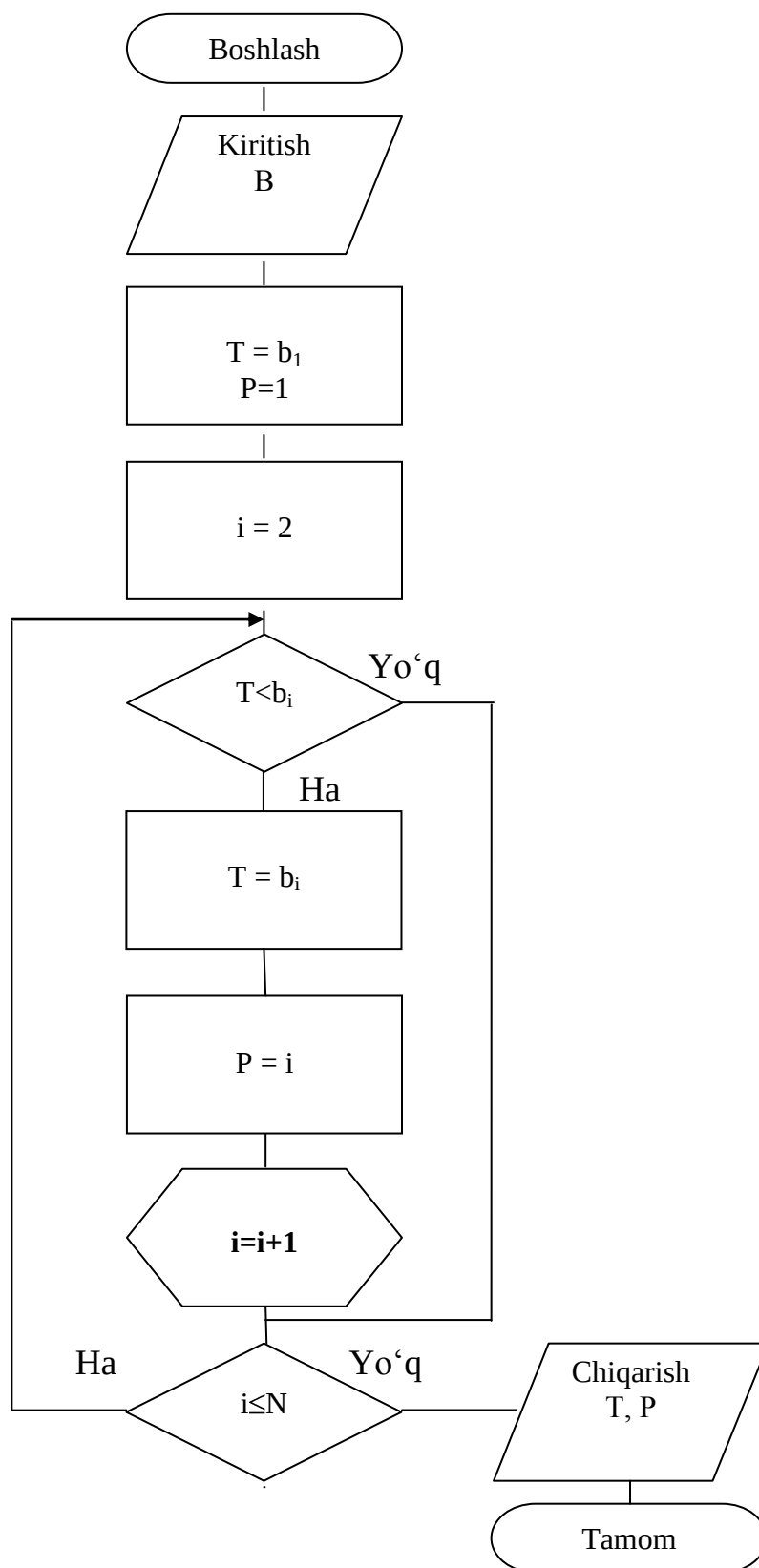
Sikl parametri qiymati	$S = S + a_i$	$i = i + 1$	$i \leq 6$	Boshqaruvni qabul qiluvchi blok
1	$S = 0 + 8$	2	$2 \leq 6(\text{Ha})$	4
2	$S = 0 + 8 + 7$	3	$3 \leq 6(\text{Ha})$	4
3	$S = 0 + 8 + 7 + 3$	4	$4 \leq 6(\text{Ha})$	4
4	$S = 0 + 8 + 7 + 3 + 2$	5	$5 \leq 6(\text{Ha})$	4
5	$S = 0 + 8 + 7 + 3 + 2 + 5$	6	$6 \leq 6(\text{Ha})$	4
6	$S = 0 + 8 + 7 + 3 + 2 + 5 + 1$	7	$7 \leq 6(\text{Yo‘q})$	7

5.1-jadvaldan ko‘rinib turibdiki, ko‘rilayotgan blok-chizmadagi harakatlar qo‘yilgan masalani to‘liq yechilishini ta‘minlaydi.



5.5- rasm. 2-masalani yechish algoritmining blok-chizmasi

Masala 3. N elementlardan iborat bo'lgan vektor berilgan. Ushbu vektorning maksimal elementini va uning koordinatasini topish kerak. T - orqali maksimal elementni belgilaymiz, R – bilan esa uning koordinatasini.



5.6 - rasm. 3 masalani yechish algoritmining blok-chizmasi

Bu masalani yechishni 5.6-rasm orqali amalga oshiramiz. Boshlanishida T-ga vektorning birinchi elementi qiymati beriladi, R –ga birinchi elementning koordinatalari kiritiladi, ya'ni 1 bo'ladi. Keyin T sikl ichida ko'rilayotgan b_i vektorning qolgan barcha elementlari bilan taqqoslanib chiqiladi. Agar har bir taqqoslanganda $T < b_i$ bajarilsa, u holda T b_i element qiymatini oladi va R esa uning koordinatasiga teng bo'ladi. Aks holda T va R-larning qiymatlari saqlanib qoladi va taqqoslash uchun V vektorning keyingi elementi olinadi. Vektorning barcha elementlari ko'rib chiqilgandan so'ng T maksimal element qiymatini va R esa uning koordinatalariga teng bo'ladi. $T < b_i$ bajarilgandagina takrorlanuvchi blokning quyidagi 5-6 bloklar ishlaydi.

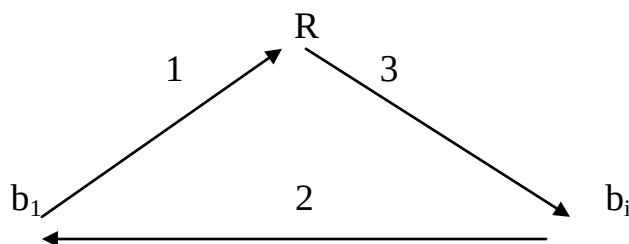
Yuqorida keltirilgan algoritmnı mantıqiyy jihatdarn ishlashini ettita elementdan 5,12,13,10,2,28,4 iborat bo'lgan vektor misolida ko'rib chiqamiz. Bu erdan ko'rinib turibdiki, dastlabki qiymat $T=5$. Takrorlanuvchi jarayon esa $i=2$ boshlanadi (5.2-jadval).

5.2 – jadval

3-masalani yechish algoritmining bajarilishi ketma-ketligi natijalari

Sikl parametri qiymati	$T < b_i$	T	R	$i=i+1$	$i \leq 7$	Boshqaruvni qabul qiluvchi blok
2	$5 < 12$ (Ha)	12	2	3	$3 \leq 7$ (Ha)	4
3	$12 < 13$ (Ha)	13	3	4	$4 \leq 7$ (Ha)	4
4	$13 < 10$ (Yo'q)			5	$5 \leq 7$ (Ha)	4
5	$13 < 2$ (Yo'q)			6	$6 \leq 7$ (Ha)	4
6	$13 < 28$ (Ha)	28	6	7	$7 \leq 7$ (Ha)	4
7	$28 < 4$ (Yo'q)			8	$8 \leq 7$ (Yo'q)	8

Faraz qilaylik vektorning maksimal elementi va uning koordinatasini aniqlab olingandan keyin, uni vektorning birinchi elementi bilan almashtirish kerak bo'lsin. Vektor elementlarini o'rnini almashtirishni 5.7-rasm orqali amalga oshirish mumkin:



5.7-rasm. Vektor elementlarini o'rnini almashtirish chizmasi

Quyidagi operatorlar bilan amalga oshiriladi: $P = b_1, b_1 = b_p, b_p = P$. Ushbu misolda elementlarni o'rnini almashtirishni quyidagicha ham amalga oshirish mumkin: $b_p = b_1; b_1 = T$.

5.4. Takrorlanishlar soni aniq bo'lmagan sikllar

Bunday sikllarda takrorlanishlar sonini oldindan belgilab bo'lmaydi. Unda sikl ma'lum bir shart bajarilishigacha davom etadi.

Takrorlanishlar soni aniq bo'lmagan sikllarga interatsion sikllarni misol qilish mumkin bo'lib, unda sikldan chiqib ketish aniq bir natijaga erishilgandan keyin amalga oshiriladi. Hisoblashlar erishish kerak bo'lgan natijaga bosqichma-bosqich yaqinlashib borish orqali bajariladi (iteratsiya usulidir).

Masala 4. Quyida keltirilgan matematik ifodani hisoblash algoritmi blok-chizmasini tuzing.

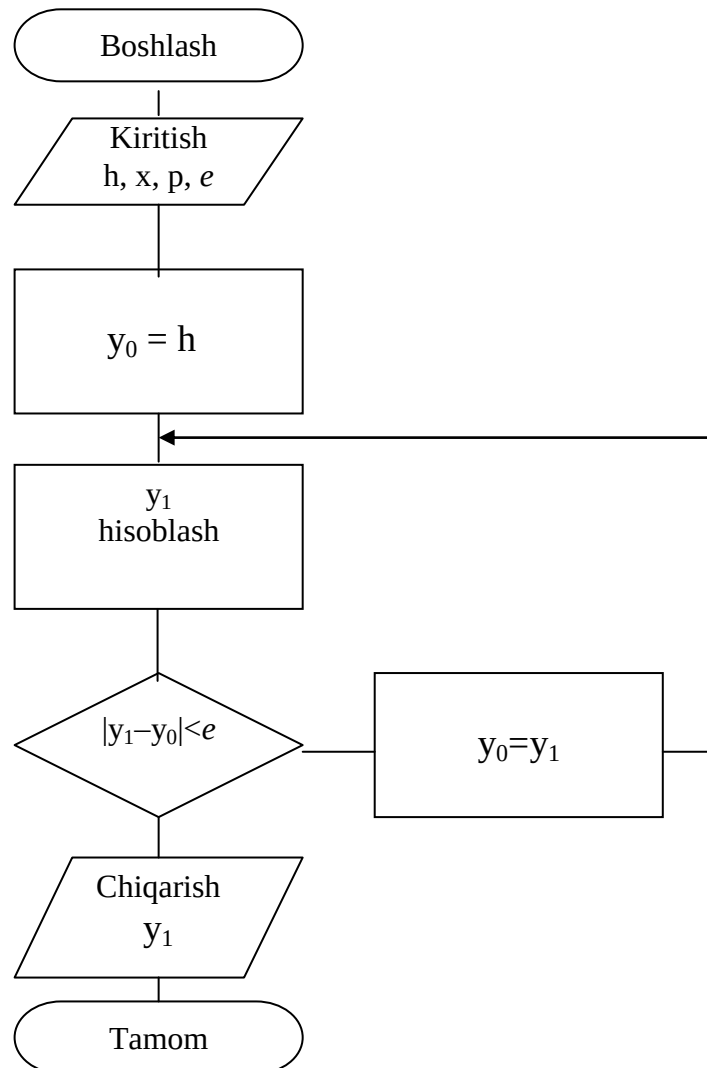
$$y = \sqrt[p]{x}$$

$$y_1 = 1/p [(p-1) y_0 + (x/y_0^{p-1})]$$

Hisoblash xatoligi $|y_1 - y_0| < e$, dastlabki yaqinlashish $y_0 = h$

berilgan e qiymatiga bosqichma-bosqich bilan yaqinlashib borish asosida berilgan formula ildizining qidirilayotgan qiymati topiladi.

Hisoblash jarayonlari amalga oshirilguncha, siklning takrorlanishlar soni noma'lumdir (5.8-rasm).



5.8-rasm. Takrorlanishlar soni aniq bo'lmagan sikllar bo'yicha algoritmnining blok-chizmasi

Siklni boshqarilishini hisoblashlar xatoligi bo'lmish e belgilaydi. Agar keyingi iteratsiyada xatolik $\geq e$ bo'lsa, u holda sikl keyingi yaqinlashib borish natijasida y_1 qiymatini hisoblaydi, aks holda sikldan chiqib ketiladi. Xatolik $|y_1 - y_0| \geq e$ bo'lganda sikl davom etadi. 5 blokda oldingi iteratsiya asosida olingan y_1 natija keyingi iteratsiyada olingan bilan almashtiriladi. Undan keyin boshqaruv jarayoni 3 blokka berilib, unda ildiz qiymatining talab qilinayotgan natijalarga yaqinlashishi hisoblanadi. $|y_1 - y_0| < e$ aniqlikka erishilgandan so'ng sikldan chiqib ketilib, 4 blokdan 6 blokka boshqarish beriladi.

Shuni aytish joizki, algoritm asosida turli xildagi iqtisodiy va ijtimoiy masalalarni ifoda etish mumkin. Bu esa o'sha hal qilinishi mumkin bo'lgan masalani tizimli yondashuv asosida kompleks ravishda echishni ta'minlab beradi.

5.5. Matritsalar (ikki o'lchamli massivlar) tushunchasi

Matritsalar (ikki o'lchamli massivlar) bilan ishlash uchun, avvalom bor undagi o'zgaruvchilarning indekslarini o'zgarib borish tartibi bilan yaqindan tanishib chiqish kerak. $A\{i=1,N; j=1,M\}$ matritsaning umumiy ko'rinishi quyidagicha bo'ladi:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1m} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nm} \end{pmatrix}$$

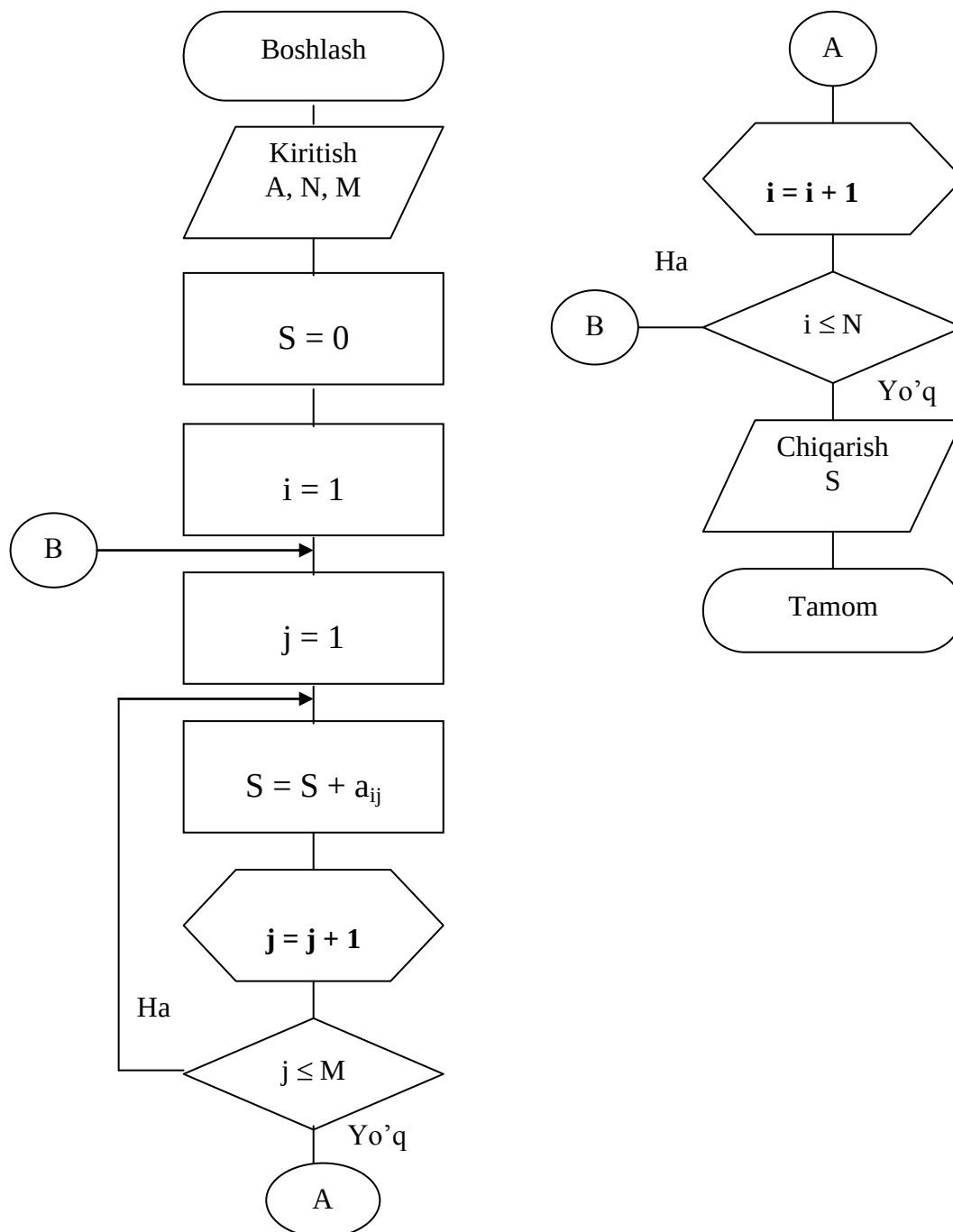
Keltirilgan matritsadan ko'rinib turibdiki, uning har bir elementi i va j indeksga egadir. Bu erda i elementning matritsani qaysi qatorida joylashganini bildiradi, j esa ushbu elementning qaysi ustundagiligini ko'rsatadi. SHKning xotirasida esa matritsa elementlari qatorma-qator joylashadi, ya'ni birinchi qator elementlaridan keyin ikkinchi qatorniki ketadi va h.k. Matritsaning elementlariga murojaat qilib, ularni barchasini ko'rib chiqish uchun ikkita indeksni ko'rsatish kerak, ya'ni oldin qator raqamini, keyin esa ustun raqamini va uni chegarab qo'yilgan m -gacha o'zgartirib boriladi, undan keyin esa qatorning raqamini bittaga ko'paytiriladi. O'z o'rnida j ustunlar raqamining o'zgarish qonuniyatini 1-dan m -gacha qaytarish kerak, ya'ni i chegarab qo'yilgan n -gacha etganicha. Indeksarning o'zgarib borish tartibini quyidagi misol ko'rinishida aks ettirish mumkin.

