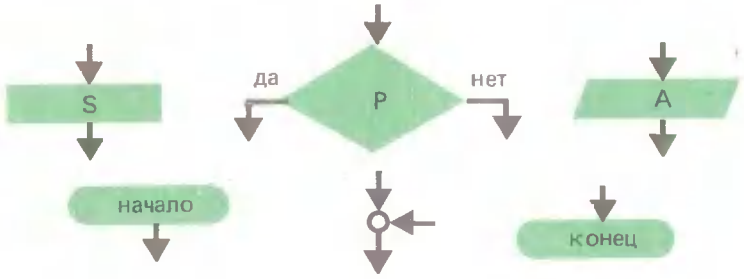
55 к.

Информатика и вычислительная техника

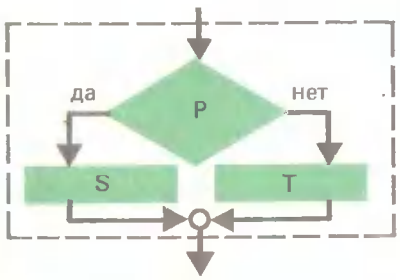
Элементы программирования на микрокалькуляторах

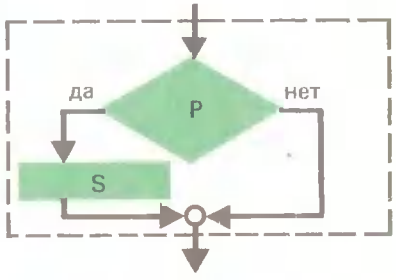
Программирование на языке БЕЙСИК

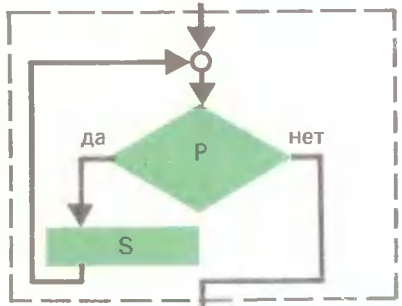
Графические элементы схем

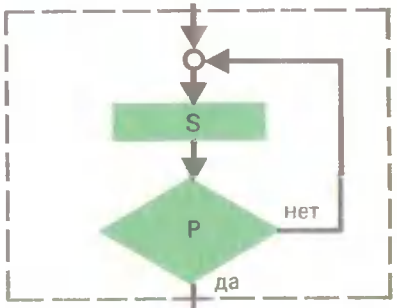


Базовые структуры









Информатика и вычислительная техника

Факультативные курсы:

Элементы программирования на микрокалькуляторах

Программирование на языке БЕЙСИК

Учебное пособие
для 9—10 классов

Допущено
Министерством просвещения БССР

Минск «Народная асвета» 1988

ББК 73я72
И 74
УДК 002.6+681.3 (075.3)

Авторы:
А. И. Павловский, В. В. Пенкрат, А. Т. Кузнецов, Л. Н. Насенникова

Рецензенты:
д-р техн. наук А. Н. Останин, канд. техн. наук Л. П. Матюшков

И $\frac{4306022100-031}{М\ 303(03)-88}$ 120—87

ISBN5—341—00350—0

© Издательство «Народная асвета» 1988

Общие сведения о вычислительных машинах

Мы живем в век научно-технической революции, в эпоху широкой автоматизации процессов производства и управления. В народном хозяйстве нашей страны постоянно расширяются масштабы работ по созданию гибких автоматизированных производств — гибких в смысле быстрой перестройки на производство новых изделий; автоматизации функций управления отдельными предприятиями, производственными объединениями, целыми отраслями промышленности. И все эти работы проводятся на базе электронно-вычислительной и микропроцессорной техники, которая является мозговым центром автоматизированного производства и управления. Экономические и социальные процессы также нужно постоянно анализировать с тем, чтобы принимать верные решения. Анализировать — это значит знать текущее состояние процесса, т. е. знать тысячи и десятки тысяч связей и их влияние друг на друга. Такой анализ невозможно выполнить вручную, нужны мощные вычислительные средства.

В истории развития вычислительной техники можно выделить четыре периода: домеханический, механический, электромеханический и электронный.

Домеханический период называют периодом абака. Принято считать, что он продолжался с древнейших времен до начала XVII в.

Простейшим «устройством», которое помогало при счете, были десять пальцев рук. Происхождение десятичной системы счисления связано именно со счетом на пальцах. Одним из первых искусственных счетных приборов была бирка. Она служила для запоминания чисел с помощью насечек. С развитием человечества совершенствовались счетные приборы. Появились палочки Непера. Наиболее совершенным прибором этого периода стал абак. Современным представителем абака являются конторские счеты, которые еще можно встретить в повседневной жизни.

Механический период вычислительной техники продолжался с начала XVII до конца XIX в. За это время было создано много разнообразных машин. В России наибольшее распространение получила машина В. Т. Однера, усовершенствованной моделью которой является арифмометр «Феликс».

Электромеханический период характеризуется появлением счетно-аналитических машин. Первая такая машина была построена Г. Холлеритом. В 1890 г. в США она была применена при обработке материалов переписи населения. В этой машине впервые управление работой осуществлялось с помощью перфокарт.

Развитие электроники повлекло за собой переход к электронной вычислительной технике. Первые промышленные образцы

электронных вычислительных машин (ЭВМ) появились в середине нашего века. Первая отечественная ЭВМ была разработана в 1948—1951 гг. под руководством академика С. А. Лебедева.

Сейчас уже можно говорить о четырех поколениях ЭВМ: первое — на электровакуумных лампах, второе — на полупроводниках, третье — на интегральных схемах (ИС), четвертое — на больших интегральных схемах (БИС).

Принцип действия электронных вычислительных машин был высказан еще в 1834 г. английским инженером и математиком Ч. Беббеджем. Понадобилось более ста лет, прежде чем этот принцип был реализован. За время своего существования ЭВМ сделали грандиозный скачок как в своем совершенствовании, так и в расширении поля применимости. Если в первых машинах можно было хранить несколько десятков чисел, то сейчас «память» современной ЭВМ способна хранить информацию крупной публичной библиотеки. Необычайно расширилась сфера применения ЭВМ. Так, ни одно сообщение ТАСС в связи с запуском очередного спутника или космической станции не обходится без уже ставшей традиционной фразы: «координационно-вычислительный центр ведет обработку поступающей информации».

В настоящее время используются универсальные ЭВМ, мини-ЭВМ, микро-ЭВМ, персональные ЭВМ, микрокалькуляторы разных типов. И для каждого типа вычислительных устройств есть своя сфера применения. Универсальная ЭВМ хороша для составления прогноза погоды или учета материалов на складе, где хранится 40—50 тысяч разных деталей. Мини- и микро-ЭВМ удобны для управления небольшими предприятиями или технологическими процессами. Микрокалькуляторы используются научными, инженерно-техническими работниками, студентами и школьниками для выполнения не очень сложных расчетов.