5.6. Matritsaning elementlar yig'indisini topish algoritmini tuzish

Masala 1. Berilgan matritsa a_{nm} . Matritsa elementlarining yig'indisini topish algoritmnining blok-chizmasini tuzing.

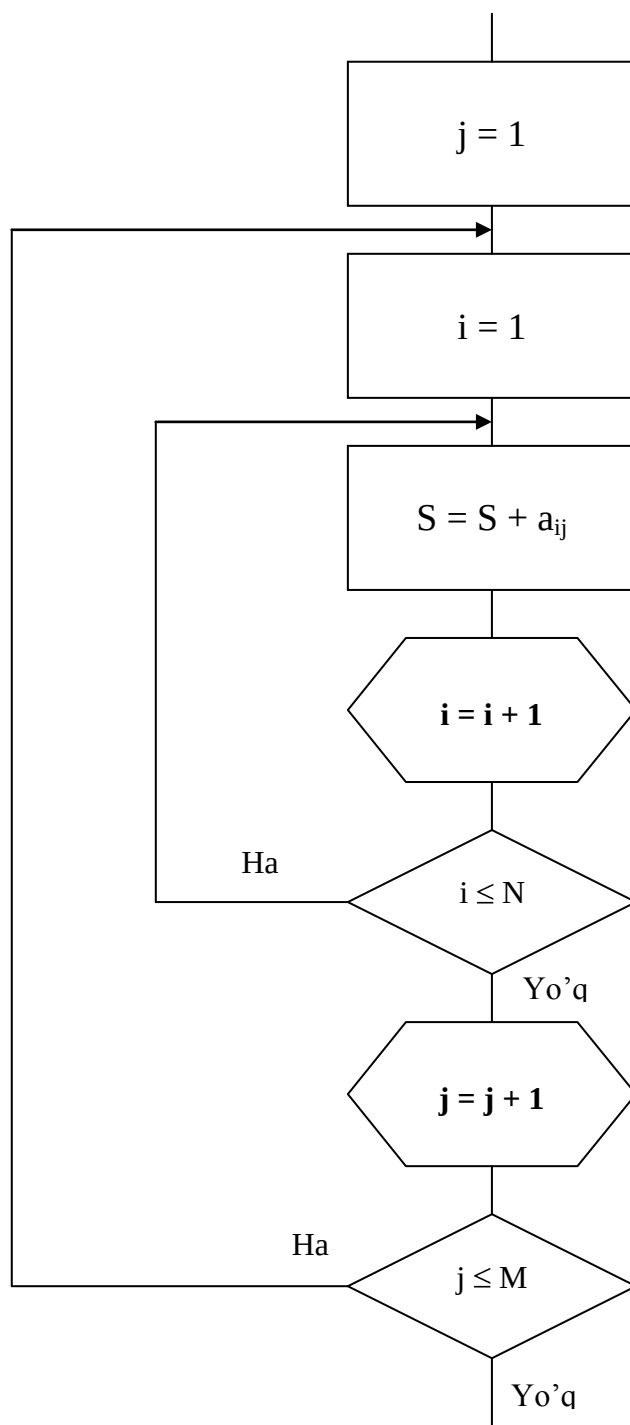
$$S = \sum_{i=1}^N \sum_{j=1}^M a_{ij}$$

Masala yechimi algoritmining blok-chizmasi 5.9-rasmda keltirilgan.



5.9-rasm. Matritsa elementlarining yig'indisini topish algoritmnining blok-chizmasi

Matritsaning ustunlari bo'yicha elementlarni ketma-ket ko'rib chiqish uchun undagi elementlarni quyidagi tartibda almashtirish kerak (5.10-rasm).



5.10 – rasm. Matritsa elementlarining yig'indisini topish algoritmining blok-chizmasi (lavha)

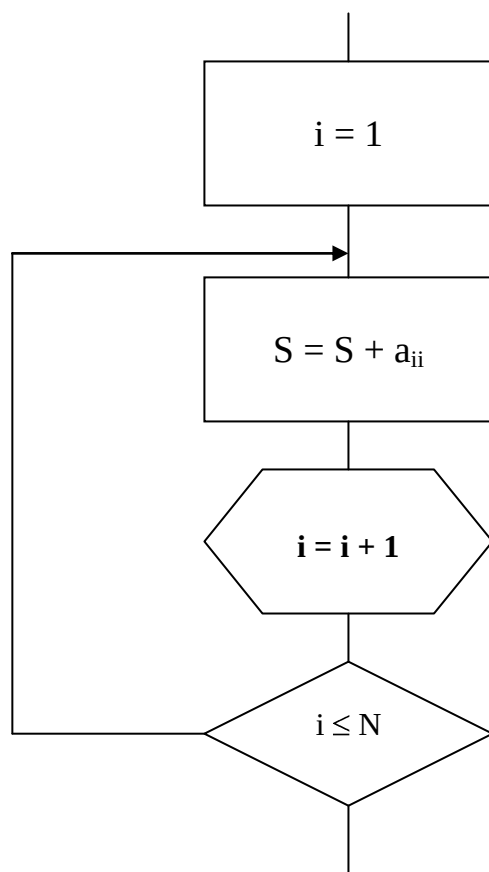
5.7. Kvadrat matritsa elementlarini qayta ishlashning variantlari

Agar $n = m$ bo'lsa, ko'rilayotgan matritsa kvadrat hisoblanib, qatorlar soni ustunlar soni bilan teng bo'ladi. Kvadrat matritsa elementlarini qayta ishlashning quyidagi variantlarini ko'rib chiqamiz:

1. Asosiy diagonalda joylashgan elementlar bilan ishlash.
2. Asosiy diagonal yuqorisida joylashgan elementlar bilan ishlash.
3. Asosiy diagonal pastida joylashgan elementlar bilan ishlash.

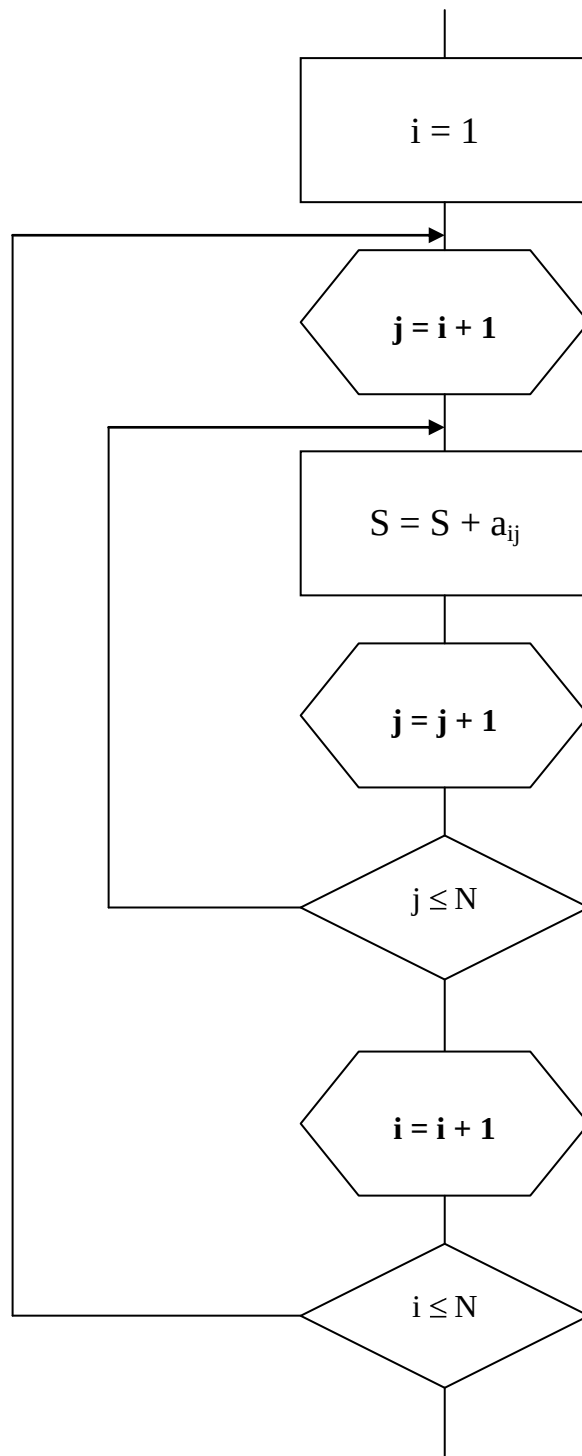
Yuqorida keltirilgan variantlarni quyidagi sxemalar ko'rinishida keltiramiz:

1. Asosiy diagonalda joylashgan elementlar bilan ishlash (5.11-rasm). Asosiy diagonalda joylashgan elementlar bilan ishlash algoritmlarining blok-chizmasida faqat bitta sikl ochiladi. Asosiy diagonalda joylashgan elementi a_{ii} indexsiga ega.



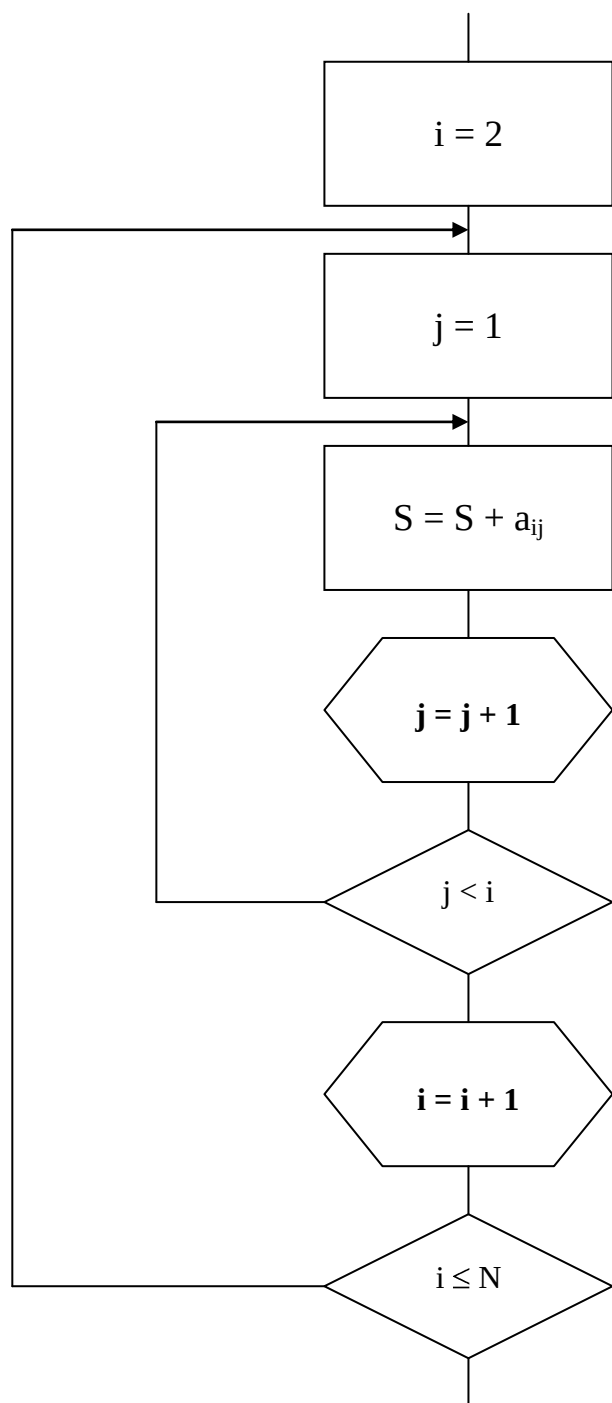
5.11-rasm. Asosiy diagonalda joylashgan elementlar bilan ishlash (lavha)

2. Asosiy diagonal yuqorisida joylashgan elementlar bilan ishlash (5.12-rasm).



5.12-rasm. Asosiy diagonal yuqorisida joylashgan elementlar bilan ishlash (lavha)

3. Asosiy diagonal pastida joylashgan elementlar bilan ishlash (5.13-rasm).



**5.13-rasm. Asosiy diagonal pastida joylashgan elementlar bilan ishlash
(lavha)**

Tayanch iboralar

takrorlanuvchi hisoblash jarayoni, sikl, siklning tanasi, blok-chizma, oddiy takrorlanuvchi algoritm, murakkab takrorlanuvchi algoritm, boshqaruvchi qism, vektor, element, massiv, kvadrat matritsa, asosiy diagonal, sikl, index.

Nazorat va muhokama uchun savollar

1. Takrorlanuvchi hisoblash jarayoni boshqa hisoblash jarayonlaridan nima bilan farq qiladi?
2. Takrorlanuvchi hisoblash jarayonida qanday bloklardan foydalaniladi?
3. Sikl tanasi deb nima ataladi?
4. Aniq berilgan son asosida takrorlashlarning oddiy sikllarining blok-chizmalari bu qanaqa sikllar?
5. Takrorlanishlar soni aniq bo'lmagan sikllar nima bilan farqlanadi?
6. Ikki o'lchovli massivlar qanday xususiyatlarga ega?
7. Kvadrat matritsa deganda siz nimani tushunasiz?

6-bob. MA`LUMOTLARNI SARALASH ALGORIMLARI

6.1. Ma`lumotlarni saralashning mohiyati.

6.2. Oddiy tanlov saralash metodi.

6.3. Oddiy almashuv saralash metodi.

6.4. Qo'shni elementlar juftliklarini izchil tartibga solinishi.

6.1. Ma`lumotlarni saralashning mohiyati

Saralash – informatika va hisoblash matematikasining eng muhim vazifasi bo'lib, u algoritmik masalalarning yechishning ko'p resurslar talab qiluvchi usullariga kiradi.

Ma`lumotlarni saralash – bu axborotlarni ishlab chiqish bo'lib, buning natijasida uning elementlari (yozuvlari) bu axborotlarning ba'zi bir alomatlarining miqdoriga ko'ra belgilangan izchillikda joylashadilar.

Ma`lumotlarni saralash axborotlarning katta massivlarini ishlab chiqish bilan bog'liq masalalarni yechishning davomiyligini ko'p marta qisqartirishga imkon beradi. Elementlar navlarga ajratilgach, telefon ma`lumotnomasidagi kabi, ularni topish, yangilash, chiqarib tashlash oson, qanday elementlar o'tkazib yuborilganligini ham qidirib topish oson.

Har qanday saralashdan ma`no elementlarni aniq berilgan tartibdagi izchillikda qayta joylashtirishdan iboratdir. Tartibga solish dastlabki massivni ko'p martalab ko'rib chiqish jarayonida amalga oshiriladi.

Saralash qaerda, kompyuterning ichki operativ xotirasidami, yoki ma`lumotlarning tashqi manbalaridami, amalga oshirilishiga ko'ra, ichki va tashqi saralash metodlari ajratiladilar. Biz faqat ichki saralash usullari ko'rib chiqiladi.

Saralashning har qanday vaziyatdagi eng yaxshi universal algoritmi mavjud emas. Ko'pgina eng yaxshi metodlar mavjud, ammo bu faqat qanday kompyuterda va qanday maqsadda navlarga ajratilishi ma`lum bo'lgan hollardagina. Algoritmning samaradorligi ko'pgina omillarga bog'liq bo'ladi:

- saralashda qancha elementlar ishtirok etmoqda;

- elementlar qanday darajada navlarga ajratilib bo'lganlar;
- navlarga ajratilayotgan elementlarning miqdorlari diapazoni va taqsimlanishi qanday;
- elementlarni davriy ravishda chiqarib tashlanishi va to'ldirilishi ko'zda tutilganmi;
- elementlarni parallel ravishda solishtirish mumkinmi?

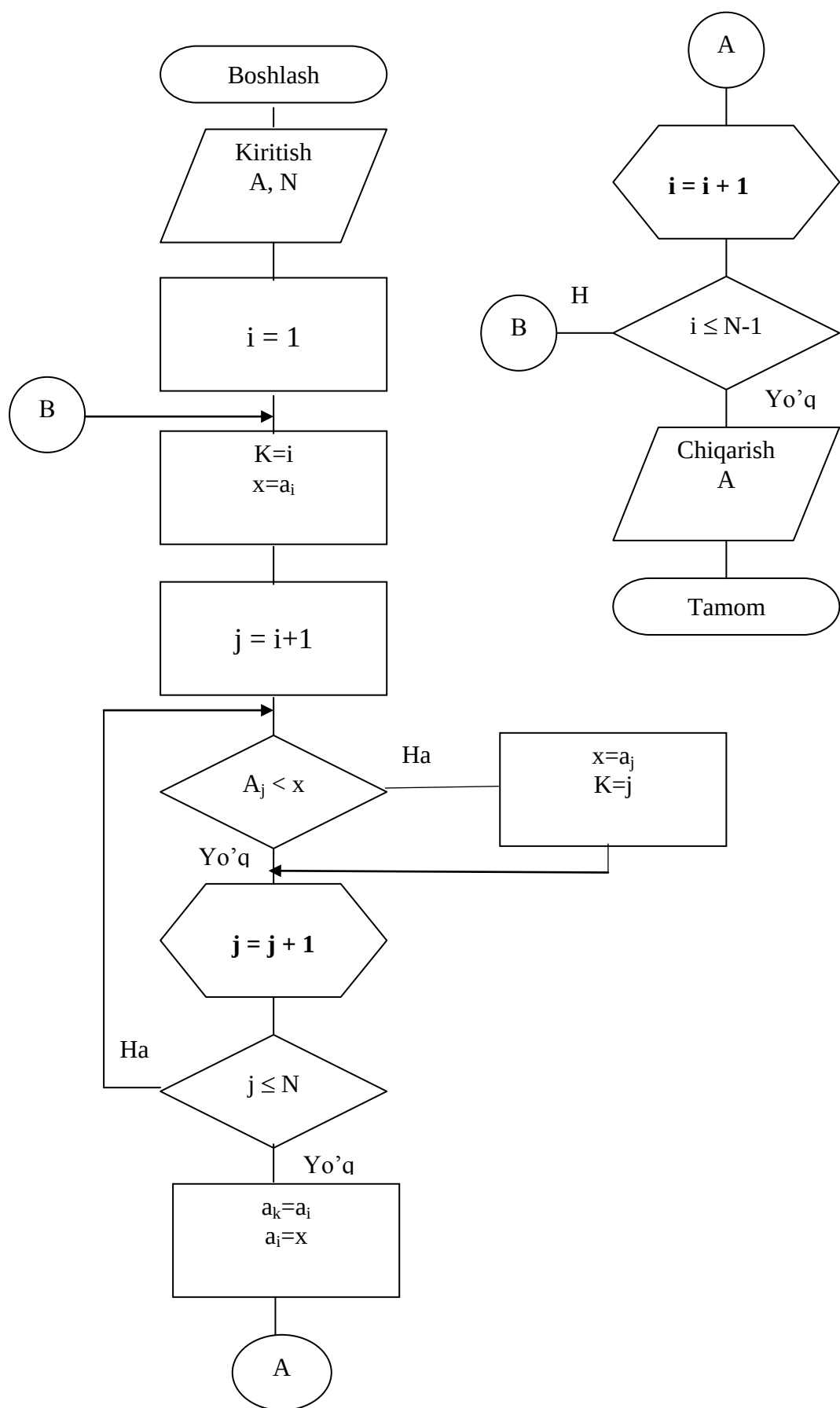
Agar teng miqdorlarga ega elementning nisbiy tartibi tartibga solingandan keyin o'zgarmasa, saralash metodi barqaror bo'ladi. Saralash algoritmlarini baholash uchun odatda vaqtning navlarga ajratilayotgan elementlarning N miqdoriga vazifaviy bog'liqligidan foydalaniladi. Saralashning asosiy metodlarini ko'rib chiqamiz. Algoritmlarni ishlab chiqishda tegishli izohlarga ega dastlabki ma'lumotlarni bosmaga chiqarish tavsiya etiladi, bu masala yechimining ko'rgazmaligini oshirishga imkon.

6.2. Oddiy tanlov saralash metodi

Oddiy tanlov metodining g'oyasi massivning birinchi elementidan boshlab to massivning oxirigacha bo'lgan eng kichik (eng katta) elementini izchil qidirish va birinchi elementni topilgan miqdorga almashtirishdan iborat bo'ladi. Birinchi element eng kichik elementning o'rniga qo'yiladi. Keyin ikkinchi elementni ko'rib chiqamiz va yana uchinchi boshlab izchillikdagi eng kattasini topamiz. Keyin ularni o'rnini almashtiramiz. Tanlov massivning oxirigidan oldingi elementgacha davom etadi.

Masala. $A(i); i = 1, \dots, N$ massivi berilgan. Uni o'sib borish bo'yicha oddiy tanlov metodi yordamida saralash.

Yechim. Algoritm kiritilgan tuzilmaning ikkita sikldan iborat. Tashqi siklda i (tashqi sikl parametri) 1 dan $N - 1$ gacha o'zgaradi.



6.1-rasm. Oddiy tanlov bilan saralash algoritmining sxemasi

Ichki siklda j (ichki sikl parametri) 1 qadamga ega $i + 1$ dan N gacha o'zgaradi. Bu siklda massiv elementlari orasidagi $i + 1$ dan boshlab N gacha eng kichik miqdor $A(j)$ tanlab olinadi. Shuning uchun ichki sikl boshlanishidan oldin massivning eng kichik elementining boshlang'ich miqdori va uning nomeri beriladi. Ichki sikl tugagandan keyin i -element va topilgan eng kichik element joylarini almashtiradilar.

O'sish bo'yicha oddiy tanlov ma'lumotlarini saralash algoritmining sxemasi 6.1-rasmda berilgan.

Bu algoritim o'rtacha $N(N - 1) / 2$ solishtirishlar va $3(N - 1)$ o'zlashtirilishlarda bajariladi.

6.3. Oddiy almashuv saralash metodi

Ushbu metodning g'oyasi shundan iboratki, agar ikkita yonma yon turuvchi elementlar tartib bo'yicha joylashmagan bo'lsalar, unda ular joylarini almashtiradilar. Bu jarayon elementlar tartibga solinmaguncha takrorlanaveradi. Boshqachasiga bu metod ko'pikchali saralash metodi (yoki oddiygina ko'pik metodi) deb ataladi.

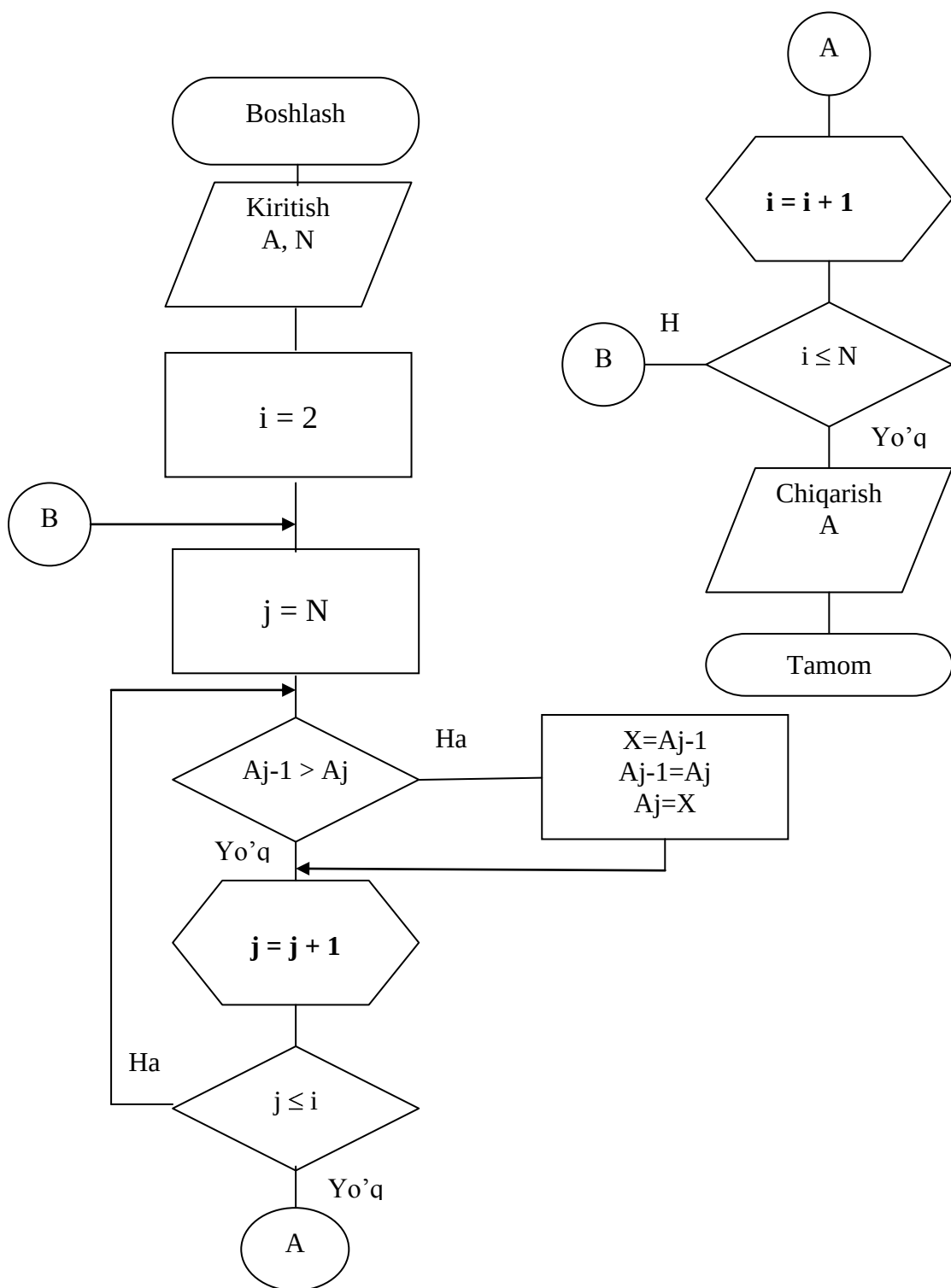
Masala. $A(i); i = 1, \dots, N$ massivi berilgan. Uni o'sish bo'yicha oddiy almashuv metodi bo'yicha navlarga ajrating.

Yechim. Saralashning mohiyati massivning yonma-yon turuvchi elementlarini ko'p martalab solishtirish va bu elementlarni o'sib borish tartibi bo'yicha qayta joylashtirishdan iboratdir. Shunday qilib, qo'shni A_1 va A_2 , A_2 va A_3 , A_3 va A_4 , ... elementlar navbatma navbat solishtiriladilar. Va agar $A_{j-1} > A_j$ bo'lsa, unda elementlar joylarini almashtiradilar.

Algoritim kiritilgan tuzilmaning ikkita siklga ega. Tashqi sikl i (tashqi sikl paramertrining) qiymati 2 dan N gacha o'zgaradi.

Ichki sikl j (ichki sikl paramertrining) qiymati N dan i gacha teskari tartibda o'zgaradi, bu stakandagi suvda ko'piklarni suzib chiqishini esga soladi. Ichki davrda qo'shni elementlarni solishtirish va agar kerak bo'lsa, ularni qayta joylashtirish sodir bo'ladi.

Ma'lumotlarni oddiy almashtirish metodi bilan saralash algoritmining sxemasi 6.2-rasmda berilgan.



6.2-rasm. Oddiy almashuv metodi bilan o'sish bo'yicha saralash algoritmining sxemasi

Almashtirib saralash algoritmi solishtirishlarning eng ko'p ehtimol bo'lgan miqdorini amalga oshiradi, ro'yxatni ko'p marta ko'rib chiqish va ko'pgina joylarni almashtirishni bajarishga to'g'ri keladi. Bu uni samarasiz algoritm deb hisoblashga

sabab bo‘ladi. Algoritm imkoniyatini, masalan «oldinga» va «orqaga» o‘tishlarni navbatma navbat amalga oshirish bilan ko‘paytirish mumkin.

6.4. Qo‘shni elementlar juftliklarini izchil tartibga solinishi

Metodning g‘oyasi qo‘shni elementlarni izchillik bilan solishtirish va agar talab qilinsa, ularning joylarini almashtirishdan iboratdir.

Masala. $A(i); i = 1, \dots, N$ massivi berilgan. Uni qo‘shni elementlar juftliklarini kamayish bo‘yicha tartibga solish metodi bo‘yicha saralash.

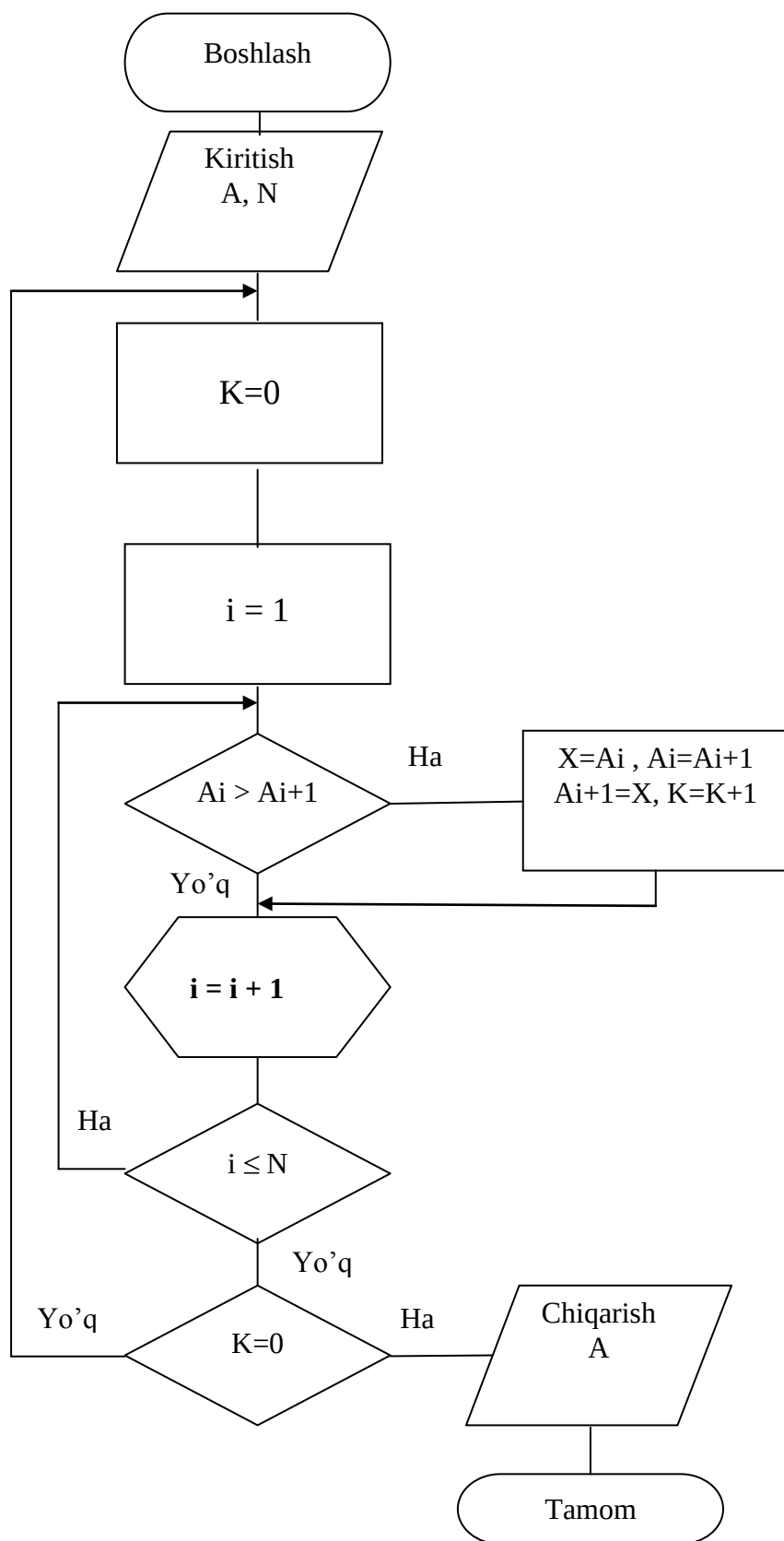
Yechim. K hisoblagichi orqali bajarilgan joy almashtirishlarning miqdorini belgilaymiz. Algoritm kiritilgan tuzilmaning ikkita sikliga ega. Tashqi siklda o‘zgaruvchan K ning nolsiz miqdorini tekshirish amalga oshiriladi.

Ichki siklda i o‘zgaruvchisi bo‘yicha qo‘shni elementlar $A(i)$ va $A(i + 1)$ ning izchil solishtirilishi tashkil qilinadi. Va agar $A(i) > A(i + 1)$ bo‘lsa, unda ularning joylarini almashtiramiz va hisoblagichga 1 ni qo‘shamiz.

Ichki sikl tugagandan keyin joy almashtirishlarning soni izchillikda tekshiriladi. Va agar $K = 0$ bo‘lsa, unda joy almashtirishlar bo‘lmagan va massiv tartibga solingan. Agar $K \neq 0$ bo‘lsa, unda ichki sikl yana qaytariladi.

Ma’lumotlarni qo‘shni elementlar juftliklarini tartibga solish metodi bilan saralash algoritmining sxemasi 6.3-rasmda berilgan.

Bajarilgan joylarni almashtirish miqdorini hisoblagichidan foydalanish ushbu algoritmnı oldingiga qaraganda samaraliroq qiladi.



6.3-rasm. Qo'shni elementlar juftliklarini tartibga solish metodi bilan saralash algoritmining sxemasi

Tayanch iboralar

saralash, ma`lumotlarni saralash, element, oddiy tanlov metodi, massiv, tashqi sikl, sikl parametri, ichki sikl, eng kichik element, oddiy almashuv saralash metodi, qo'shni elementlar juftliklarini izchil tartibga solinishi.

Nazorat va muhokama uchun savollar

1. Ko'pik metodi bilan saralash g'oyasini bayon qiling.
2. Qaysi metodda yonma-yon turuvchi ikkita element joylarini almashtiradilar?
3. Tanlash metodi ko'pik metodidan nimasi bilan farqlanadi?
4. Ma`lumotlar izchilligini saralashning samaradorligi qanday omillarga bog'liq?
5. Saralash algoritmlarining samaradorligini baholash uchun qanday metodikadan foydalaniladi?
6. Qanday metodda joylarni almashtirish miqdorlarini hisoblagichi kiritiladi va bu hisoblagichni kiritishdan ma`no nimada?
7. Talabalar guruhining testdan o'tkazish natijalari berilgan. Talabalar bilimlarini baholash reytingini tuzing.

7-bob. ALGORITMIK DASTURLASH TILLARINING UMUMIY TA'RIFI

7.1. Dasturlash tillari va ularning tasniflanisi.

7.2. Paskal tilining umumiy ta`rifi.

7.3. Paskal algoritmik tilining alifbosi.

7.4. Paskal tilida ma'lumotlar turlari.

7.5. Paskal algoritmik tilida ifodalar.

7.6. Paskal tilida yozilgan dasturning tuzilishi.

7.1. Dasturlash tillari va ularni tasniflashi

Kompyuterda dasturlash - bu kompyuter mikroprotsessori uchun turli buyruqlar berish, qachon, qayerda nimani o'zgartirish va nimalarni kiritish yoki chiqarish haqida buyruqlar berishdir. Hozirgi kunda dasturlash tillarini u yoki bu belgisi bo'yicha tasniflash mumkin. Dasturlash tilining kompyuterga bog'liqlik darajasi bo'yicha tasniflash eng umumiy hisoblanadi .

Yuqorida aytilgan belgiga qarab, dasturlash tillari kompyutera bog'liq va kompyuterga bog'liq bo'lmagan tillarga bo'linadi.

Kompyuterga bog'lik tillar, o'z navbatida, kompyuter tillari va kompyuterga mo'ljallangan tillarga ajratiladi.

Dasturlash tilining kompyuter tiliga yaqinligi darajasini tariflash uchun til darajasi tushunchasi qo'llaniladi.

Kompyuter tili 0 daraja deb qabul qilingan bo'lib, sanoq boshi hisoblanadi. Odamning tabiiy tili "eng yuqori darajadagi til" deb qaraladi.

Kompyuterga bog'liq bo'lmagan tillar ham ikkita turga bo'linadi: birinchisi protseduraga mo'ljallangan tillar, ikkinchisiga - muammoga mo'ljallangan tillar.

Protseduraga mo'ljallangan tillar turli masalalarni yechish algoritmlarini (protseduralarni) tavsiflashga mo'ljallangan; shuning uchun ular ko'pincha oddiy qilib "algoritmik tillar" deb ataladi.