Последнее достижение в области электронных вычислительных машин — персональные ЭВМ, которые построены на основе микропроцессоров и предназначены самому широкому кругу индивидуальных пользователей. Работают они в режиме диалога, правила которого несложные. Нужно набрать вопрос на клавиатуре, он появится на экране дисплея, и, если вопрос корректно сформулирован, машина немедленно даст ответ или же сама задаст уточняющий вопрос. С помощью персональной ЭВМ можно вести деловую и личную переписку, хранить черновики писем, адреса знакомых и друзей, вести картотеку библиотеки с записью выданных книг, каталог журнальных статей с аннотациями, решать игровые и занимательные задачи, учиться машинописи, изучать тот или иной школьный предмет или иностранный язык.

Персональная ЭВМ снабжается пакетом готовых программ, которыми можно пользоваться немедленно. Но заранее на все случаи жизни и на любой вкус программ не напасешься. Вот здесь-то и необходимо знание языка программирования. Используя этот язык, можно написать собственную программу и тем

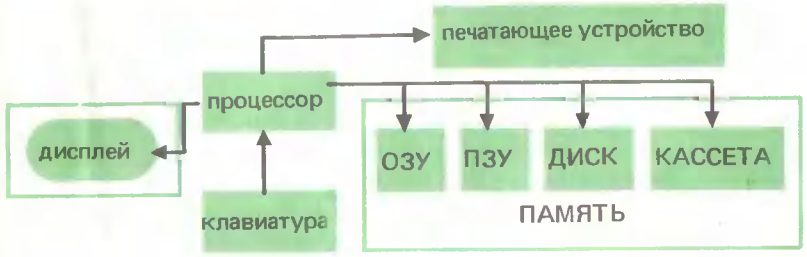


Рис. 1

самым расширить пакет программ или создать собственный пакет программ.

На основе персональных ЭВМ можно создавать автоматизированные рабочие места инженера-конструктора, архитектора, плановика, руководителя учреждения, научного работника, учителя, директора школы. Персональный компьютер справедливо называют «индивидуальным усилителем природных возможностей человеческого разума».

Общение с компьютером происходит с помощью специально созданных искусственных языков, которые называют языками программирования. Сейчас создаются новые, упрощенные языки, близкие к нашему естественному языку, с тем, чтобы вычислительными машинами могло пользоваться как можно больше людей, а не только профессионалы-программисты. Эти языки называют диалоговыми. Примером такого языка является БЕЙСИК — алгоритмический язык для начинающих.

Общая схема персональной ЭВМ приведена на рисунке 1. В ее комплект входят: процессор, дисплей с клавиатурой, печатающее устройство, внешняя память на кассетах или дисках. В некоторых случаях в комплект входит графопостроитель.

В процессоре машины происходит выполнение арифметических и логических операций. Процессор содержит также блок управления всем ходом вычисления.

Дисплей служит для отображения на экран текста программы, вводимых данных, полученных в процессе вычислений окончательных результатов. С помощью клавиатуры дисплея набираются и заносятся в оперативную память ЭВМ программы и исходные данные для вычислений, производится редактирование и исполнение программ.

Печатающее устройство используется для получения на бумаге окончательных результатов вычислений (числа, таблицы, графики). На печать также можно вывести саму программу.

Запоминающее устройство (или память) машины служит для хранения программ и данных и делится на оперативное (ОЗУ), постоянное (ПЗУ) и внешнее (диск, кассета). В оперативную память помещаются программа и исходные данные, используемые в данный момент. В постоянном запоминающем устройстве

хранятся программы алгоритмов арифметических действий, элементарных функций. Внешняя память на магнитных дисках или кассетах используется для хранения ранее составленных программ, а также для хранения интерпретатора с языка программирования, который применяется в данной машине. Интерпретатор — это большая заранее написанная программа, поставляемая вместе с персональной ЭВМ, которая автоматически переводит программу, например с языка БЕЙСИК на внутренний язык машины. Три уровня памяти машины (оперативная, постоянная и внешняя) позволяют быстро манипулировать очень большой информацией. Нужная для работы ЭВМ информация по команде «заносится» в ее оперативную память с диска или кассеты.

Процесс вычислений на машине очень сложный. Необходимо обеспечить согласованную работу всех устройств ЭВМ. Эти функции выполняет так называемая операционная система (ОС) — большая сложная программа, которая также поставляется вместе с ЭВМ. ОС, в частности, анализирует введенную программу и выдает на экран сообщения об ошибках в ней.

Принято говорить, что общая логическая организация вычислительной системы есть архитектура ЭВМ. В понятие архитектуры включают все, что машина предоставляет пользователю для организации эффективной, быстрой и удобной работы.

Достижения в микроэлектронике позволили уменьшить габариты вычислительных машин. С 1964 г. началось распространение настольных клавишных электронных вычислительных машин. Выпускаемые теперь нашей промышленностью микрокалькуляторы можно разделить на три типа: микрокалькуляторы, выполняющие только четыре арифметические операции — «Электроника БЗ-04», «Электроника БЗ-14 М» и др.; микрокалькуляторы инженерного типа, выполняющие арифметические операции, вычисляющие значения элементарных функций — «Электроника БЗ-18», «Электроника БЗ-36», «Электроника МКШ-2» и др.; микрокалькуляторы, выполняющие все операции первых двух типов, а также работающие в режиме «Программирование», так называемые программно-управляемые микрокалькуляторы — «Электроника БЗ-21», «Электроника БЗ-34», «Электроника МК-54», «Электроника МК-56», «Электроника МК-61» и др.

Алгоритмы и их схемы

Понятие алгоритма. Для решения задачи исполнителю необходимо указать последовательность действий, которые он должен выполнить для достижения цели — получения результата. Иначе говоря, испольнителю должен быть указан алгоритм решения задачи, представленный на понятном ему языке. Под исполнителем подразумевается как человек, так и вычислительная машина.

Алгоритм решения задачи — это конечная последовательность

четко сформулированных правил решения некоторого класса задач.

В обыденной жизни мы постоянно сталкиваемся с алгоритмами как последовательностью действий, приводящих к достижению поставленной цели,— это правила перехода улицы, рецепты приготовления различных блюд, поиск нужного слова в словаре и др. Алгоритмы в математике — это правила нахождения корней квадратных алгебраических уравнений, правила выполнения арифметических действий, правила нахождения наибольшего общего делителя и наименьшего общего кратного, разложение числа на простые множители и др.

Алгоритмы должны обладать целым рядом свойств: дискретностью, точностью, понятностью, результативностью, массовостью.

Дискретность — это разбиение алгоритма на ряд отдельных законченных действий — шагов. *Точность* — это указание последовательности шагов. *Понятность* — это однозначное понимание и исполнение каждого шага алгоритма его исполнителем. *Результативность* — обязательное получение результата за конечное число шагов. *Массовость* — применимость алгоритма к решению целого класса однотипных задач.

Если бы мы знали алгоритмы решения всех задач, то их исполнение можно было бы поручить машине. Но оказалось, что не все задачи, которые нам хотелось бы решить, имеют алгоритмы решения. Задачи, в принципе не имеющие общего решения, называют алгоритмически неразрешимыми. К примеру, мы знаем, как решить любое квадратное алгебраическое уравнение, пользуясь одним и тем же алгоритмом. Похожие формулы существуют и для кубических уравнений и для уравнений четвертой степени. Но уже для уравнений степени выше четвертой таких формул нет и в принципе быть не может, хотя в частном случае отдельные уравнения можно решить. Есть и другие алгоритмически неразрешимые задачи, например задача о трисекции угла, о квадратуре круга и др.

Существуют различные способы представления алгоритмов. В пособии рассматривается представление алгоритма в виде схем, на языке программируемого микрокалькулятора и на языке программирования БЕЙСИК.

Схема алгоритма. Схема алгоритма представляет собой графическое изображение алгоритма с помощью определенным образом связанных между собой блоков. Под блоком понимается любой конечный этап вычислительного процесса, принимаемый в данной схеме как целое. Типы блоков представлены на рисунке 2. Внутри каждой геометрической фигуры дается описание операций, содержащихся в данном блоке. При этом используются символы математических операций и операций отношений.

Функциональный блок (рис. 2, *а*) используется для представления функций. В этом блоке осуществляется преобразование информации по заданному действию *S*. *Блок проверки условия*

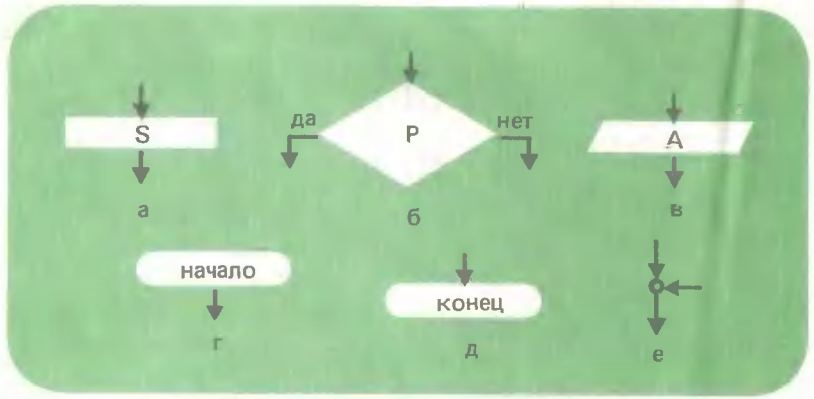


Рис. 2

(рис. 2, *б*) используется для управления преобразованием информации. В результате анализа условия *P* выбирается одно из двух возможных направлений «да» или «нет», и управление передается блоку, записанному на выбранном направлении. *Информационный блок* (рис. 2, *в*) используется для обозначения операций обмена информации между устройствами. Блоки, изображенные на рисунках 2, *г* и *д*, называют соответственно *начальным* и *конечным*. *Объединяющий блок* (рис. 2, *е*) указывает на передачу управления от входящих стрелок к одной выходящей. Для изображения направления потока управления используются соединительные линии со стрелкой. Алгоритм решения задачи, представленный схемой, начинается в начальном блоке и заканчивается в конечном.

Базовые структуры. При изображении алгоритмов с помощью схем используются базовые управляющие структуры: СЛЕДОВАНИЕ, РАЗВИЛКА, ПОВТОРЕНИЕ.

СЛЕДОВАНИЕ: структура (рис. 3) означает, что действия S_1 и S_2 должны быть исполнены одно за другим.

РАЗВИЛКА: действие, определяемое структурой, осуществляет анализ условия *P* (истинно или ложно) и альтернативный выбор дальнейшего направления в последовательности выполнения действий в зависимости от значения *P*. Различают ПОЛНУЮ РАЗВИЛКУ (рис. 4) и НЕПОЛНУЮ РАЗВИЛКУ (рис. 5). Словесно ПОЛНАЯ РАЗВИЛКА описывается так: если *P* истинно, то исполнить *S*, иначе *T*. НЕПОЛНУЮ РАЗВИЛКУ словесно можно описать так: если *P* истинно, то исполнять *S*.

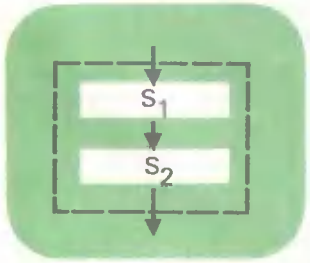


Рис. 3

ПОВТОРЕНИЕ: структура описывает циклические вычислительные процессы. Структуру, изображенную на рисунке 6, словесно можно выразить следующим образом: пока *P* истинно, исполнять *S*. Эту структуру называют цикл-ПОКА.

Структуру, изображенную на рис. 7, называют цикл-ДО. Словесно ее можно описать так: исполнять *S* до истинности *P*.

Базовые структуры имеют один вход и один выход.

Схема алгоритма, представленная базовыми структурами, называется *структурированной*.

Пять вышеуказанных структур могут комбинироваться одна с другой, как того требует алгоритм.

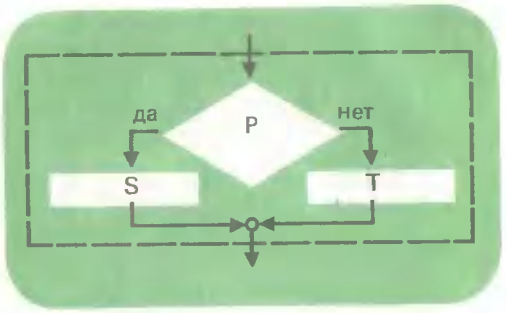


Рис. 4

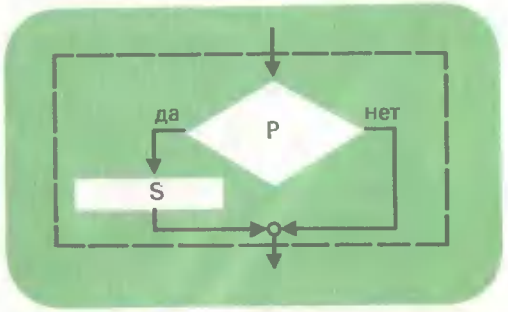


Рис. 5

Конструирование схем алгоритмов. Рассмотрим различные алгоритмы, встречающиеся при решении задач.