Ushbu tillar yechilayotgan masalalar xususiyatlarini to'la hisobga oladi va kompyuterning turiga deyarli bog'liq emas. Bu xildagi tillar tarkibi kompyuter tiliga qaraganda tabiiy tilga, masalan, ingliz tiliga yaqinroq.

Hozirgi kunda hisoblash, muhandis-texnik, iqtisodiy, matnli va sonli axborotlarni taxlil qilish va boshqa masalalarni yechish tillari malum.

Masalan: FORTRAN tili 1954 yili ishlab chiqilgan bo'lib, FORMula TRANslator - formulalar translyatori degan manoni anglatadi va ilmiy va muhandis - texnik masalalarni hisoblashlarda qo'llaniladi.

ALGOL tili 1960 yili yaratilgan bo'lib, ALGORITMIC Language -algoritmik til degan ma'noni anglatadi va ilmiy-texnik masalalarni hisoblashlarda qo'llaniladi.

KOBOL tili 1959 yili yaratilgan bo'lib, Common Business Oriented Language - "savdo-sotiq masalalariga mo'ljallangan til" degan ma'noni anglatadi. Korxona va tarmoqning moddiy boyligini, moliyasini, ishlab chiqargan mahsulotini hisobga olish bilan bog'liq iqtisodiy masalalarni echish uchun qo'llaniladi.

PASKAL tili 1971 yilda e'lon qilingan bo'lib, fransuz olimi Blez Paskal nomiga qo'yilgan. Turli xildagi masalalar echimini olishda tartiblangan (strukturaviy) dasturlar tuzishda qo'llaniladi.

PL/1 tili 1964 yilda yaratilgan bo'lib, Programming Language/ 1 - 1-tartib raqamli dasturlash tili ma'nosini anglatadi. Ushbu til universal tillar turkumiga kiradi.

Bu tilda ishlab chiqilgan dasturlar kompyuterni yangisi bilan almashtirilganda qaytadan tuzib chiqilishi zarur emas.

BEYSIK (BASIC - Beginner's All Purpose Symbolic Instruction Code - boshlovchilar uchun ko'p maqsadli dasturlash tili) hisoblash algoritmlarini yozish uchun qo'llaniladigan algoritmik tildir. Bu til 1965 yilda Dartmut kolleji xodimlari Kemini va Kurtslar tomonidan ishlab chiqilgan.

Protseduraga mo'ljallangan tillardan masalalarning matematik ifodalari, algoritmlar va dasturlash usullari bilan tanish bo'lgan mutaxassislar foydalaniladilar.

Bunda ulardan kompyuterning tuzilishini mukammal bilish talab qilinmaydi.

Muammoga mo'ljallangan tillar kompyuterda masala yechish usullari va dasturlash usullari bilan tanish bo'lmagan foydalanuvchilar uchun yaratilgandir.

7.2. Paskal tilining umumiy ta`rifi

Pascal dasturlash tili yuqori darajali obyektga yo'naltirilgan dasturlash tili. Pascal dasturlash tili 1970-yilning boshida Niklaus Wirth tomonidan yaratilgan. Pascal dasturlash tili, dasturlashni tizimli, strukturaviy va protseduraviy o'rgatish uchun ishlab chiqilgan. Lekin Pascal dasturlash tilidan foydalanib ham yuqori darajali mukammal dasturlarni tuzish mumkin.

Pascal dasturlash tilining asosi Algol tilidir. Pascal dasturlash tili juda ko'plab ma'lumotlar toifasiga ega til va dasturlarni strukturaviy dasturlashga juda qulay. Hozirgi Pascal kodlarini o'qisangiz xuddi ingliz tilida matn o'qiyotgandek bo'lasiz.

Pascal dasturlash tili ko'plab mamlakatlarda hozirda ham qo'llanilayotganining sabablari :

- * O'rganish juda oson;
- * Strukturalashgan dasturlash tili;
- * Mustahkam, ishonchli dasturlar tuzish imkoni mavjudligi;
- * Juda ko'plab OTlarda kompilyatsiya qilishni imkoni borligi;
- * va boshqalar...

Pascal dasturlash tilining eng zo'r imkoniyatlari :

- * Pascal kuchli tillardan biri;
- * Xatoliklarni juda aniqlikda tekshirilishi;
- * Juda ko'plab ma'lumotlar turlari borligi (massivlar, yozuvlar, fayllar va setlar);
- * Turli xil yo'nalishlarda dasturlash imkoniyati borligi;
- * Strukturaviy, funksiyaviy va protseduraviy dasturlash juda qulayligi;
- * OYD (OOP) ni qo'llay olganligi.

Pascal dasturlash tili haqida faktlar :

- * Pascal dasturlash tili nomi Fransuz olimi Bleise Pascal sharafiga qo'yilgan;
- * Pascal dasturlash tili Algol dasturlash tilidan kelib chiqqan;
- * Pascal dasturlash tilining standart versiyasi, 1983-yilda chiqdi;

* Apple korporatsiyasi Lisa va Mac operatsion sistemalarini ishlab chiqishda Pascal dasturlash tilidan foydalanilgan. Xozirda esa o'z dasturlash tillariga o'girilgan.²²

7.3. Paskal algoritmik tilining alifbosi

Paskal algoritmik tilining alifbosi quyidagilardan iborat:

- 1) 26 ta lotin harflari;
- 2) 0 dan 9 gacha arab raqamlari;
- 3) 32 ta kirill harflari;
- 4) maxsus belgilar - (+, -, *, /, :, ;, [., [., =, >, <)

Paskal tilida so'z deb bir nechta belgilar ketma-ketligi tushuniladi.

Xizmatchi so'z deb Paskal tilidagi standart nom tushuniladi. Bu nom maxsus ma'noni anglatadi va uni ma'lumotlarga berib bo'lmaydi.

Masalan, PROGRAM, BEGIN, END va h.k.

Paskal tilidagi ma'lumotlarning elementlari bo'lib o'zgaruvchilar, o'zgarmaslar, izohlar xizmat qiladi.

O'zgaruvchilar deb hisoblash jarayonida o'z qiymatini o'zgartiradigan kattaliklarga aytiladi. O'zgaruvchilarning nomlari (identifikatorlar) harflardan yoki harf va raqamlardan iborat buladi. Belgilar soni 8 tadan oshmasligi kerak.

O'zgarmaslar (const) deb hisoblash jarayonida o'z qiymatini o'zgartirmaydigan kattaliklarga aytiladi. Bularga ham o'zgaruvchilar kabi nom beriladi.

Izohlar - dasturning ma'lum qismini tavsiflash uchun ishlatiladi va bu qatorda hech qanday amal bajarilmaydi, ya'ni dasturning biror blokini yaxshiroq tushunishga xizmat qiladi.

Izoh (*,*) yoki {} simvollar orasida beriladi. Katta qavslar {*,*} va [*,*] lar ham ishlatilishi mumkin.

²² <https://texnoman.uz/post/pascal-dasturlash-tili-haqida.html> Axborot texnologiyalari sohasining o'zbek tilidagi bloggerlar portali

7.4. Paskal tilida ma'lumotlar turlari

Paskal tilida ma'lumotlarning toifalari ikki xil bo'ladi: oddiy-skalyar va murakkab. Skalyar toifa, o'z navbatida, o'zgaruvchi va standart toifalarga bo'linadi.

O'zgaruvchi toifaga qayd qilingan va chegaralangan toifalar kirsar, standart toifaga butun - INTEGER, haqiqiy - REAL, mantiqiy - BOOLEAN, simvol - CHAR toifalar kiradi. Bundan tashqari, Turbo Paskalda qator - STRING turi kiritilgan.

Murakkab toifalarga esa ma'lumotlarning muntazam (massiv), to'plam, aralash (yozuvlar), murojaat va faylli toifalari kiradi.

Ma'lumotlarning standart turlari. Standart turdagi ma'lumotlar dasturda maxsus xizmatchi so'zlar yordamida tavsiflanadi:

INTEGER - BUTUN toifadagi ma'lumotlar faqat butun sonlarni qabul qiladi. Ular Paskal tilida quyidagicha ifodalanadi: -7, 10,89, -35, 0 va h.k.

REAL - haqiqiy toifadagi ma'lumotlar haqiqiy sonlarni qabul qiladi va ikki xil ko'rinishda bo'ladi:

a) qo'zg'almas nuqtali haqiqiy sonlar (3.56, 0.88,-150.45);

b) qo'zg'aluvchan nuqtali haqiqiy sonlar ($0.546=546E-03$, $96.78*100=96.78E02$ va h.k.

CHAR - BELGI (simvol) toifadagi ma'lumotlar qiymat sifatida kodlar jadvalidagi belgilarning bittasini qabul qiladi. Bu ma'lumotlar apostrof ('belgisi) ichiga olib yoziladi. Masalan 'F', 'A', 'R', '5', '8' va h.k.)

BOOLEAN - MANTIQUIY toifadagi ma'lumotlar. Bu nom ingliz matematigi Jorj Bul sharafiga shunday deb ataladi va shu turdagi o'zgaruvchilar faqat ikkita qiymat qabul qiladi, TRUE -ROST yoki FALSE –YOLG'ON.

Bu ma'lumotlar ustida mantiqiy ko'paytirish AND, mantiqiy qo'shish OR va mantiqiy inkor qilish NOT amallarini bajarish mumkin.

7.1-jadvalda BOOLEAN turidagi A va B o'zgaruvchilarda mantiqiy amallarning bajarilishi natijasi keltirilgan.

7.1-jadval

A	B	NOT A	NOT B	A AND B	A OR B
TRUE	TRUE	FALSE	FALSE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
FALSE	TRUE	TRUE	FALSE	FALSE	TRUE
FALSE	FALSE	TRUE	TRUE	FALSE	FALSE

STRING [N] - qator. Bu erda N qatordagi simvollar soni (ko'rsatilmagan bo'lsa, u 256 ga tenglashtiriladi). Bu turdagi ma'lumotlar bitga yoki bir necha belgilar ketma-ketligidan tashkil topib, apostrof ichida beriladi.

Masalan: 'Informatika', 'STUDENT'.

7.5. Paskal algoritmik tilida ifodalar

Paskal algoritmik tilida arifmetik, mantiqiy va belgili ifodalar aniqlangan.

Arifmetik ifodalar o'zgaruvchilar, o'zgarmaslar, arifmetik amal belgilari qavs va standart funksiyalardan tashkil topadi.

Arifmetik ifodada qatnashayotgan amallar ustivorligiga qarab bajariladi. Amallar quyidagi ustivorlikka ega:

1. Qavs ichidagi amallar va standart funksiyalarni hisoblash.
2. Ko'paytirish, bo'lish, DIV, MOD amallari.
3. Qo'shish va ayirish amallari.

Misollar: $\ln x^2 + \cos(x) - \sin^2(x) + e^{3x}$ ifoda berilgan.

Paskal tilida bu ifoda quyidagi ko'rinishda yoziladi:

$\ln(\text{sqr}(x)) + \cos(x) - \text{sqr}(\sin(x)) + \exp(3*x)$

Turbo Paskal algoritmik tilida 10^n ko'rinishidagi sonlarni $E+n$, 10^{-n} ko'rinishidagi sonlarni esa $E-n$ kabi tasvirlash mumkin. 10 ning musbat darajasini ifodalashda «+» ishorasini tashlab ketish xam mumkin.

Masalan: $10^6 = E6$; $3000 = 3E3$.

7.6. Paskal tilida yozilgan dasturning tuzilishi

Paskalda yozilgan dastur quyidagi qismlardan iborat:

<dastur sarlavhasi>

< belgilar bo'limi >

< konstantalar bo'limi >

< turlar bo'limi >

< o'zgaruvchilar bo'limi >

< protsedura va funksiyalar bo'limi >

< operatorlar bo'limi >

Dastur sarlavhasi PROGRAM xizmatchi so'z bilan boshlanadi, bu so'zdan keyin shu dasturga berilgan nom yoziladi (bu nom dastur ichida hech qanday maqsadda ishlatilishi mumkin emas). Dastur nomidan keyin yumaloq qavs ichida dastur parametrlari ro'yxati beriladi. Bu parametrlar fayllar nomi bo'lib, ular orqali dastur tashqi muhit bilan bog'lanadi.

Mavjud bo'lgan ko'p fayllar orasidan 2 ta standart matnli fayllar ajratilgan. Bular dasturni boshlang'ich ma'lumotlar va ma'lumotlarni qayta ishlash natijalari bilan bog'laydigan INPUT va OUTPUTlardir.

Dastur sarlavhasi yozilishini quyidagi misolda ko'ramiz:

PROGRAM PASCAL (INPUT, OUTPUT);

Paskalning ko'p versiyalarida INPUT va OUTPUT yozilmasligi ham mumkin.

Dasturning asosiy vazifasi - bu kompyuterga ma'lumotlarni (berilganlarni) qayta ishlash bo'yicha buyruqlar berishdir. Bu buyruqlar operatorlar yordamida beriladi. Shuning uchun ham operatorlar bo'limi asosiy hisoblanadi va har qanday dasturda ishtirok etishi shart.

Operatorlar bo'limidan oldingi bo'limlar ba'zi dasturlarda ishtirok etmasliklari ham mumkin. Ular tavsiflash bo'limlari deyiladi. Tavsiflashning bu bo'limlarida operatorlar bo'limida ishlatilgan kattaliklar tavsiflanadi, ya'ni e'lon qilinadi.

Belgilar bo'limi. Dasturning ixtiyoriy operatori oldiga belgi va ikki nuqta qo'yib qolgan operatorlardan ajratish mumkin. Operatorning belgisi bittadan ortiq bo'lishi mumkin emas. Operatorlarning belgilari bir xil bo'lishi mumkin emas.

Operatorning belgisiga uning nomi sifatida xaraladi. Dasturda ishlatilayotgan har bir belgi avval belgilar bo'limida tavsiflash yo'li bilan e'lon qilinishi kerak:

Belgilar bo'limi label (belgi) xizmatchi so'z bilan ochiladi, undan keyin belgilar, ya'ni ketma-ket vergul bilan ajratilgan konstanta shaklida yoziladi. Belgilar bo'limida e'lon qilinayotgan belgilar ixtiyoriy ketma-ketlikda yozilishi mumkin.

Belgilar (0-9999) orasidagi sonlardan olinadi (Turbo Paskalda belgi sifatida alfavitning boshqa elementlarini ham ishlatish mumkin).

Misol: label 24, 25, 26;

Konstantalar-o'zgarmlar bo'limi. Konstanta deganda aniq bir u yoki bu turdagi qiymatni tushunamiz, dastur ishlash jarayonida bu qiymat o'zgarmaydi shu sababli o'zgarmlar deb ham yuritiladi.

Paskal tilida konstantalarga nom beriladi. Konstantalar bo'limi const xizmatchi so'z bilan boshlanadi va nuqta vergul (;) bilan tugaydi.

Misol: const C=34; D=6.3; kaf = 'fizika';

Turlar bo'limi. Paskal tilida turtta standart turdagi qiymatlar bor, bular: integer (butun), real (haqiqiy), char (matnli), boolean (mantiqiy).

Bu turdagi qiymatlar bilan bir qatorda, Paskal tilida boshqa turdagi qiymatlarni ham ishlatish mumkin. Lekin standart turlardan farqli ravishda boshqa turlar dasturda aniq tavsiflanishi kerak.

Turlar bo'limi type (tur) xizmatchi so'zi bilan boshlanadi, keyin turlar tavsifi yoziladi. Turlar tavsifi bir-biridan nuqta vergul (;) orqali ajratiladi.

Misol:

Type

Mantiq=boolean;

Hafta=(dush,sesh,chor,payjum,shan,yaksh);

ish kuni= sesh..jum;

Bu misolda uch tur tavsiflari bor. Birinchi qatorda standart mantiqiy turga boshqa nom berilyapti - mantiq. Ikkinchisida yangi qayd qilingan tur kiritilyapti va unga hafta degan nom berilyapti. Uchinchisida yangi chegaralangan tur kiritilyapti, unga ish kuni deb nom berilyapti.

O'zgaruvchilar bo'limi. Alohida o'zgaruvchining tavsifi quyidagicha bo'ladi:

<o'zgaruvchi nomi>: <tur>

Misol: a: real; I:integer;

Birinchi qatorda dastur jarayonida ishlatishga real turdagi a uzgaruvchi kiritilyapti, ikkinchisida I nomli butun qiymatli o'zgaruvchi kiritilyapti.

O'zgaruvchilar bo'limi var (variable - o'zgaruvchi) xizmatchi so'z bilan boshlanadi.

Misol: Var i, j, k: integer;

Protseduralar va funksiyalar bo'limi. Boshqa bo'limlardan farqli ravishda, bu bo'lim maxsus xizmatchi so'z bilan belgilanmaydi. Bu bo'limning boshi Procedure yoki Function so'zi bilan boshlanadi.

Protsedura tavsifi har doim funksiya tavsifidan oldin turadi. Bu bo'lim dasturda standart protsedura va funksiyalardan tashqari, protsedura va funksiyalar ishlatilayotgan bo'lsa va agar ularga dasturdan murojaat qilinayotgan bo'lsa ishlatiladi.

Operatorlar bo'limi. Bu bo'lim dasturning asosiy bo'limi hisoblanadi. Dasturni bajarish bo'limi operatorlik xizmatchi so'zlari begin va end ichiga joylashgan operatorlar ketma-ketligini bajarishga keltiriladi. Operatorlar bir-biridan nuqtali vergul (;) yordamida ajratiladi.

Shuni aytish kerakki, protsedura va funksiya bo'limidagi algoritmlar faqat operatorlar bo'limidan murojaat qilinganda bajariladi. Operatorlar yordamida masalani echish algoritmini bajarishga kerak bo'lgan amallar tavsiflanadi.

Dasturning ohiri nuqta bilan belgilanadi.

Tayanch iboralar

dasturlash, dasturlash tili, Pascal dasturlash tili, o'zgaruvchilar, o'zgarmaslar, izohlar, arifmetik, mantiqiy, belgili ifodalar, dastur sarlavhasi, belgilar bo'limi, konstantalar-o'zgarmaslar bo'limi, turlar bo'limi, o'zgaruvchilar bo'limi, protseduralar va funksiyalar bo'limi, operatorlar bo'limi.