Задача 1. Составить схему алгоритма вычисления расстояния между двумя точками, заданными координатами $A(x_1, y_1)$, $B(x_2, y_2)$.

Решение. Расстояние между двумя точками вычисляется по формуле

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}. \quad (1)$$

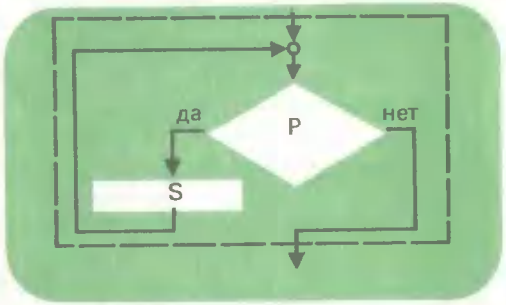


Рис. 6

Какими бы действительными числами ни были заданы координаты точек, по формуле (1) можно найти значение *d*. Последовательность вычисления значения *d* можно описать в виде изображенной на рисунке 8 схемы алгоритма. При изображении схемы все блоки будем нумеровать. Блок *1* указывает на необходимость определения исходных величин перед исполнением алгоритма. В блоках *2—4* вычисляются значения по формулам, записанным справа от знака «:=». После этого численное значение присваивается переменной,

записанной слева от этого знака. Знак «:=» означает «присвоить». Результат вычисления нужно тем или иным способом зафиксировать.

Задача 2. Составить схему алгоритма нахождения суммы дробей $\frac{d}{b}$ и $\frac{c}{d}$.

Решение. Сумма вычисляется по формуле $S=\frac{ad+bc}{bd}$.

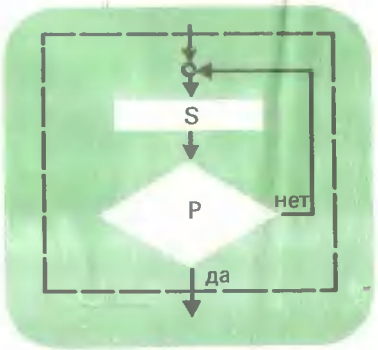


Рис. 7

Поскольку b и d предполагаются отличными от нуля, то схема алгоритма будет содержать только функциональные блоки. Приведенная на рисунке 9 схема алгоритма содержит базовые структуры СЛЕДОВАНИЕ.

Алгоритмы решения задач 1 и 2 называют *линейными*.

Задача 3. Составить схему алгоритма вычисления значения функции $y=\frac{x+7}{x^2-5x+6}$, где x задано.

Решение. При некоторых значениях x знаменатель принимает значение 0. Поэтому, прежде чем находить y, нужно вычислить знаменатель. Если знаменатель отличен от нуля, то значение y существует, иначе — значение y не определено. Схема алгоритма изображена на рисунке 10.

Рис. 8

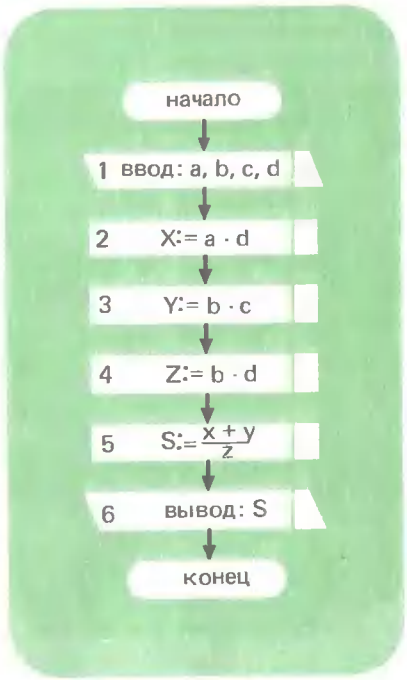


Рис. 9

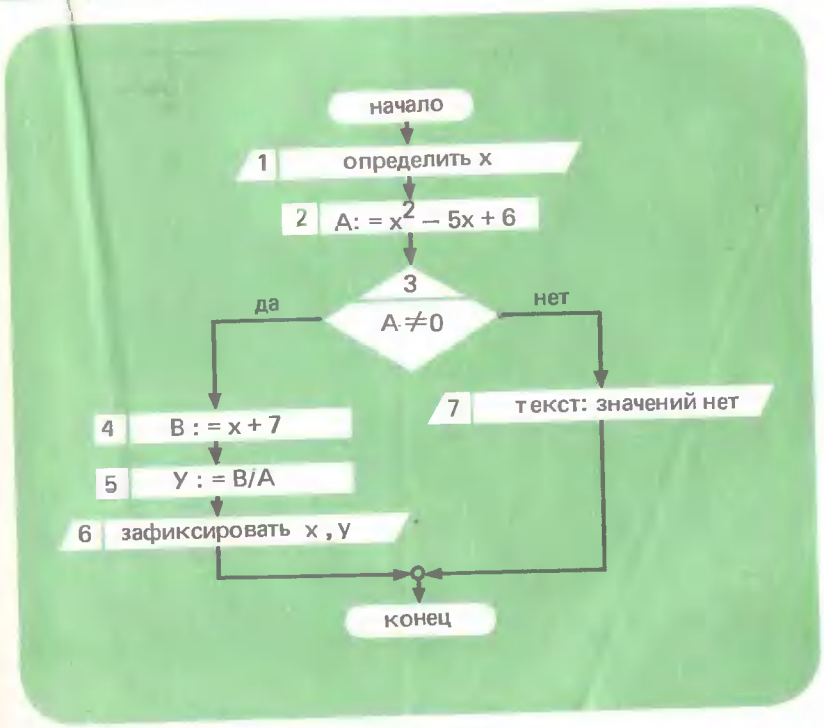


Рис. 10

Задача 4. Заданы два числа A и B. Составить схему алгоритма нахождения и присваивания значения большего из них переменной x. В случае равенства чисел присвоить переменной x значение A.

Решение. Схема алгоритма решения задачи изображена на рисунке 11.

Задача 5. Заданы три попарно неравных числа A, B и C. Расположить числа в порядке возрастания.

Решение. Из трех элементов (букв) A, B и C можно образовать шесть различных перестановок: ABC, ACB, BAC, BCA, CAB, CBA. Учитывая, что каждой из букв A, B и C присвоено числовое значение, в зависимости от исходных значений A, B и C одна из шести указанных перестановок соответствует решению задачи. По сути дела решение задачи состоит в определении нужной перестановки. Чтобы найти последовательность, в которой три числа располагаются в порядке возрастания, достаточно трех

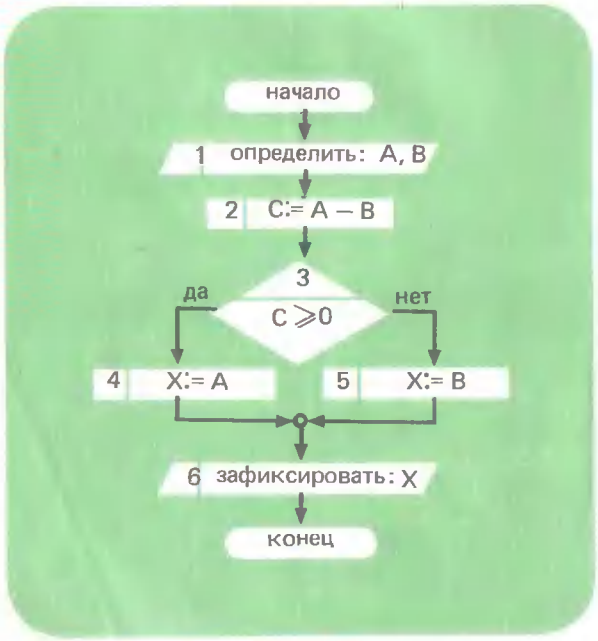


Рис. 11

отношений. Схема алгоритма решения задачи изображена на рисунке 12.

Если алгоритм содержит базовую структуру РАЗВИЛКА, то его называют *ветвящимся*.

Алгоритмы решения задач 6—12 будут содержать базовые структуры цикл-ДО или цикл-ПОКА. Такие алгоритмы называют *циклическими*.

Задача 6. Составить схему алгоритма нахождения суммы первых m членов последовательности, которые определяются по формуле $x_n = \frac{4n^2}{15n-6}$, $n=1, 2, ..., m$.

Решение. Необходимо просуммировать $x_1, x_2, ..., x_m$. Сумму обозначим буквой S. Чтобы получить сумму, можно поступить следующим образом: вычислить x_1 и положить $S := x_1$, а затем вычислить x_2 и положить $S := S + x_2$ и т. д. до тех пор, пока $n \leqslant m$.

Заметим, что, взяв первоначальную сумму S равной 0 и подсчитав x_1, можно записывать $S := S + x_1$. На рисунке 13 дана схема алгоритма решения задачи. При каждом изменении значения переменной n переменной x присваивается очередное значение.

Задача 7. Составить схему алгоритма вычисления значений функции

$$y = \begin{cases} 2+7x, & \text{если } x<0, \\ x-2, & \text{если } 0 \leqslant x \leqslant 2, \\ x, & \text{если } x>2 \end{cases}$$