Nazorat va muhokama uchun savollar

1. Dasturlash tillariga ta'rif bering va ularning tasnifini keltiring.
2. Paskal dasturlash tili qanday imkoniyatlarga ega?
3. Paskal tilida ma'lumotlarning qanday turlarini bilasiz?
4. Paskal tilidagi dastur qanday tarkibiy qismlardan iborat?
5. Paskal tilida o'zgaruvchilarni tavsiflash bo'limi qanday ifodalanadi?
6. Dasturda konstantalarni qanday tavsiflash va ishlatish mumkin?
7. Dasturning operatorlar bo'limida qanday jarayonlar bajariladi?

8-bob. DASTURLASH TILINING SODDA OPERATORLARI

8.1. Paskal tilidagi operatorlar.

8.2. O'zlashtirish operatori.

8.3. Ma'lumotlarni kiritish va chiqarish operatorlari.

8.1. Paskal tilidagi operatorlar

Funksional vazifalari bo'yicha Paskal tilidagi operatorlar quyidagi guruhlariga bo'linadi: o'zlashtirish, kiritish-chiqarish, boshqarish.

O'zlashtirish operatorlari - dasturda o'zgaruvchilarga ma'lum qiymatlarni o'zlashtirish uchun ishlatiladi.

Kiritish-chiqarish operatorlari - tezkor xotiraga boshlang'ich ma'lumotlarni kiritadilar va yozuvga chiqaruvchi qurilmalarga hisoblashlar natijasini chiqaradilar.

Boshqarish operatorlari - dastur operatorlarini ketma-ket bajarilishini tashkil etadi.

Paskal tilida yozilgan dastur to'g'ridan-to'g'ri kompyuterga emas, balki biror bajaruvchiga mo'ljallanganki, unga qanday toifadagi qiymatlar bilan ishlay olishi va qo'shimcha ko'rsatmalarsiz qanday amallarni bajara olishi belgilab berilgan bo'lishi kerak.

U yoki bu masalani yechish uchun kattaliklar ustida bajarilishi kerak bo'lgan amallarni ko'rsatish uchun algoritmik tilda operator tushunchasi xizmat qiladi. Dasturdagi har bir operator ma'lumotlarga ishlov berishning mustaqil, mantiqan tugallangan bosqichini ifodalaydi. Paskalda operatorlarning 8 ta turi ko'zda tutilgan. Har qanday masalani yechish aniq qoidalar bo'yicha, berilgan kattaliklardan boshqa kattaliklarni keltirib chiqarish jarayonidan iboratdir.

Ko'p oddiy algoritmlarni dasturlashtirishda asosan o'zlashtirish, kiritish-chiqarish operatorlaridan foydalaniladi.

8.2. O'zlashtirish operatori

Masalani yechish jarayoni qator bajariluvchi bosqichlarga bo'linib ketadi. Bu bosqichlarning har birida ma'lum qiymatlar bo'yicha yangi qiymatlar hisoblanadi.

Yangi qiymatlarni hisoblash uchun ifoda tushunchasi xizmat qiladi, har bir ifoda bitta qiymatni hisoblash qoidasini belgilaydi. Hisoblangan qiymatni hisoblash jarayonining keyingi bosqichida foydalanish uchun eslab qolish zarur, bunday eslab qolish hisoblangan qiymatni ma'lum o'zgaruvchiga o'zlashtirish yo'li bilan amalga oshiriladi.

Bunday amalni bajarish, asosiy operatorlardan biri hisoblanuvchi qiymat berish, ya'ni o'zlashtirish operatori bilan bajariladi.

O'zlashtirish operatorining bajarilishida «:=» belgisining o'ng tomonidagi ifodaning qiymati hisoblanib, belgining chap tomonidagi o'zgaruvchi tomonidan o'zlashtiriladi.

Shunday qilib, o'zlashtirish operatori mustaqil mantiqiy tugallangan hisoblash jarayonini ifodalaydi: o'zlashtirish operatori bajarilishi natijasida ma'lum o'zgaruvchilar keyingi bosqichlarda ishlatilishi mumkin bo'lgan yangi joriy qiymatni qabul qiladi.

Masalan:

A:=10; T:=2; H:=4.4;

8.3. Ma'lumotlarni kiritish va chiqarish operatorlari

Dasturda ma'lumotlarning qiymatlarini xotiraga kiritishni bir necha usullarda bajarish mumkin.

Sonli o'zgaruvchilarga ularning qiymatini berishda o'zlashtirish operatoridan foydalaniladi. Masalan:

A:=15; B :=-3.1;

Dasturni o'zgaruvchilarning turli qiymatlarida bajarish uchun **READ** - kiritish operatori mo'ljallangan.

Kiritish operatori quyidagicha ko'rinishlarda ishlatilishi mumkin:

1) READ(a1,a2,...,an);

bunda, a1,a2,...,an - o'zgaruvchi qiymatlarini ketma- ket standart INPUT protsedura faylidan oluvchi o'zgaruvchilar. O'zgaruvchilarga qiymatlar turiga mos ravishda klaviaturadan kiritiladi.

Aytaylik, A, B, C o'zgaruvchilarga dastur bajarilishi davomida quyidagi qiymatlarni berish kerak bo'lsin: A=15, B=11, S=2.3.

Operator READ(A,B,C) ko'rinishiga ega bo'lib, sonlar qiymatlarini dastur bajarilishi davomida quyidagicha kiritish mumkin:

5 17 6.2 [Enter].

Agar o'zgaruvchi REAL toifada aniqlangan bo'lsa, uning qiymatini butun son yoki haqiqiy son ko'rinishida kiritiladi. Kompyuterning o'zi butun sonni haqiqiy songa o'tkazib oladi.

Masalan:

VAR A, B:REAL;

READ (A, B) operatorining ishlatilishi natijasida 4 va 5 sonlarini probel (bo'sh joy) orqali kiritish mumkin.

2) READLN - bu operator kiritish jarayonida bo'sh qator qoldiradi;

3) READLN(al,a2,...,an);

- operatorning bajarilishida avval al,a2,...,an ga qiymat kiritilib, so'ng keyingi satrga o'tiladi. Bu operator oldingi ikki operatorga teng kuchlidir.

KOMPYUTER xotirasidagi ma'lumotlarni displey ekraniga chiqarish operatori - **WRITE** dir. Operator quyidagi bir nechta ko'rinishlarda ishlatilishi mumkin:

1) WRITE(al,a2,..., an);

bunda al,a2,...,an oddiy o'zgaruvchilar, o'zgarmaslar yoki ifodalar bo'lishi mumkin va ular standart OUTPUT protsedura fayliga chiqariladi.

Masalan:

WRITE('B ning qiymati =', B) operatori displey ekraniga:

B ning qiymati = va undan so'ng B o'zgaruvchining qiymatini chiqaradi.

WRITE operatorida butun va haqiqiy sonlarni ma'lum formatda chiqarish mumkin. Bu format ikki nuqta orqali o'zgaruvchidan so'ng ko'rsatiladi.

Masalan:

WRITE(Y:4:2); operatori bilan Y ning qiymatini chiqarishda, Y ning hamma qiymatini chiqarish uchun 4 ta xona ajratilishi, ulardan ikkitasi kasr qismi uzunligini

anglatadi (bunda sonning butun va kasr qismini ajratuvchi vergul (nuqta) ham hisobga olinishi zarur).

Butun sonlarni chiqarishda kasr qismi formati ko'rsatilmaydi.

Aytaylik, $N=111$ butun sonli qiymatni chiqarish kerak bo'lsin. Chiqarish operatori buning uchun quyidagi ko'rinishda bo'lishi mumkin:

WRITE ('N=', N:3)

Paskal tilida boshqa chiqarish operatorlari ham ishlatiladi.

2) Parametrlarsiz chiqarish operatori

WRITELN - displey ekranida yangi satrga o'tishni ta'minlaydi.

3) WRITELN($a_1, a_2, \dots, a_n$);

- chiqarish operatori oldin $a_1, a_2, \dots, a_n$ larning qiymatlarini chiqaradi, so'ng yangi qatorga o'tishni ta'minlaydi. Shunday qilib, bu ham quyidagi ikki operatorga ekvivalent:

WRITE($a_1, a_2, \dots, a_n$); WRITELN;

Quyida oddiy masalalarni dasturlash namunalari ko'rsatilgan.

Masala 1. Uchburchakning ikkita kateti berilgan $a= 3$ sm va $b= 5$ sm. gipotenuza uzunligini topish dasturini tuzing.

Program gipot;

Const $a=3$; $b= 5$;

Var c : real;

Begin

$c:=\text{sqrt}(\text{sqr}(a)+\text{sqr}(b));$

Write('c=', c)

End.

Masala 2. Arrenius qonuniyati asosida reaksiya tezligi konstantasini temperaturaga bog'liqligini ifodalovchi formula uchun dastur tuzish.

Program arren;

Var K, k_0, E, R, T :real;

Begin

Write('o'zgaruvchilarni kiriting');

Read(k0,E,R,T);

K:= k0* exp(-E/R*T);

Write ('K=',K)

End.

Tayanch iboralar

Pascal dasturlash tili, o'zlashtirish operatorlari, kiritish operatori, chiqarish operatori, boshqarish operatori, o'zgaruvchi.

Nazorat va muhokama uchun savollar

1. Paskal tilida qanday operatorlardan foydalanadi?
2. Paskal tilida o'zlashtirish operatoridan qanday foydalanadi?
3. Paskal tilida qanday tartibda ma'lumotlarni kiritish operatoridan foydalanadi?
4. Paskal tilida qanday tartibda ma'lumotlarni chiqarish operatoridan foydalanadi?
5. Nima uchun read o'qish protsedurasining parametrlari sifatida sonlar va ifodalar (masalan, $k + 1$) emas, faqat o'zgaruvchilar bo'lishi kerak ?

9-bob. TARMOQLANGAN HISOBLASH JARAYONLARNI DASTURLASH

9.1. Tarmoqlanuvchi hisoblash jarayonlarini dasturlash.

9.2. Shartsiz o'tish operatori.

9.3. Shartli o'tish operatori.

9.4. Tanlash operatori.

9.1. Tarmoqlanuvchi hisoblash jarayonlarini dasturlash

Ko'pgina masalalarni yechishda ba'zi bir jarayonlar ma'lum shart yoki shartlarning qo'yilishiga nisbatan bajariladi. Bunday jarayonlar tarmoqlanuvchi jarayonlar deb yuritiladi.

Tarmoqlanuvchi hisoblash jarayonlari oddiy va murakkab bo'lishi mumkin. Bu esa jarayondagi tarmoqlar soniga bog'liq. Ma'lum bir tarmoqlanuvchi jarayon tarkibida yana tarmoqlanishlar bo'lishi mumkin. Bunday tarmoqlanishlari bor bo'lgan hisoblash jarayonlari murakkab tarmoqlanuvchi hisoblash jarayonlari deb ataladi.

Paskal tilida tarmoqlanuvchi jarayonlarni dasturlash uchun shartsiz, shartli o'tish va tanlash operatorlaridan foydalaniladi.

9.2. Shartsiz o'tish operatori

Dasturda ba'zi bir hollarda boshqaruvni to'g'ridan-to'g'ri biron-bir operatorga uzatishga, ya'ni dasturning bajarilish ketma-ketligini buzishga to'g'ri keladi. Bu jarayon shartsiz o'tish operatori yordamida bajariladi. Shartsiz o'tish operatorining umumiy ko'rinishi quyidagicha:

GOTO < operator belgisi>;

Belgi tavsiflash bo'limining LABEL bo'limida albatta tavsiflangan bo'lishi shart.

Misol Program PP;
 LABEL L;
 Var A,V : real;

```

begin
readln (A,V);
IF A> B THEN GOTO L
    ELSE A := A + 1 ;
L: A := A + V ;
writeln (A)
end.

```

9.3. Shartli o'tish operatori

Dasturda boshqaruvni ma'lum shart asosida u yoki bu tarmoqqa uzatish shartli o'tish operatori yordamida amalga oshiriladi. Shartli o'tish operatori ikki xil ko'rinishda ishlatilishi mumkin: to'liq va qisqa.

Shartli o'tish operatorining to'liq ko'rinishi:

IF <mantiqiy ifoda> THEN S1 ELSE S2;

Bu erda IF (agar), THEN (u holda) va ELSE (aks holda) degan xizmatchi so'zlar, S1 va S2 ixtiyoriy operatorlar.

Operatoridagi mantiqiy ifoda boshqaruvni uzatish shartini belgilaydi.

Operatorning ishlash tartibi quyidagicha:

Agar keltirilgan mantiqiy ifoda TRUE (rost) qiymatni qabul qilsa, ya'ni qo'yilgan shart bajarilsa, THEN - xizmatchi so'zidan keyingi operator bajariladi, aks holda ELSE xizmatchi so'zidan keyingi operator bajariladi.

Mantiqiy ifodalarda munosabat amallari, mantiqiy amallar ishlatilishi mumkin.

Masalan,

$A > 5$, $A = B$, $X < 1.5$ va h.k.

Shartlar oddiy va murakkab bo'lishi mumkin. Agar mantiqiy ifodada bitta munosabat amali berilgan bo'lsa, «oddiy shart» ni ifodalaydi.

Kattaliklar orasidagi shartlar HAM, YOKI, EMAS (Paskal tilida AND, OR, NOT) mantiq amallari belgilari orqali bog'lanuvchi bir necha munosabatlardan iborat bo'lsa „murakkab shartlar" deb ataladi.

Masalan,

Matematik yozilishi Algoritmik tilda yozilishi

1) $2 \leq X < 5$ $(X \geq 2) \text{ AND } (X < 5)$

AND amalining natijasi uning ikkala argumenti ham rost bo'lsa rost bo'ladi.

OR amalining natijasi rost bo'lishi uchun argumentlardan birining rost bo'lishi etarli.

NOT amalining natijasi argumentning inkor qiymatiga teng, ya'ni argument rost bo'lsa - natija yolg'on, argument yolg'on bo'lsa - natija rost bo'ladi.

Masalan:

$(4 < 5) \text{ AND } (5 < 100)$ - mantiqiy ifoda TRUE (rost),

$(\sin(X) > 1) \text{ AND } (5 \text{ DIV } 2 = 0)$ ifoda FALSE (yolg'on) qiymatga teng.

Shuni ta'kidlab o'tish kerakki, agar mantiqiy ifodalar, biz yuqorida aytganimizdek, mantiqiy amallar yordamida (AND, OR, NOT) murakkab ko'rinishga ega bo'lsa, ular qavslarga olib yoziladi.

Shartli o'tish operatorining ishlatilishini misollarda ko'rib chiqamiz.

1) IF $x > 0$ THEN $y := \text{sqr}(x)$ ELSE $z := \text{sqr}(x)$;

Operatorning bajarilishi natijasida $x > 0$ bo'lsa, u holda $y := \text{sqr}(x)$ operatori, aks holda $z := \text{sqr}(x)$ operatori bajariladi.

Ayrim algoritmlarda ba'zan shunday xol uchrashi mumkinki, bunda hisoblash jarayonida ayrim amallar ba'zi bir shartlar bajarilgandagina hisoblanadi, aks holda, hech qanday amal bajarilmaydi. Bu holda shartli o'tish operatorini qisqa ko'rinishda ifodalash mumkin.

IF <mantiqiy ifoda> THEN <operator>;

Misol:

IF $X < 1$ THEN $Y := \text{sqr}(X)$;

Shartli o'tishda operator o'rnida, o'z navbatida, yana shartli o'tish operatorining to'la va qisqa ko'rinishlari ishlatilishi mumkin. Masalan:

1) IF B1 THEN IF B2 THEN A;

Bu erda B1 va B2 - mantiqiy ifoda, A - operator.

Bu operatorning bajarilishi natijasida B1 mantiqiy ifoda tekshiriladi, agar TRUE qiymat qabul qilsa, B2 mantiqiy ifoda tekshiriladi, u ham rost bo'lsa (TRUE), A operator bajariladi.

Agar B1 yoki B2 mantiqiy ifodalar yolg'on bo'lsa (FALSE), shartli o'tish operatoridan keyingi operator bajariladi.

Agar shartli o'tish operatorida THEN yoki ELSE dan keyin bir necha operator guruhi bajarilsa, ular tarkibiy operator ko'rinishida yozilishi kerak, ya'ni operatorlar qavsi - BEGIN va END lar orasida yoziladi.

9.4. Tanlash operatori

Juda ko'p tarmoqlanish jarayonlarida tarmoqlanish ikki yoki undan ortiq tarmoqqa ajraladi. Umuman olganda, buni shartli o'tish operatori yordamida amalga oshirish mumkin:

```
IF B1 THEN A1 ELSE
```

```
IF B2 THEN A2 ELSE
```

```
IF BK THEN AK ;
```

Lekin bu hollarda shartli o'tish operatorlarining yozilishi noqulay.

Ko'p hollarda dasturchi uchun shartli operatorning umumiy lashgan ko'rinishi - tanlash (variant) operatorini ishlatish qulay.

Tanlash operatorining metaformulasi quyidagicha yoziladi:

```
< tanlash operatori>::= CASE <operator selektori> OF < tanlash ro'yxati  
elementi>; END
```

bunda:

Tanlash operatorining umumiy ko'rinishi:

```
CASE S OF
```

```
M1 : A1;
```

```
M2: A2;
```

```
Mp: An
```

```
END;
```

Bu erda CASE (tanlash) -xizmatchi soʻz, OF (dan), S - selektor, Mi – operatorlar belgilari, Ai -operatorlar (i=1 dan n gacha).

CASE operatori tarmoqlanish jarayonini berilgan bir necha operatorlardan birini tanlash yoʻli bilan amalga oshiradi. Tanlash operatorida barcha operatorlar, shu jumladan bajarilishi uchun tanlangan operator ham aniq ravishda keltiriladi (berilgan operatorlar ketma-ketligi chegaralangan).

Bajarilishi kerak boʻlgan operator yoki operatorlar ketma-ketligi operator selektorining qiymatiga koʻra aniqlanadi. Operator selektori sifatida haqiqiy boʻlmagan, skalyar koʻrinishdagi har qanday ifoda yoki oʻzgaruvchi ishlatilishi mumkin.

Operatorning ishlashida uning tarkibidagi har bir operator tanlash belgisi deb ataluvchi belgi bilan ta'minlanadi. Bu belgi operatorning bajarilishi uchun zarur boʻlgan selektorning maxsus qiymatini qabul qiladigan selektorning tavsifiga mos konstantadir. Operator bir necha mavjud qiymatlar bilan ishlashi uchun, unda tanlash belgilari roʻyxati keltirilishi kerak.

Tanlash operatoridagi belgili operatorlar oddiy belgiga ham ega boʻlishlari mumkin. Bu holda oldin tanlash belgilari, soʻngra oddiy belgilar yoziladi.

Shuni ham inobatga olish lozimki, tanlash operatoriga faqat CASE xizmatchi soʻz orqali kirish mumkin, ya'ni tanlash operatoridan tashqaridagi oʻtish operatori orqali bu operatorga murojaat qilish mumkin emas.

Tanlash operatorining bajarilishi uning tarkibidagi operatorlar ketma-ketligidagi bitta operatorning bajarilishiga olib keladi. Shuning uchun ularning biridan biriga GOTO operatori yordamida oʻtish xato demakdir.

Shartli oʻtish operatorining quyidagi

IF B THEN A1 ELSE A2

koʻrinishi tanlash operatorining quyidagi koʻrinishiga ekvivalentdir:

CASE B OF

TRUE: A1;

FALSE:A2;

END;

qisqa ko‘rinishdagi shartli o‘tish operatorining IF B THEN A ko‘rinishi tanlash operatorining quyidagi ko‘rinishga ekvivalentdir:

CASE B OF

TRUE: A;

FALSE

END;

Misol:

CASE T OF

'*', '/': R:=1;

'+', '-': R:=2

End;

Bu operatorning bajarilishi natijasida, agar T-belgili o‘zgaruvchi "+" yoki "-" belgi qiymatlarni qabul qilsa, R o‘zgaruvchi 2 qiymatni, agar T o‘zgaruvchi "*" yoki "/" belgini qabul qilsa, R o‘zgaruvchi 1 qiymatni qabul qiladi.

Masala 1. Berilgan ikkita son. Agar birinchi son absolyut qiymat jihatidan ikkinchi sondan katta bo‘lsa, u holda birinchi sonni besh marta kamaytirish, aks holda sonlarni o‘zgarishsiz qoldirish dasturi tuzilsin.

Program misol;

Var x,y; real;

Begin

Read(x,y);

If abs(x) > abs (y) then x:=x / 5;

Write (x,y)

End.

Masala 2. Agar temperatura T 30^0 dan oshib ketsa, u holda bosim P $10T^2$ ga , 30^0 dan kamayib ketsa, u holda $15 T^3$ ga, 30^0 ga teng bo‘lsa u holda bosim $10 T^3 + \sin^2 T$ ga teng bo‘ladi. Shunday holatni ifodalovchi dastur tuzish.

Program temp;

Var P,T: real;

Begin

Read(T);

If $T > 30$ then $P := 10 * \text{sqr}(T)$;

If $T < 30$ then $P := 15 * \exp(3 * \ln(T))$ else $P := 10 * \exp(3 * \ln(T)) + \text{sqr}(\sin(T))$;

Write ('P= ', P)

End.

Tayanch iboralar

tarmoqlanuvchi hisoblash jarayonlari, shartsiz o'tish operatori, operator belgisi, shartli o'tish operatori, mantiqiy ifoda, operator, tanlash operatori, operator selektori, tanlash ro'yxati elementi.

Nazorat va muhokama uchun savollar

1. Tarmoqlanuvchi hisoblash jarayoniga ta'rif bering.
2. Shartsiz o'tish operatori va uning ishlatilishini aytib bering.
3. Shartli o'tish operatorining to'liq ko'rinishi qanday?
4. Shartli o'tish operatorining qisqa ko'rinishi qanday?
5. Tarmoqlanuvchi hisoblash jarayonida ikkitadan ko'p tarmoqdan iborat bo'lgan holni tushuntiring.
6. Tanlash operatori va uning dasturda ishlatilishi tartibini ayting.

10-bob. SODDA TAKRORLANUVCHI HISOBLASH JARAYONLARNI DASTURLASH

10.1. Takrorlanuvchi jarayonlarini tashkil qilish.

10.2. Sharti avval tekshiriladigan takrorlanish jarayoni.

10.3. Sharti keyin tekshiriladigan takrorlanish jarayoni.

10.4. Parametrli takrorlanish jarayoni.

10.1. Takrorlanuvchi jarayonlarini tashkil qilish

Amaliyotda murakkab jarayonlarni dasturlashda ma'lum buyruqlar ketma-ketligini ma'lum shartlar asosida qayta-qayta bajarish zaruriyati tug'iladi. Ma'lum bir o'zgaruvchining turli qiymatlarida ma'lum buyruqlar tizimining biron-bir qonuniyatga asosan qayta-qayta bajarilishi takrorlanuvchi hisoblash jarayoni - sikl deb ataladi.

Takrorlanuvchi hisoblash jarayonining takror-takror hisoblanadigan qismini takrorlanishning tanasi (jismi) deb ataladi.

Takrorlanish ichida qiymatlari o'zgarib boradigan o'zgaruvchini takrorlanish o'zgaruvchisi yoki takrorlanishni boshqaruvchi o'zgaruvchisi (sikl parametri) deb yuritiladi.

Takrorlanuvchi jarayonning algoritmi umumiy holda quyidagilarni o'z ichiga olishi kerak:

1. Takrorlanishni tayyorlash - takrorlanishni boshlashdan oddin takrorlanishda qatnashadigan o'zgaruvchilarning boshlang'ich qiymatlari yoki takrorlanish o'zgaruvchisining boshlang'ich qiymati o'rnatiladi, takrorlanish o'zgaruvchisining o'zgarish qadami belgilanadi.

2. Takrorlanish tanasi - takrorlanish o'zgaruvchilarining turli qiymatlari uchun takror bajariladigan amallar ketma-ketligi ko'rsatiladi.

3. Takrorlanish o'zgaruvchisiga yangi qiymat berish - har bir takrorlanishdan avval o'zgaruvchiga o'zgarish qadamiga mos ravishda yangi qiymat beriladi.

4. Takrorlanishni boshqarish - takrorlanishni davom ettirish sharti tekshiriladi, takrorlanishning boshiga o'tish ko'rsatiladi.

Paskal algoritmik tilida uch xil ko'rinishda takrorlanuvchi hisoblash jarayonini tashkil qilish mumkin va bu jarayonlarni dasturlash uchun maxsus operatorlar belgilangan:

- sharti avval tekshiriladigan takrorlanish jarayoni, bu maxsus WHILE operatori orqali amalga oshiriladi;

- sharti keyin tekshiriladigan takrorlanish jarayoni, bu jarayonni maxsus REPEAT operatori orqali amalga oshiriladi;

- parametrli takrorlanish jarayoni, bu jarayon maxsus FOR operatori yordamida amalga oshiriladi.

10.2. Sharti avval tekshiriladigan takrorlanish jarayoni

Takrorlanuvchi jarayonning bu ko'rinishi takrorlanish soni oldindan noma'lum bo'lgan hollarda, ya'ni takrorlanishdan chiqish ma'lum shartga bog'liq bo'lgan hollarda ishlatiladi. Takrorlanishning bu jarayonida takrorlanishdan chiqish sharti takrorlanish tanasini bajarishdan oldin tekshiriladi.

Ushbu operatorning umumiy ko'rinishi quyidagicha:

WHILE L DO M ;

bu erda, **WHILE** - toki, do - bajarish ma'nosini anglatuvchi xizmatchi so'zlar, L - mantiqiy ifoda, M - operatorlar yoki operatorlar guruhi, u takrorlanish tanasini belgilaydi. Takrorlanish tanasida bitta yoki bir necha operatorlar guruhi bo'lishi mumkin.

Bunda operatorlar guruhi, albatta, Begin va End orasida yozilishi kerak.

Operatorning bajarilishi quyidagicha: L mantiqiy ifodaning qiymati hisoblanadi. Agar L mantiqiy ifoda rost qiymatga ega bo'lsa, M operatori bajariladi va bu operator L mantiqiy ifodaning qiymati yolg'on bo'lgungacha qayta-qayta bajariladi.

Agar L mantiqiy ifodaning qiymati birinchi tekshirishdayoq yolgʻon boʻlsa, M operatori biron marta ham bajarilmaydi va boshqaruv WHILE operatoridan keyingi operatorga uzatiladi.

Misollar.

1) N! ni hisoblash dasturi tuzilsin.

Program fact;

Var n, i, p: integer;

Begin

Read(n); P:=1; I:=0;

While I<n do

Begin

I:=I+1;P:=P*I;

End;

Write ('n! =', P)

End.

2) Sonni oʻnlik sanoq tizimidan ikkilikka oʻtkazish dasturi tuzilsin

PROGRAM sanoq;

TYPE massiv=array [1..50] of integer;

var a:massiv;

n,i:integer;

begin write('vvedite chislo:');

readln(n);

i:=1;

while n>=2 do

begin a[i]:= n mod 2;

i:=i+1;

n:= n div 2;

end;

i:=i-1;

```

write(n);
while i<>0 do
begin write(' ',a[i]);
i:=i-1;
end;
writeln
end.

```

3) Quyidagi dasturning bajaradigan funksiyasini tushuntiring

```

program pasc;
const a = pi / 4;
b = pi/2;
h = pi/60;
var x,fx :real;
begin x:=a;
writeln(' x y');
while x<=b do
begin fx:=sin(x)+0.5*cos(x);
writeln ('f (',x:4:2,')=',fx:4:2);
x:=x+h;
end;
end.

```

10.3. Sharti keyin tekshiriladigan takrorlanish jarayoni

Takrorlanish jarayonining bu ko‘rinishi ham takrorlanish soni oldindan ma'lum bo‘lmagan hollarda ishlatiladi. Bunday jarayonda biror buyruq yoki buyruqlar tizimi berilgan shart bajarilgunga qadar takror va takror bajariladi. Bu jarayonni avvalgisidan farqi shundaki, bunda takrorlanishning tanasi hech bo‘lmaganda bir marta bo‘lsa ham bajariladi, chunki takrorlanishdan chiqish sharti takrorlanishning tanasi bajarilgandan keyin tekshiriladi.

Bunday takrorlanish jarayonini dasturlashda maxsus REPEAT operatoridan foydalaniladi.

Ushbu operatorning umumiy ko‘rinishi quyidagicha:

REPEAT M UNTIL L;

bu erda

REPEAT - takrorlamoq, UNTIL - «gacha» ma'nosini anglatuvchi xizmatchi soʻzlar;

M - takrorlanish tanasi, uning tarkibida bitta operator yoki operatorlar guruhi boʻlishi mumkin. Operatorlar guruhini Begin va End orasiga olish shart emas.

Operator quyidagi tartibda bajariladi:

- takrorlanish tanasiga kirgan operatorlar birin-ketin bajariladi. Soʻng L mantiqiy ifodaning qiymati topiladi, ya'ni shart tekshiriladi. Agar bu shart bajarilsa (L ning qiymati TRUE (rost) boʻlsa) boshqaruv takrorlashdan tashqariga, UNTIL soʻzidan keyingi operatorga uzatiladi. Aks holda takrorlanish davom etadi.

Takrorlanish jarayonining bunday koʻrinishi yuqorida keltirilgan takrorlanishdan shu bilan farq qiladiki, bunda takrorlanish tanasi hech boʻlmaganda bir marta bajariladi.

Misol.

1) $S=1+1/2+1/3+\dots+1/n$ yigʻindini hisoblash dasturi tuzilsin.

```
program prep(input,output);
```

```
var i,n : integer; s: real;
```

```
begin
```

```
read(n);
```

```
s:=0;I:=1;
```

```
Repeat
```

```
S:=S+1/ I;
```

```
Write (s);
```

```
I:=I+1;
```

```
Until I>n
```

```
End.
```

10.4. Parametrli takrorlanish jarayoni

Yuqorida keltirilgan takrorlanish operatorlarini, odatda, takrorlanish soni noma'lum bo'lgan hollarda ishlatish maqsadga muvofiqdir. Takrorlanish jarayonining takrorlanish soni uni bajarilishidan oldin ma'lum bo'lsa hamda takrorlanish o'zgaruvchisining qiymatlari orasidagi farq (o'zgarish qadami) 1 ga yoki -1 ga teng bo'lsa, parametrli takrorlanish operatoridan foydalanish qulay.

Bu buyruq quyidagi ko'rinishga egadir:

FOR V:=L1 TO (DOWNTO) L2 DO M

Bu erda: FOR operatorning nomi; V- takrorlanish o'zgaruvchisining nomi; L1, L2 - mos ravishda V ning boshlang'ich va oxirgi qiymati; TO - "gacha" ma'nosini anglatuvchi xizmatchi so'z; DOWNTO - V ning qiymati -1 ga o'zgarganda TO ning o'rnida ishlatiladi; DO - bajarilsin degan ma'noni anglatadi; M - takrorlanish tanasi.

Tanada bir necha operator bo'lsa, ular Begin va End xizmatchi so'zlari orasida yoziladi.

Takrorlanish o'zgaruvchisining toifasi REAL va cheklanmagan INTEGER toifasidan tashqari, barcha oddiy toifalarda berilishi mumkin va uning qiymatlari orasidagi farq bitta kattalikka teng. L1 va L2 V ning toifasidagi kattaliklar bo'lib, uning o'rnida o'zgaruvchi nomlarini, ifodalarni va o'zgarmaslarni ishlatish mumkin.

Misol. 1 dan 100 gacha bo'lgan musbat sonlar yig'indisini hisoblash dasturini tuzish.

```
Program sum;  
Var s:real; I: integer;  
Begin  
S:=0;  
For I:=1 to 100 do  
Begin  
S:=S+I;  
End;  
Write('s=',s)
```


End.

Tayanch iboralar

takrorlanuvchi hisoblash jarayoni, sharti avval tekshiriladigan takrorlanish jarayoni, mantiqiy ifoda, takrorlanish tanasi, sharti keyin tekshiriladigan takrorlanish jarayoni, parametrli takrorlanish jarayoni, takrorlanish o'zgaruvchisi

Nazorat va muhokama uchun savollar

1. Takrorlanuvchi hisoblash jarayoni deb qanday jarayonga aytiladi?
2. Takrorlanuvchi hisoblash jarayonlarining algoritmi qanday qismlarni o'z ichiga oladi?
3. Takrorlanuvchi jarayon tanasi va takrorlanish o'zgaruvchisi haqida tushuncha bering.
4. Takrorlanuvchi hisoblash jarayonlari algoritmlarining turlarini aytib bering.
5. Avval sharti tekshiriladigan takrorlanish jarayonini tashkil qilish tartibini tushuntirib bering.
6. Sharti keyin tekshiriladigan takrorlanish jarayonini tashkil qilish tartibini tushuntirib bering.
7. Parametrli takrorlanish jarayoni qanday xususiyatlarga ega?

11-bob. MURAKKAB TAKRORLANUVCHI HISOBLASH JARAYONLARNI DASTURLASH

11.1. Murakkab takrorlanuvchi hisoblash jarayonining tushunchasi.

11.2. Ma'lumotlar massivi haqida tushuncha.

11.3. Massiv elementlarini saralash bo'yicha dasturlarni tuzish.

11.4. Massivda eng kichik elementni izlash algoritmining dasturi.

11.1. Murakkab takrorlanuvchi hisoblash jarayonining tushunchasi

Agar takrorlanish jarayonlarining tanasi takrorlanish strukturasidan tashkil topgan bo'lsa, u holda bunday takrorlanish ichma-ich joylashgan yoki murakkab deb ataladi, ya'ni boshqacha qilib aytganda, bitta takrorlanish ichiga bir yoki bir necha boshqa takrorlanishlar kirsa murakkab tarkibli takrorlanishli dasturlar hosil bo'ladi.

Boshqa takrorlanish jarayonlarini o'z ichiga olgan takrorlanish tashqi sikl deb ataladi.

Takrorlanish jarayonlarini ichida joylashgan takrorlanish ichki sikl deb ataladi.

Murakkab sikllarda quyidagi talablar bajarilishi zarur.

- ichki sikl tashqi sikl ichida to'liq yotishi kerak;
- tsikllar bir-biri bilan kesishmasligi kerak;
- tsikl ichiga tashqaridan to'g'ridan-to'g'ri kirish mumkin emas;
- tsikl parametrlari har xil identifikatorlar bilan belgilanishi kerak.

Masala: $S = \sum_{i=1}^n \prod_{j=1}^n (i + j)^2$ ko'rinishidagi yig'indi va ko'paytmalar ishtirok etgan formula bo'yicha S ning qiymatini hisoblash algoritmini tuzish talab etilsin.

Buni biz yuqorida tanishgan yig'indini hisoblash algoritmi yordamida hisoblashimiz mumkin. Buning uchun $P = \prod_{j=1}^n (i + j)^2$ deb belgilasak, u holda

$S = \sum_{i=1}^n P$ yozuvga ega bo'lamiz. hisoblash algoritmi quyidagi tartibda amalga oshiriladi. S ga 0 boshlanhich qiymat beriladi, i - parametri bo'yicha sikl ochiladi. P

ga 1 qiymat beriladi, undan keyin j-parametri bo'yicha sikl ochiladi. Ichki sikl hisoblanadi, ichki sikl tugagandan keyin tashqi sikl (i-bo'yicha) hisoblash ya'ni $S=S+P$ amalga oshiriladi. Sikl $i < n$ bajarilguncha davom ettiriladi. Ayrim hollarda rekursiyadan (olinadigan natijadan foydalanib keyingisini hosil qilish) foydalanib ichma-ich joylashgan siklik jarayondan qutilish mumkin.

Masalan: $S = \sum_{i=1}^n \frac{i}{(2i+1)!}$ yig'indini hosil qilish algoritmini tuzish uchun $(2i+1)! = \prod_{j=1}^{2i+1} j$ deb yozib olinadi. U holda, $S = \sum_{i=1}^n i / \prod_{j=1}^{2i+1} j$ ga ega bo'lamiz.

Bu esa yig'indi va ko'paymani hisoblash algoritmini qo'llashga keltiriladi.

Bu operator ichma-ich joylashgan sikl jarayoni bo'lib, n ning katta qiymatlari uchun hisoblash ko'p vaqt talab etadi. i- ning har bir qiymati uchun ichki siklda arifmetik amallar soni oshib boradi. Bu kamchilikdan qutulish uchun rekursiyani qo'llaymiz $T=1!=1$ deb olsak, bu holda $T=(2i+1)!=1*2*.....(2*i)*(2i+1)$ ga ega bo'lamiz.

Misol.

$Y=K^2+3A$ hisoblansin.

$K=1-5 \quad h=1$

$A=2-10 \quad h=2$

Program kap (input,output);

Var k,a,y:integer;

Begin

For k: =1 to 5 do

Begin

a:=2 ;

While a<=10 do

Begin

$Y:=k*k+3*a;$

Writeln ('k=',k: 3:3; ' a= ',a: 3 : 3; ' y= ', y :3:3);

a:=a+2;

end;
end;
end.