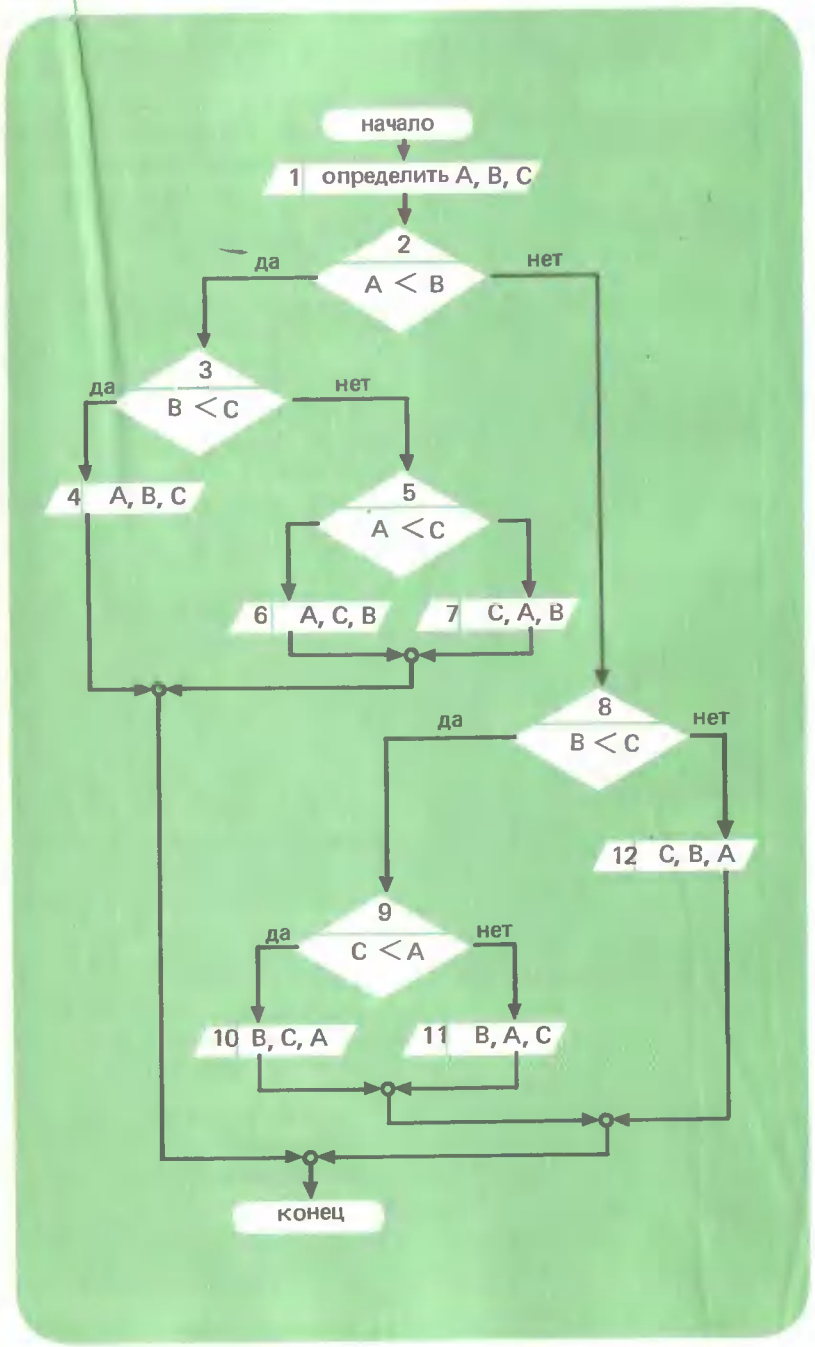


Рис. 12

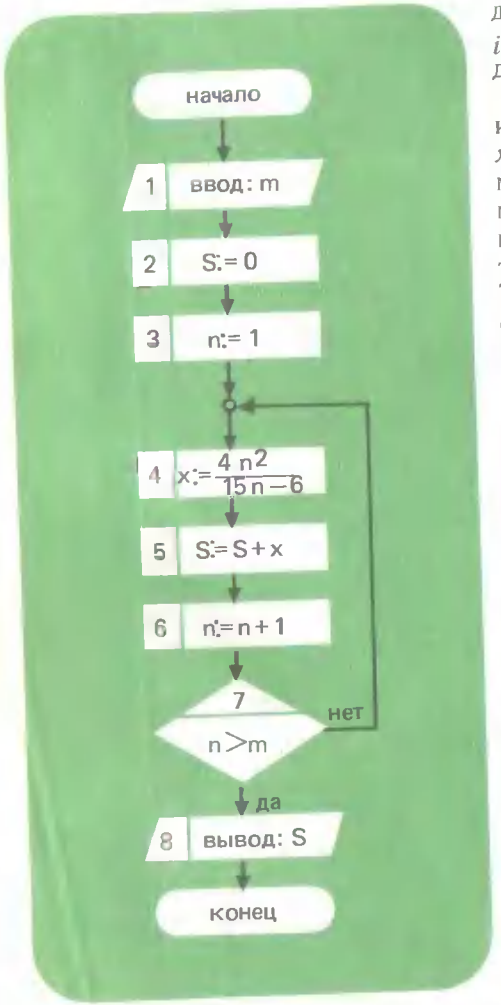


Рис. 13

для $x=x_i$, где $x_i=x_0+i\cdot h$, $i=\overline{1,n}$. Значения n, x_0, h заданы.

Решение. В качестве исходной информации заданы: x_0 — начальное значение аргумента, h — шаг изменения аргумента, n — количество значений функции. Нужно составить таблицу значений функции y. Таблица будет содержать три столбца (номер строки, значение аргумента, значение функции) и n строк.

Изменение значений переменной x связано с изменением параметра i. Изменение i от 1 до n с шагом 1 влечет за собой изменение x от x_1 до x_n с шагом h. Вычисления значений функции выполняются до тех пор, пока параметр не станет больше заданного значения n. В зависимости от того, какому из промежутков $x<0$, $0\leqslant x\leqslant 2$ или $x>2$ принадлежит значение x, вычисление y выполняется по одной из формул: $y:=2+7x$, $y:=x-2$, $y:=x$. После вычисления y фиксируются значения параметра i, аргумента x и функции y. Схема алгоритма вычисления значений функции изображена на рис. 14.

Задача 8. Составить схему алгоритма нахождения наибольшего общего делителя (НОД) двух натуральных чисел N и M.

Решение. При разработке алгоритмов, как правило, исходные данные сохраняют (в этом случае переменные M и N). В связи с этим в схеме (рис. 15) используются переменные x и y. После выполнения алгоритма значения переменных N и M сохранятся.

Задача 9. Составить схему алгоритма вычисления значения суммы $y=1-\frac{1}{2}+\frac{1}{3}-\frac{1}{4}+\frac{1}{5}-\frac{1}{6}+\ldots$. Суммирование продолжать до тех пор, пока очередное слагаемое по модулю не станет меньше заданного числа h — погрешности суммирования.

Решение. Решение задачи напоминает решение задачи 6. Здесь также нужно находить сумму. Поэтому начальное значение суммы положим равным нулю, т. е. $y:=0$. В отличие от задачи 6 здесь неизвестно число слагаемых. Окончание вычислений определяется отношением $|a|\geqslant h$, где a — слагаемое. В алгоритмах решения задач с незаданным числом повторения цикла будем