11.2. Ma'lumotlar massivi haqida tushuncha

Kundalik hayotimizda ko'p turdagi jadvallardan foydalanamiz: dars jadvali, ko'paytirish jadvali, shaxmat yoki futbol o'yinlari bo'yicha musobaqa jadvali, matematik jadval va boshqalar. Jadvalni tashkil etuvchilar uning elementlari deyiladi.

Jadval ko'rinishidagi miqdorlar bir o'lchovli (chiziqli), ikki o'lchovli (to'g'ri to'rtburchakli) yoki ko'p o'lchovli shakllarda bo'ladi. Paskal tilida bir, ikki va ko'p o'lchovli jadval miqdorlar ustida ishlash mumkin.

Bir o'lchovli jadvallar ba'zan chiziqli jadvallap deb ham yuritiladi. Chiziqli jadvallap satr yoki ustun shaklida ifodalanadi. Jadvalning har bir elementiga aniq tartib soni mos keladi va aksincha, tartib soni berilgan bo'lsa, jadvalning qaysi elementi to'g'risida so'z borayotgani tushunarli bo'ladi. Bir o'lchovli jadvalga misol sifatida haftaning har kunida o'lchangan havoning o'rtacha haroratini keltirish mumkin:

Hafta kuni	1	2	3	4	5	6	7
O'rtacha harorat	18	19	15.9	16	17	16.8	17.4

Agar bu jadvalning nomini A deb belgilasak, u holda 18 dan 17,4 gacha bo'lgan sonlar A jadval elementlari, 1 dan 7 gacha bo'lgan sonlar esa A jadval elementlarining tartib raqamini bildiradi. Bu yerda jadvalning 5-tartib raqamiga ega bo'lgan son 17 sonidan iborat bo'lib, u $A(5)=17$ kabi belgilanadi. Umuman, har bir jadval kattalik o'z nomiga va uning tashkil etuvchi elementlari o'z tartib raqamlariga ega bo'ladi, chunki kerakli element ustida amal bajarish uchun uning tartib raqamini bilish talab etiladi.

Ikki o'lchovli jadvallar to'g'ri to'rtburchakli jadvallar deb ham yuritiladi. Ikki o'lchovli jadvallarni yozish ham chiziqli jadvallarni yozish kabi bo'ladi. Faqat bu holda elementni ifodalash uchun ustun va satr qatnashadi. Bunda birinchi o'lchov satrning tartib raqami bo'lsa, ikkinchi o'lchov ustunlarniki bo'ladi. Ana shunday jadvallardan Paskal va Delphi programmalash tillarida foydalanish uchun **massiv** tushunchasi kiritilgan.

Programmalashda massivning (ya'ni jadvalning) nomi qaysi turga (butun, haqiqiy yoki belgili) tegishli bo'lsa, massivning barcha elementlari ham shu turga tegishli bo'ladi. Programmalash tillarida jadval miqdorlardan foydalanish uchun jadvalni tavsiflash va kerakli jadval elementini ko'rsatish kerak bo'ladi.

Jadvalni tavsiflash deganda jadvalning nomini, jadval elementlarining turini (tipini), jadvalning o'lchovini va jadval elementlarining sonini aniqlash tushuniladi.

Jadval elementini ko'rsatish deganda esa jadval nomi va elementining jadvaldagi tartib raqamini aniqlash tushuniladi.

Massiv elementining jadvaldagi o'rnini ko'rsatishda unda kesishgan satr va ustun tartib raqamlaridan foydalaniladi.

Massiv elementining tartib raqamlari uning indeksi deyiladi va ular qavs ichida ko'rsatiladi. Satr va ustun tartib raqamlari har doim 0 dan boshlanadi. Masalan, ikki o'lchovli E massiv berilgan bo'lsin:

$$E = \begin{bmatrix} 1 & 2 & 0 \\ 5 & 3 & 4 \\ 8 & 9 & 6 \end{bmatrix}$$

Ko'rinib turibdiki, bu jadvalda uchta satr va uchta ustun mavjud bo'lib, uning elementlari faqat butun sonlardan iborat. Demak, bu E nomli 3x3 elementdan iborat to'g'ri kvadrat jadvalni ifodalaydi. Agar satr tartib raqamini *i* va ustun tartib raqamini *j* deb belgilasak, jadvalni quyidagicha tasvirlash ham mumkin:

$$E = \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix} = \begin{bmatrix} E_{ij} \end{bmatrix}$$

bu yerda, $i=1, 2, 3$ va $j=1, 2, 3$.

Endi bu jadvalning elementlarini $E(1,1)$, $E(1,2)$ va hokazo kabi yozish mumkin. Umumiy holda indeks sifatida o'zgaruvchi yoki ifoda qo'llaniladi. Masalan, $i=1$ va $j=2$ bo'lsa, $E(i, j) = 2$ bo'ladi. Bu birinchi satr va uchinchi ustun kesishgan joydagi elementdir.

Paskal tilida programma tuzganda massivlarni dastlab programma sarlavhasida xuddi oddiy o'zgaruvchilarni tasvirlagandek tasvirlash mumkin.

<Massiv nomi>:Array[o'lcham] of <element turi>;

Masalan:

A,B:Array[1..100] of real;

C,A1,D:Array[1..10,1..15] of real;

Bu erda A va B massivlari 100tadan elementga ega. C,A1,D1 massivlari esa $10 \times 15 = 150$ tadan elementga ega.

Massivlarni e'lon qilishdan maqsad massiv elementlari uchun kompyuter xotirasidan joy ajratishdir.

Massiv elementlari qiymatlarini kiritish uchun sikl operatorlaridan foydalaniladi.

Misol: For i:=1 to 10 do Read(A[i]);

Bu misolda A massivning 10 ta elementi qiymatini ekrandan ketma-ket kiritish kerak bo'ladi. Xuddi shunday massiv qiymatlarini ekranga chiqarish ham mumkin.

Misol: For i:=1 to 10 do Write(A[i]);

Dasturda massiv elementlarini ishlatganda ularning indeksi e'lon qilingan chegaradan chiqib ketmasligi kerak.

11.3. Massiv elementlarini saralash bo'yicha dasturlarni tuzish

Massivni saralashning bir necha usullari (algoritmlari) mavjud. Ulardan quyidagi usullarni qarab chiqamiz:

- oddiy tanlash usuli;
- oddiy almashtirish usuli.

Tanlash usuli yordamida massivni o'sish bo'yicha saralash algoritmi quyidagicha:

1. Massivning birinchi elementidan boshlab qarab chiqilib eng kichik element topiladi.

2. Birinchi element bilan eng kichik element joylari almashtiriladi.

3. Ikkinchi elementidan boshlab qarab chiqilib eng kichik element topiladi.

4. Ikkinchi element bilan eng kichik element joylari almashtiriladi.

5. Bu protsess bitta oxirgi elementgacha takrorlanadi.

Bu algoritmi dasturi quyidagicha bo'ladi:

Program Sort;

Const n=5;

Var i, j, min, k, buf: Integer; a: Array[1..n] of Integer;

Begin

Writeln ('Massivni saralash');

Write (n:3, ' -ta massiv elementini kiriting');

For k:=1 to n Do Read(a[k]);

For i:=1 to n-1 Do

Begin { kichik elementni topish }

min:=i;

For j:=i+1 to n Do

Begin

If a[j]<a[min] then min:=j;

buf:=a[i]; a[i]:=a[min]; a[min]:=buf;

For k:=1 to n Do Write (a[k], ' ');

```
        Writeln;  
    End;  
End;  
Writeln('Massiv saralandi.');
```

End.

Dastur natijasi:

Massivni saralash

5 ta massiv elementini kiriting

12 -3 56 47 10

Saralash

-3 12 56 47 10

-3 10 56 47 12

-3 10 12 47 56

-3 10 12 47 56

Massiv saralandi.

Almashtirish usuli yordamida massiv elementlarini o'sib borishda saralash algoritmi quyidagicha:

1. Massivning birinchi elementidan boshlab ketma-ket hamma qo'shni elementlar bir-biri bilan solishtirilib, agar birinchisi ikkinchisidan kichik bo'lsa ular joyi almashtirilib boriladi.

2. Bu protsess davomida kichik qiymatli elementlar massiv boshiga katta elementlar esa oxiriga siljutilib boriladi. SHu sabab bu usul «puzirka» usuli ham deyiladi.

3. Bu protsess massiv elementlar sonidan bitta kam marta takrorlanadi.

Masalan:

3 2 4 5 1 bunda 3 bilan 2 va 5 bilan 1 almashtiriladi.

2 3 4 1 5 bunda 4 bilan 1 almashtiriladi.

2 3 1 4 5 bunda 3 bilan 1 almashtiriladi.

2 1 3 4 5 bunda 2 bilan 1 almashtiriladi.

1 2 3 4 5

Bu algoritm dasturi quyidagicha bo‘ladi:

Program Sort;

Const n=5;

Var i,j,min,k,buf: Integer; a: Array[1..n] of Integer;

Begin

Writeln ('Massivni puzirek (ko‘pikcha) usulida saralash’);

Write (Size:3,’ta massiv elementini kiriting’);

For k:=1 to n Do Read(a[k]);

Writeln ('Saralash’);

For i:=1 to n-1 Do

Begin

For k:=1 to n-1 Do

Begin

If a[k]>a[k+1] then

Begin

buf:=a[k]; a[k]:=a[k+1]; a[k+1]:=buf;

End;

End;

For k:=1 to n Do Write (a[k],’ '); Writeln;

End;

Writeln('Massiv saralandi.’);

End.

Dastur natijasi:

Massivni puzirek usulida saralash

5 ta massiv elementini kiriting

3 2 4 1 5

Saralash

2 3 4 1 5

2 3 1 4 5

2 1 3 4 5

1 2 3 4 5

Massiv saralandi.

11.4. Massivda eng kichik elementni izlash algoritmining dasturi

Massivda eng kichik yoki eng katta elementni izlash algoritmi ma`lumki birinchi element eng kichik (katta) deb olinib keyin boshka elementlar bilan ketma-ket solishtirilib chiqiladi. Solishtirish oxirgi elementgacha bajariladi. Quyida bu algoritm dasturi keltirilgan:

Program MinMax;

Var i,min: Integer; a: Array[1..10] of Integer;

Begin

Writeln ('Massivdan eng kichik elementni izlash');

Write (' 10-ta massiv elementini kiriting');

For i:=1 to 10 Do Read(a[i]);

min:=1;

For i:=2 to 10 Do

If a[i]<a[min] Then min:=i;

Writeln('Izlanayotgan eng kichik element:',a[min]);

Writeln('Element nomeri', min);

End.

Tayanch iboralar

tashqi sikl, ichki sikl, ikki o'lchovli jadval, massiv, element, indeks, massivni saralash, tanlash usuli, almashtirish usuli, eng kichik element, eng katta element.

Nazorat va muhokama uchun savollar

1. Murakkab takrorlanuvchi hisoblash jarayoniga ta'rif bering.
2. Massiv deganda siz nimani tunasiz?
3. Bir o'lchovli massivlar qanday xususiyatlarga ega?
4. Ikki o'lchovli massivlarga ta'rif bering.
5. Massiv elementlarini saralashning qanday usullarini bilasiz? Ularga misollar keltiring.

12-bob. PROTSEDURA VA FUNKSIYALAR

12.1. Paskal tilida protsedura konstruksiyasi.

12.2. Funksiya qism dasturi haqida.

12.3. Protsedura va funksiya qism dasturiga murojaat qilish.

12.4. Protsedura va funksiya qism dasturlarining parametrlari.

12.5. Standart protsedura va funksiyalarga misollar.

12.1. Paskal tilida protsedura konstruksiyasi

Dasturlash jarayonida murakkab dasturlarning bir necha joyida bir xil vazifani bajaruvchi operatorlar guruhini qo'llashga to'g'ri keladi va dasturda bir-biriga aynan o'xshash bir necha qismlar vujudga keladi.

Kompyuter xotirasini va dastur tuzuvchining vaqtini tejash maqsadida ushbu qismdagi o'xshash operatorlar bir marta asosiy dasturdan ajratib yoziladi va unga asosiy dastur bajarilishi jarayonida murojaat qilinadi.

Dasturning ixtiyoriy qismidan murojaat qilib, bir necha bor ishlatish mumkin bo'lgan bunday operatorlar guruhiga qism dastur deb ataladi va u asosiy dastur bilan bir butunni tashkil etadi.

Qism dasturlarni ishlatish dasturning hajmini kichraytiradi va uning ko'rinishini, o'qilishini hamda xatolar sonining kamayishiga olib keladi.

Turbo Paskal tilida qism dasturlar protsedura va funksiya qism dasturi ko'rinishida bo'ladi.

Protsedura qism dasturi qism dasturda bir necha natija hosil qilish kerak bo'lgan hollarda ishlatiladi va u asosiy dasturda struktura bo'yicha qism dastur bo'limida, ya'ni o'zgaruvchilar va operatorlar bloki bo'limlari orasida tasvirlanadi.

Har bir protsedurani tasvirlash sarlavhadan boshlanadi. Bunda protseduraning nomi va parametrlarning ro'yxatlari ularning turlari bilan birga beriladi.

Protsedura parametrsiz ham bo'lishi mumkin. Bu holda protsedura sarlavhasida faqat uning nomi ko'rsatiladi. Parametrlar yordamida belgilarni protseduraga berish

hamda hosil bo'lgan natijani protsedurani chaqirayotgan dasturga berish global parametrlar yordamida amalga oshiriladi.

Protsedura sarlavhasining umumiy ko'rinishi quyidagicha bo'ladi:

PROCEDURE < nom > (< formal parametrlar ro'yxati >);

bu erda PROCEDURE - xizmatchi so'z; <nom>- protseduraning nomi; <formal parametrlar ro'yxati>- turlari ko'rsatilgan berilganlarni va natijalarni ifodalaydigan nomlar turlari bilan sanab o'giladigan qator.

Formal parametrlar ro'yxatida parametr-qiymat, parametr-o'zgaruvchilar (bularning oldida VAR xizmatchi so'zi turishi kerak), parametr-protseduralar (bularning oldida PROCEDURE xizmatchi so'zi turishi kerak) va parametr-funksiyalar (bularning oldida FUNCTION xizmatchi so'zi turishi kerak) bo'lishi mumkin.

Protsedura sarlavhasidan keyin keladigan bo'limlar asosiy dasturda qanday tartibda kelsa, protsedurada ham shu tartibda keladi.

Protsedura bloki nuqta vergul (;) bilan tugaydi.

Misol 1. Berilgan x va y sonlari uchun $U = \text{MAX}(x + y, x * y)$;

$V = \text{MAX}(0.5, U)$ topilsin.

Program max;

Var x, y, u, v, s: real;

Prosedure *prosed* (p1, p2: real);

Begin

If p1 > p2 **then** s := p1 **else** s := p2

End;

Begin

Write ('x = '); **readln** (x);

Write ('y = '); **readln** (y);

Prosed (x + y, x * y); u := s;

Prosed (0.5, u); v := s;
Writeln (u:3:1, v);
End.

12.2. Paskalda funksiya tushunchasi

Funksiya bu shunday qism dasturki, bunda bitta natija chiqadi va u funksiya nomida hosil bo'ladi. Shuning uchun funksiya protsedura qism dasturining xususiy holi bo'lib, undan farqi shundaki, birinchidan, funksiya qism dasturining natijasi bitta qiymatdan iborat bo'ladi.

Protsedura qism dasturida natija esa bitta yoki bir necha bo'lishi mumkin. Funksiya qism dasturining natijasi asosiy dasturga funksiyaning natijasi sifatida beriladi.

Protsedura qism dasturi natijalari esa uning parametrlari qiymati sifatida hosil bo'ladi.

Funksiya qism dasturini aniqlanishi protsedura qism dasturini aniqlanishiga o'xshaydi va u ham sarlavha va blokdan iborat bo'ladi.

Funksiya qism dasturining umumiy ko'rinishi quyidagicha bo'ladi:

Function <nom> (<formal parametrlar ro'yxati>): <tur>;

bu erda function - xizmatchi so'z, <nom> - funksiyaning nomi, < formal parametrlar ro'yxati> - kiritiladigan sohta parametrlar sanab o'tiladi, <tur> - funksiya natijasining turi.

Bunda natija funksiya nomida hosil bo'ladi. Bundan tashqari, funksiyaning parametrsiz ko'rinishini ham yozish mumkin:

Function <nom>: <tur>;

Har ikkala holda ham funksiya natijasi funksiya qism dasturi tanasida funksiya nomiga hech bo'lmaganda bir marta o'zlashtirilishi shart.

Misol: Ikki o'zgaruvchi faktorialining ayirmasini hisoblash dasturi funksiya qism dasturi yordamida amalga oshirilsin.

$F=m!-n!$

```
Program factorial;
var f,m,k,:integer;
function fakt (n: integer): integer;
var p,i:integer;
begin
p:=1;
for i : =2 to n do
p:=p*i;
fact := p;
end;
begin
readln (m,k);
f := fakt(m)-fakt(k);
writeln ('f='; f);
end.
```

12.3. Protsedura va funksiya qism dasturiga murojaat qilish

Protsedurani bajarish uchun asosiy dasturning ixtiyoriy (kerakli) joyidan unga murojaat qilish kerak.

Protseduraga murojaat va uni bajarish quyidagi operator yordamida amalga oshiriladi:

<protsedura nomi>(<aniq parametrlar ro'yxati>);

Bunda qism dastur tasvirida keltirilgan protseduraga murojaat qilishda formal va

aniq parametrlar orasida to'liq moslik bo'lishi shart, ya'ni sohta va aniq parametrlar soni bir xil bo'lishi, kelish tartiblari mos tushishlari va ularning turlari bir xil bo'lishi kerak.

Protsedura chaqirilganda formal parametrlar o'rniga aniq parametrlar mos qo'yiladi.

Bunda parametr-qiymatlar o'rniga kiritiladigan malumotlar beriladi. Natija esa parametr-o'zgaruvchilarda hosil bo'ladi.

Protsedura chaqirilganda sarlavhada ko'rsatilgan formal parametrlar aniq parametrlar bilan ularning kelishi tartibida almashtiriladi, ya'ni birinchi formal parametrqa birinchi aniq parametr, ikkinchi formal parametrqa ikkinchi aniq parametr va h.k. mos qo'yiladi.

Funksiya qism dasturiga murojaat uning nomi orqali amalga oshiriladi. Funksiyaga murojaat qilinadigan joyda funksiyaning nomi yoziladi va undan keyin qavs ichida aniq parametrlar yoziladi.

Agar funksiya qism dasturi parametrsiz ko'rinishda bo'lsa, u holda faqat uning nomi yoziladi.

12.4. Protsedura va funksiya qism dasturlarining parametrlari

Protsedura va funksiya qism dasturlari aniqlanayotganda ularning formal parametrlar ro'yxatida parametrlarning to'rt xil ko'rinishi ko'rsatilishi mumkin:

- 1) parametr- qiymatlar;
- 2) parametr- o'zgaruvchilar;
- 3) parametr- protseduralar;
- 4) parametr- funksiyalar.

Formal va aniq parametrlar protsedura va funksiya qism dasturlarida bir xil qayta ishlanadi.

Har bir parametr turi ko'rsatilgan holda ro'yxatga kiritiladi.

Masalan:

PROCEDURE SUMMA(X,Y,Z: REAL);

PROCEDURE SUMMA1(A,B: REAL; N: INTEGER);

PROCEDURE TH(VAR A,R: REAL);

FUNCTION SS1(B: REAL; I,J: INTEGER): REAL;

Agar ikkita yoki undan ko‘p formal parametrlar ro‘yxatda bir xil turda bo‘lsa, u holda ularni bitta tasvirlashga kiritish mumkin. Formal parametrlarni ro‘yxatda ixtiyoriy tartibda ko‘rsatish mumkin. Protsedura va funksiya qism dasturlariga murojaat qilinayotganda formal parametrlar qanday tartibda kelsa, aniq parametrlar ham shu tartibda kelishi kerak.

Protsedura va funksiya qism dasturlarini o‘z ichiga olgan asosiy dasturda nom tanlanayotganda lokal va global parametrlarning bir xil bo‘lmagani maqul.

Parametr-qiymatlar parametrlari berilganlarni qism dasturlar (protsedura va funksiya) ga uzatish uchun ishlatiladi.

Bunda yuqorida aytilganidek, ular formal parametrlar ro‘yxatida turlari ko‘rsatilgan holda, bir-birlaridan vergul bilan ajratilgan holda sanab o‘tiladi.

12.5. Standart protsedura va funksiyalarga misollar

Paskal tilida bir qancha maxsus protsedura va funksiyalar mavjud bo‘lib, ular quyidagi guruhlariga bo‘linadi:

- qatorni qayta ishlash;
- fayllar bilan ishlash;
- dinamik o‘zgaruvchilar uchun xotirani boshqarish;
- arifmetik funksiyalar;
- ekran bilan ishlash.

Ularning ayrimlarini ko‘rib chiqamiz:

Halt- dasturni bajarishdan to‘xtatish;

Odd(i)- I-toq bo‘lsa “True” aks holda “False” qiymat oladi;

Edit- bajarilayotgan blokdan chiqish;

Random- 0 dan 1 gacha bo‘lgan sonni tasoddifan olish;

GotoXY(x,y)- kursorni ko‘rsatilgan joyga qo‘yish;

ClrScr- ekranni tozalab, kursorni ekran boshiga qo'yish;

Trunc- argumentning butun qismi:

Str(I;Var S:String)- raqamni simvolga o'tkazish (I-ifoda yoki o'zgaruvchi);

Val(S:String; Var P;ko:Integer)- simvolni raqamga o'tkazish (P-o'zgaruvchi);

Length (S:String)- qator uzunligini aniqlash.

Pos(st1,st) - qatordagi qator qismi holatini aniqlash.

Misol: st:='Toshkent'; st1:='kent'; p:=pos(st1,st); Javob: p=4. Agar izlanayotgan qator qism qatorda yo'q bo'lsa qiymat nulgga teng bo'ladi.

Copy(st,m,n) - qatordan fragment kesib oladi.

Misol: st:='Toshkent'; p:=Copy(st,4,4); Javob: p='kent'.

Delete(st,m,n) - qatordan fragment kesib olib tashlaydi.

Misol: st:='Toshkent'; p:=Delete(st,4,4); Javob: p='Tosh'.

Tayanch iboralar

qism dastur, parameter, protsedura, formal parametrlar, parametr-qiymat, parametr-o'zgaruvchi, funksiya, aniq parametrlar, standart protsedura va funksiyalar.

Nazorat va muhokama uchun savollar

1. Qism dastur tushunchasiga ta'rif bering.
2. Protsedura qism dasturi nima uchun foydalaniladi va unga qanday murojaat qilinadi?
3. Funksiya qism dasturi nima uchun foydalaniladi va unga qanday murojaat qilinadi?
4. Protsedura va funksiya qism dasturlarining farqlari nimadan iborat?
5. Parametrsiz protsedura qaysi hollarda foydalaniladi?
6. Formal parametrlar sifatida massivlardan foydalanish qanday amalga oshiriladi?
7. Lokal va global parametrlarga ta'rif bering.

GLOSSARIY

Atamaning o'zbek tilida nomlanishi	Atamaning ingliz tilida nomlanishi	Atamaning rus tilida nomlanishi	Atamaning ma'nosi
Algoritm	Algorithm	Алгоритм	berilgan masalani yechish uchun ma'lum tartib bilan bajarilishi kerak bo'lgan chekli sondagi buyruqlar ketma-ketligini tushunamiz.
Algoritm-lashtirish jarayoni	The algorithmization process	Процесс алгоритмизации	masalani echish algoritmini tuzishni ta'minlaydi va birlamchi axborotni natijaviyga aylantirish bosqichlar ketma-ketligini aniqlash va ifoda etish uchun xizma-t qiladi.
Blok-chizmalar	Flowchart	Блок-схема	bu algoritm bosqichlar ketma-ketligini grafik jihatdan tasvirlab berishdir.
Boshqarish operatorlari	Operator management	Оператор управления	dastur operatorlarini ketma-ket bajarilishini tashkil etadi.
Chiziqli hisoblash jarayoni	Linear computing process	Линейный вычислительный процесс	shunday jarayonki, unda hisoblashning barcha bosqichlari chiziqli ketma-ketlik ko'rinishida bajariladi
Dastur	Program	Программа	rasmiy kompyuter tilida bayoni, zarur harakatlar ketma-ketligini tushunadi
Dasturlash	Programming	Программирование	bu shaxsiy kompyuter dasturlash tilida algoritmni tasvirlash jarayonidir.

Funksiya	Function	Функция	bu shunday qism dasturki, bunda bitta natija chiqadi va u funksiya nomida hosil bo‘ladi
Kiritish-chiqarish operatorlari	Input/output statement	Оператор ввода-вывода	tezkor xotiraga boshlang‘ich ma'lumotlarni kiritadilar va yozuvga chiqaruvchi qurilmalarga hisoblashlar natijasini chiqaradilar.
Kvadrat matritsa	Square matrix	Квадратная матрица	agar qatorlar soni ustunlar soni bilan teng bo‘lsa, unda matritsa kvadrat hisoblanadi
Ma`lumotlarni saralash	Sorting data	Сортировка данных	bu axborotlarni ishlab chiqish bo‘lib, buning natijasida uning elementlari (yozuvlari) bu axborotlarning ba`zi bir alomatlarining miqdoriga ko‘ra belgilangan izchillikda joylashadilar.
Masalaning qo‘yilishi	Formulation of the problem	Постановка задачи	avvalom bor uni to‘g‘ri va ravon ifoda qilmoq, hamda uni to‘g‘ri yechish uchun qo‘yilayotgan hamma shartlarini aniqlash tushuniladi.
Murakkab siklik algoritmi	Complex repeating algorithm	Сложный циклический алгоритм	unda siklning tanasi takrorlanuvchi hisoblash jarayonlaridan iborat bo‘ladi
Murakkab tarmoqlan-gan hisoblash jarayoni	A complex branching computational process	Сложный разветвляющийся вычислительный процесс	bitta yoki ikkala tarmoqlarda ham mantiqiy ifodalarni hisoblash bloklari bo‘ladi
O‘zlashtirish	Assignment	Оператор присваивания	dasturda o‘zgaruvchilarga ma’lum qiymatlarni o‘zlashtirish uchun

operatorlari	operator		ishlatiladi.
Oddiy takrorlanuvchi algoritmi	A simple cyclic process	Простой циклический процесс	unda siklning tanasi chiziqli yoki tarmoqlangan hisoblash jarayonlaridan iborat bo'ladi
Oddiy almashuv (ko'pik) saralash metodi	Method of sorting with a simple exchange (bubble)	Метод сортировки с помощью простого обмена (пузырька)	ushbu metodning g'oyasi shundan iboratki, agar ikkita yonma yon turuvchi elementlar tartib bo'yicha joylashmagan bo'lsalar, unda ular joylarini almashtiradilar. Bu jarayon elementlar tartibga solinmaguncha takrorlanaveradi.
Oddiy tanlov saralash metodi	Sorting method with a simple choice	Метод сортировки с помощью простого выбора	oddiy tanlov metodining g'oyasi massivning birinchi elementidan boshlab to massivning oxirigacha bo'lgan eng kichik (eng katta) elementini izchil qidirish va birinchi elementni topilgan miqdorga almashtirishdan iborat bo'ladi. Birinchi element eng kichik elementning o'rniga qo'yiladi.
Protsedura	Procedure	Процедура	qism dasturda bir necha natija hosil qilish kerak bo'lgan hollarda ishlatiladi va u asosiy dasturda struktura bo'yicha qism dastur bo'limida, ya'ni o'zgaruvchilar va operatorlar bloki bo'limlari orasida tasvirlanadi
Qism dastur	Subprogramme	Подпрограмма	dasturning ixtiyoriy qismidan murojaat qilib, bir necha bor ishlatish

			mumkin bo'lgan operatorlar guruhi
Saralash	Sorting	Сортировка	informatika va hisoblash matematikasining eng muhim vazifasi bo'lib, u algoritmik masalalarning yechishning ko'p resurslar talab qiluvchi usullariga kiradi.
Shartli o'tish operatori	Conditional branch operator	Оператор условного перехода	ushbu operator yordamida dasturda boshqaruvni ma'lum shart asosida u yoki bu tarmoqqa uzatish jarayoni amalga oshiriladi
Shartsiz o'tish operatori	The unconditional branch operator	Оператор безусловного перехода	ushbu operator yordamida dasturda boshqaruvni to'g'ridan-to'g'ri biron-bir operatorga uzatish, ya'ni dasturning bajarilish ketma-ketligini buzish jarayoni amalga oshiriladi
Sikl parametri	Cycle parameter	Параметр цикла	bu takrorlanish ichida qiymatlari o'zgarib boradigan o'zgaruvchisi
Siklning tanasi	Body of the cycle	Тело цикла	hisoblash jarayonlarini ko'p marta qaytariladigan qismi.
Sozlash	Debugging	Отладка	bu dasturdagi xatoliklarni aniqlash va ularni to'g'rilash jarayoni.
Takrorlanuvchi (siklik) hisoblash jarayonlari	Repeating (cyclic) computing process	Повторяющийся (циклический) вычислительный процесс	hisoblash bosqichlarini ko'p marta qaytarishi mumkin bo'lgan hisoblash jarayonlari.

Tanlash operatori	Selection operator	Оператор выбора	tarmoqlanish jarayonini berilgan bir necha operatorlardan birini tanlash yoʻli bilan amalga oshiradi
Tarmoqlangan hisoblash jarayoni	Branching Computing Process	Разветвляющийся вычислительный процесс	shunday jarayonki, unda u birlamchi yoki oraliq maʼlumotlar xususiyatidan kelib chiqqan holda bir yoki bir necha yoʻnalish boʻyicha bajarilishi mumkin boʻladi.
Operator	Operator	Оператор	kompyuter dasturi bajarilayotganda maʼlum bir hara-katlarni amalga oshirish boʻyicha koʻrsatma beradi.

FOYDALANILGAN ADABIYOTLAR RO'YXATI

Asosiy adabiyotlar

1. Algoritmash va dasturlash asoslari: kasb-hunar kollejlari uchun o'quv qo'llanma/A.R. Azamatov; O'zbekiston Respublikasi Oliy va o'rta maxsus ta'lim vazirligi, O'rta maxsus, kasb-hunar ta'limi markazi. To'rtinchi nashri. – T.: Cho'lpon nomidagi nashriyot-matbaa ijodiy uyi, 2013. – 232 b.

2. J.B.Dixit. Fundamental of computer programming and IT. – Laxmi Publication Pvt. Ltd., 2011. – 557 p.

3. R.Dadabayeva, Sh.Nasridinova, N.Shoaxmedova, L.Ibragimova, Sh.Ermatov. Axborot-kommunikatsion texnologiyalar va tizimlar.-T.: "Sano-standart" nashriyoti. O'quv qo'llanma. 2017-552 bet.

4. Семакин И.Г., Шестаков А.П. Основы алгоритмизации и программирования Издательство: Академия. 2016 г. - 304 с.

5. Основы алгоритмизации и программирования: учеб. пособие /Т.А. Жданова, Ю.С. Бузыкова.– Хабаровск: Изд-во Тихоокеан. гос.ун-та, 2011.– 56 с.

6. Основы алгоритмизации и программирования : учебное пособие /Г. Р. Кадырова. – Ульяновск : УлГТУ, 2014. – 95 с.

7. Основы алгоритмизации: учеб.-метод. пособие для студ. оч. и заоч. обуч. технич. специальностей/ В.И. Логинов, Л.Н. Шемагина. – Н. Новгород: Изд-во ФГОУ ВПО«ВГАВТ», 2010. – 81 с.

8. Трофимов В.В., Павловская Т.А. Алгоритмизация и программирование. Учебник для академического бакалавриата. Издательство: Москва – ЮРАЙТ. 2018 г. - 137 с.

Qo'shimcha adabiyotlar

9. O'zbekiston Respublikasi Konstitutsiyasi - T.: O'zbekiston, 2017. - 46 b.

10.O'zbekiston Respublikasining "Axborotlashtirish to'g'risida" gi Qonuni. «Xalq so'zi», 2004, 6 may, 1-2-bet.

11.O'zbekiston Respublikasi Prezidenti Shavkat Mirziyoevning Oliy Majlisga Murojaatnomasi.// «Xalq so'zi», 2018 y., 29 dekabr.

12.O‘zbekiston Respublikasi Prezidentining “Axborot texnologiyalari va kommunikatsiyalari sohasini yanada takomillashtirish chora-tadbirlari to‘g‘risida”gi 2018 yil 19 fevraldagi PF-5349-sonli Farmoni.

13.Гагарина Л.Г., Кокорева Е.В., Виснадул Б.Д. Технология разработки программного обеспечения: учебное пособие /под ред. Л.Г. Гагариной. – М.: ИД «ФОРУМ»: ИНФРА-М, 2008. – 400 с.

14.Голицына О.Л., Максимов Н.В., Партыка Т.Л., Попов И.И. Информационные технологии: учебник. – 2-е изд., перераб. и доп. – М.: ФОРУМ: ИНФРА-М, 2008. – 608 с.

15.Иванова Г.С. Технология программирования : учебник - 3-е изд., стер. - М.: КНОРУС, 2016. — 334 с.

16.Информатика: Практикум по технологии работы на компьютере. / Под ред. В. Макаровой. - 3-е изд., перераб. - М.: ФиС, 2008. –256 с.

17.Информационные технологии управления: Учеб. пособие для вузов /Под ред. проф. Г.А Титоренко. М: ЮНИТИ-ДАНА, 2015. - 439 с.

18.Информационные системы и технологии в экономике и управлении: учебник для бакалавров / под ред. В. В. Трофимова. - 4-е изд., перераб, и доп. - М.: Издательство Юрайт, 2016. – 544 с.

19.Исаев Г.Н.. Информационные системы в экономике: Учебник для студентов вузов. – М.: Омега-Л, 2012. -462 с.

20.Мельников П.П. Компьютерные технологии в экономике. Учеб.пособие. – М.: КНОРУС, 2012. – 224с.

21.Павловская Т.А. Программирование на языке высокого уровня Паскаль (2-е изд.). М.: НОУ «Интуит», 2016 г.- 154 с.

22.Уткин В.Б., Балдин К.Б. Информационные системы в экономике. – Academia, 2015. – 288 с.

Internet saytlari

23.www.gov.uz - O‘zbekiston Respublikasi davlat portali.

24.<https://tproger.ru/> - dasturchilar uchun sayt.

25. www.tsue.uz - Toshkent davlat iqtisodiyot universiteti web sayti.

26.<https://developers.google.com/android/> - dasturchilar uchun sayt

27.www.lex.uz – O‘zbekiston Respublikasi Qonun hujjatlari ma’lumotlari milliy bazasi.

28.www.ziyonet.uz - O‘zbekiston Respublikasi ta’lim portali

29.<http://www.stat.uz> – O‘zbekiston Respublikasi Davlat Statistika Qo‘mitasining rasmiy sayti.

30.<http://wikipedia.org> - Erkin entsiklopediya.

31.www.UzReport.uz – UzReport agentligining axborot sayti.

32.<http://www.citforum.ru> – axborot texnologiyalar bo‘yicha axborot sayti.

33.www.webofscience.com - Xalqaro ilmiy maqolalar platformasi

34.www.scopus.com - Xalqaro ilmiy maqolalar platformasi

A.A. Sobirov, Dadabayeva R.A., D.P. Xashimova

ALGORITMLASHTIRISH VA DASTURLASH ASOSLARI

O‘quv qo‘llanma

“IQTISODIYOT” – 2019.

Muharrir
Mirhidoyatova D.

Musahhih
Matxo`jayev A.O.

Litsenziya AI № 240 04.07.2013. Terishga berildi 05.09.19. Bosishga ruxsat etildi 05.09.2019. Qog‘oz bichimi 60x80 1/16. Times garniturası. Ofset bosma. Ofset qog‘ozı. Shartlı bosma tabog‘ı 14,7. Hisob nashr varag‘ı 14,1. Adadı ____ nusxa.

“IQTISODIYOT” nashriyoti DUKning matbaa bo‘limida chop etildi.
100003. Toshkent shahri O‘zbekiston shohko‘chasi, 49-uy.

000000 Algoritmlashtirish va dasturlash asoslari. O'quv
qo'llanma. /Sobirov A.A., Dadabayeva R.A.,
Xashimova D.P.- T.: « IQTISODIYOT», 2019. -
132 bet.

Sobirov A.A.,
Dadabayeva R.A.,
Xashimova D.P.

ISBN 000000000

UO`K 000000
KBK 000000