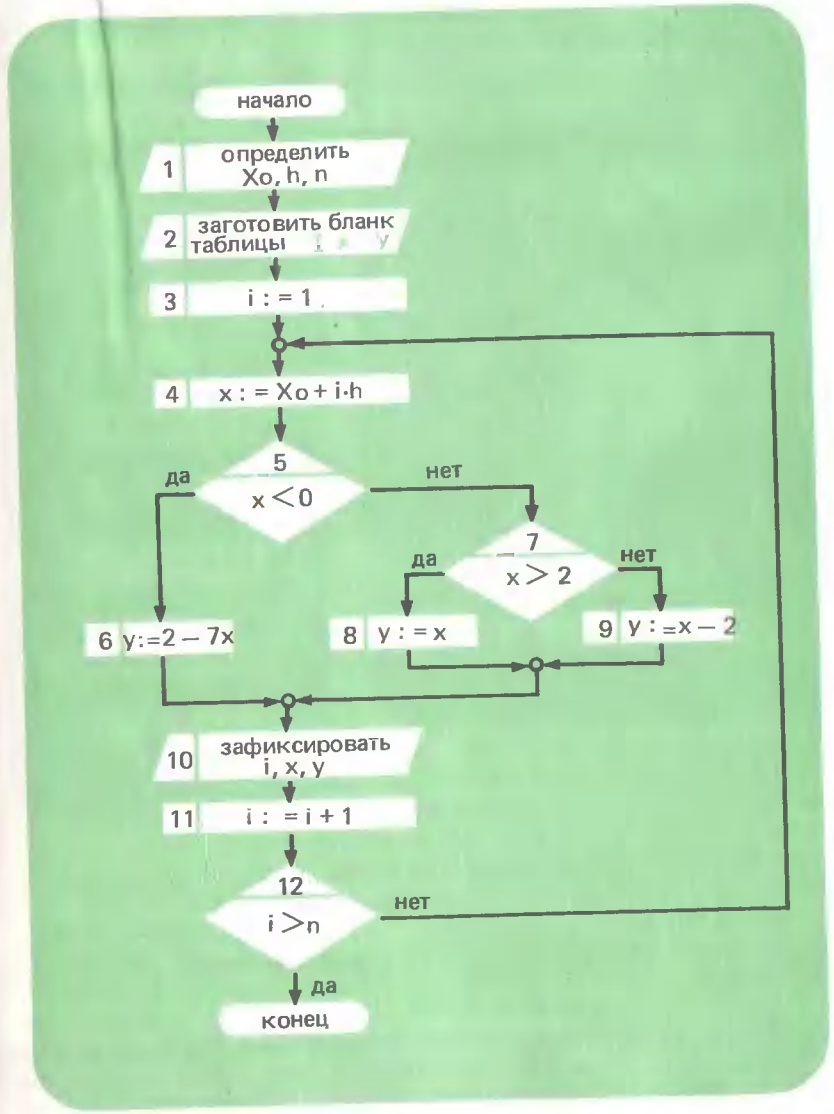


Рис. 14

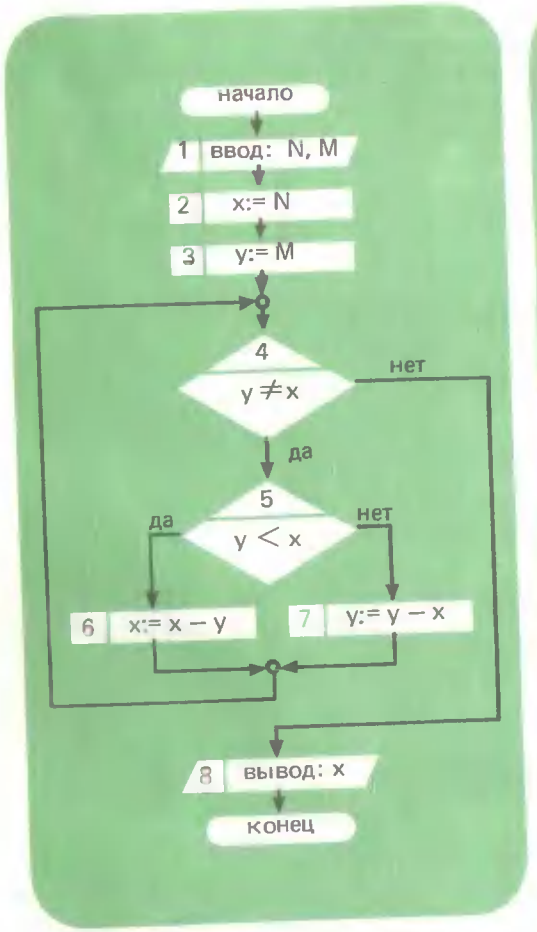


Рис. 15

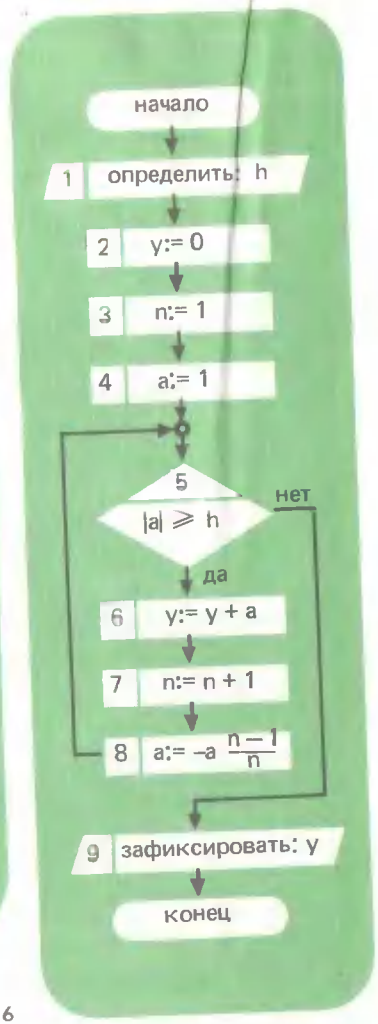


Рис. 16

использовать базовую структуру цикл-ПОКА. На рисунке 16 изображена схема алгоритма решения задачи.

Задача 10. Задано множество A, содержащее n членов. Составить схему алгоритма вычисления суммы квадратов этих членов (элементов).

Решение. Обозначенное именем (например, буквой) множество элементов, имеющих одинаковые свойства, называется массивом (или таблицей, или списком). Элементы массива задаются именем массива и индексом — порядковым номером следования элемента в массиве. Например, задан массив $A(3; 8; 6; 8; 9; 5)$. Если $i:=2$, то A_i определяет число 8, стоящее на вто-

ром месте в массиве. Если $i:=5$, то определен элемент $A_i=9$.

В рассматриваемой задаче задан массив, содержащий n элементов $A(A_1, A_2, ..., A_n)$. Приведем два варианта решения задачи 10 (рис. 17 и 18).

Задача 11. Задан массив $A(A_1, A_2, ..., A_n)$, содержащий n элементов. Составить схему алгоритма расположения элементов массива в порядке возрастания.

Решение. Одним из простых методов является последовательный выбор меньшего из элементов и расположение его на крайнюю левую позицию в ряду рассматриваемых элементов.

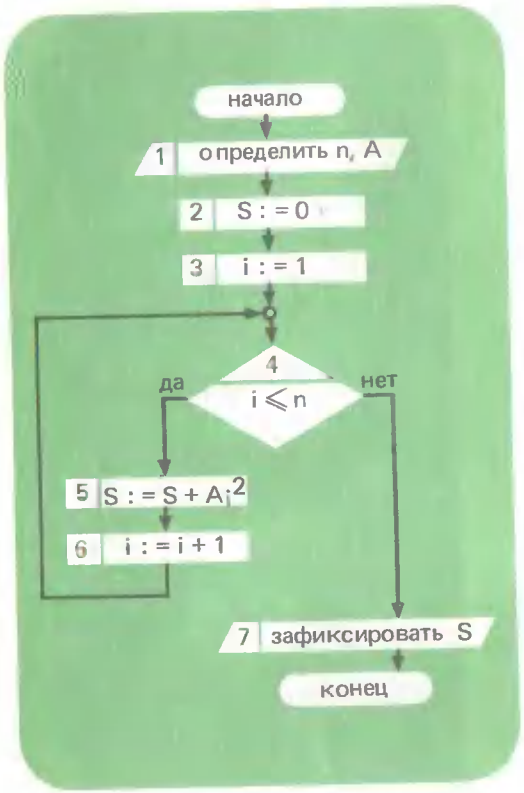


Рис. 17

Выбор наименьшего элемента начинается со сравнения двух первых. Наименьший помещается на первое место в массиве. Элемент, расположенный на первом месте, сравнивается с третьим элементом. После их сравнения наименьший элемент помещается на первое место в массиве. Процесс сравнения элемента, расположенного на первом месте, со всеми остальными элементами массива и расположение наименьшего из каждой пары на первое место продолжается до тех пор, пока не будет рассмотрен последний элемент массива. После этого переходят ко второму элементу массива и определяют наименьший элемент из оставшихся. Для поиска такого элемента повторяем те же операции. Такой процесс продолжается до тех пор, пока не останется последний элемент, являющийся наибольшим элементом массива.

При алгоритмизации задачи вместо понятия «место» будем говорить «ячейка». В каждую ячейку записан один элемент массива. Местоположение элемента в массиве определяется номером ячейки. Первый элемент занимает первую ячейку, второй — вторую и т. д.

Содержимым ячейки можно воспользоваться любое число раз, пока в ячейку не будет записано новое содержимое. Отсюда ясно, что простой обмен содержимым двух ячеек приведет к потере

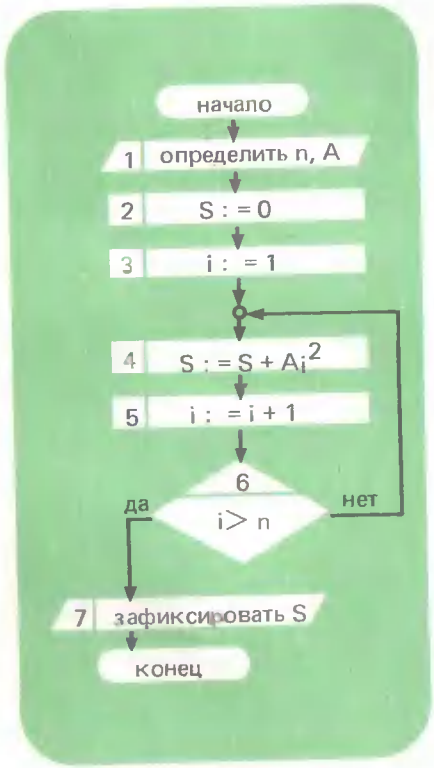


Рис. 18

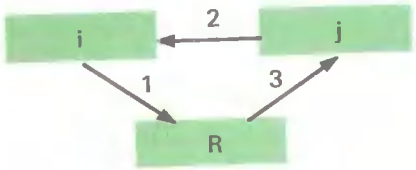


Рис. 19

одного числа. Чтобы этого не произошло, при обмене содержимым нужно использовать третью, свободную ячейку. Тогда обмен содержимым между двумя ячейками можно осуществить, например, в последовательности, показанной на рисунке 19. Буквы i, j, R — номера ячеек. Цифры на стрелках указывают последовательность перемещения элементов по ячейкам. Такое перемещение легко выполнить следующим образом:

$B_R := A_i$; $A_i := A_j$; $A_j := B_R$.

Через A_i обозначен элемент массива A_i, находящийся в ячейке с номером i, A_j — элемент того же массива из ячейки с номером j. Выбор наименьшего из двух элементов производится с помощью отношения $A_i > A_j$.

Пусть i — номер ячейки, в которую нужно поместить наименьшее из оставшихся чисел. Содержимое всех оставшихся ячеек с номерами $i+1$, $i+2$, ..., n последовательно сравниваем с содержимым i-й ячейки. Наименьшее число из пары помещается в эту ячейку. В результате $n-1$ сравнений выявим наименьшее из чисел. Отсюда ясно, что индекс j должен изменяться от $i+1$ до n. Отношение $j > n$ определяет окончание процесса сравнения элемента A_i с оставшимися элементами. Процесс упорядочения элементов продолжается до тех пор, пока не останется один элемент. Окончание процесса определяется истинностью отношения $i > n-1$. Иначе говоря, на первом шаге находят наименьший элемент массива $A(A_1, A_2, ..., A_n)$ и помещают в ячейку с номером 1, т. е. $A_1 = \min\limits_{i=1,n}\{A_i\}$. На втором шаге находят наименьший элемент массива $(A_2, A_3, ..., A_n)$ и помещают в ячейку с номером 2, т. е. $A_2 = \min\limits_{i=2,n}\{A_i\}$ и т. д. На последнем $n-1$ шаге рассматривается массив (A_{n-1}, A_n), и его

элементы упорядочиваются по возрастанию. Схема алгоритма решения задачи изображена на рисунке 20.

Задачу 11 можно решить иначе. В ней можно выделить две подзадачи: 1) найти наименьший элемент массива, 2) расположить заданный наименьший элемент на первое место в массиве.

При таком подходе алгоритм решения подзадачи 2 будет *головным*. Чтобы оперировать с наименьшим элементом, его нужно найти, задать. Это можно сделать, используя алгоритм решения подзадачи 1. Такой алгоритм называют *вспомогательным*.

Сформулируем и решим каждую из подзадач.

1. Составить схему алгоритма определения порядкового номера наименьшего элемента одномерного массива A, содержащего элементы, занумерованные от m до n включительно, $m < n$.

2. Составить схему алгоритма упорядочения n элементов массива A в порядке возрастания, используя алгоритм решения подзадачи 1 в качестве вспомогательного.

Схемы алгоритмов решения подзадач 1 и 2 представлены на рисунках 21 и 22 соответственно.

Чтобы воспользоваться вспомогательным алгоритмом, нужно указать его имя и задать исходные значения: начальный и конечный номера массива, значения элементов массива и переменную, которой в результате выполнения вспомогательного алгоритма будет присвоен номер наименьшего элемента массива. Наименьший элемент ставится на первое место в рассматриваемом массиве, но этот массив после каждого шага

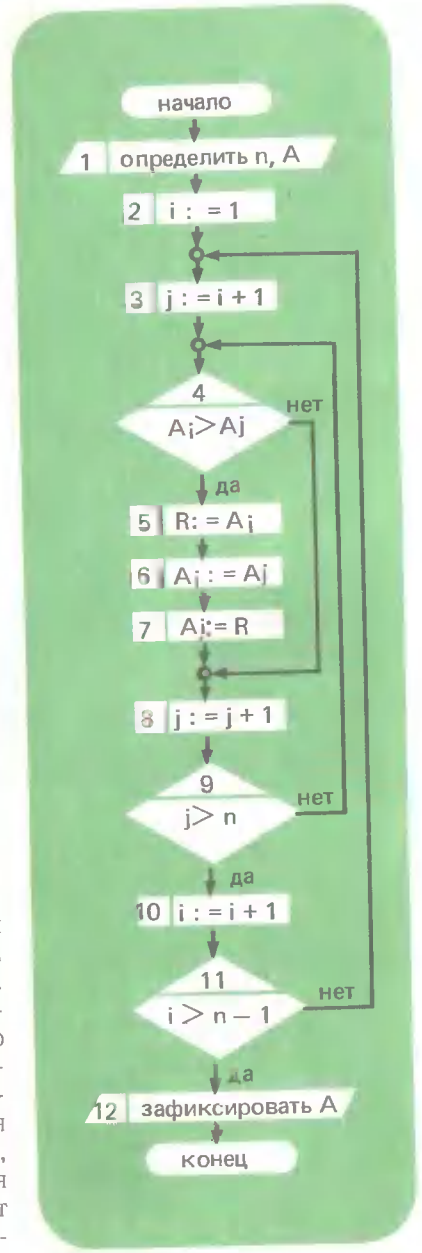


Рис. 20

(цикла) уменьшается на один элемент. Поэтому номер первого элемента будет принимать разные значения, равные значению переменной k. ПРи обращении к вспомогательному алгоритму, а это осуществляется при исполнении блока 4 головного алгоритма (рис. 22), переменным m и n вспомогательного алгоритма присваиваются значения соответственно k и n, определенные в головном алгоритме. Элементам массива, используемого во вспомогательном алгоритме, присваиваются значения элементов массива A, заданных в головном алгоритме и занумерованных от k до n включительно. Переменная l получает значение номера наименьшего элемента массива, рассматриваемого во вспомогательном алгоритме.

Рассмотрим третий вариант решения задачи 1.1. Выделим две подзадачи.

1′. Составить схему алгоритма определения порядкового но-

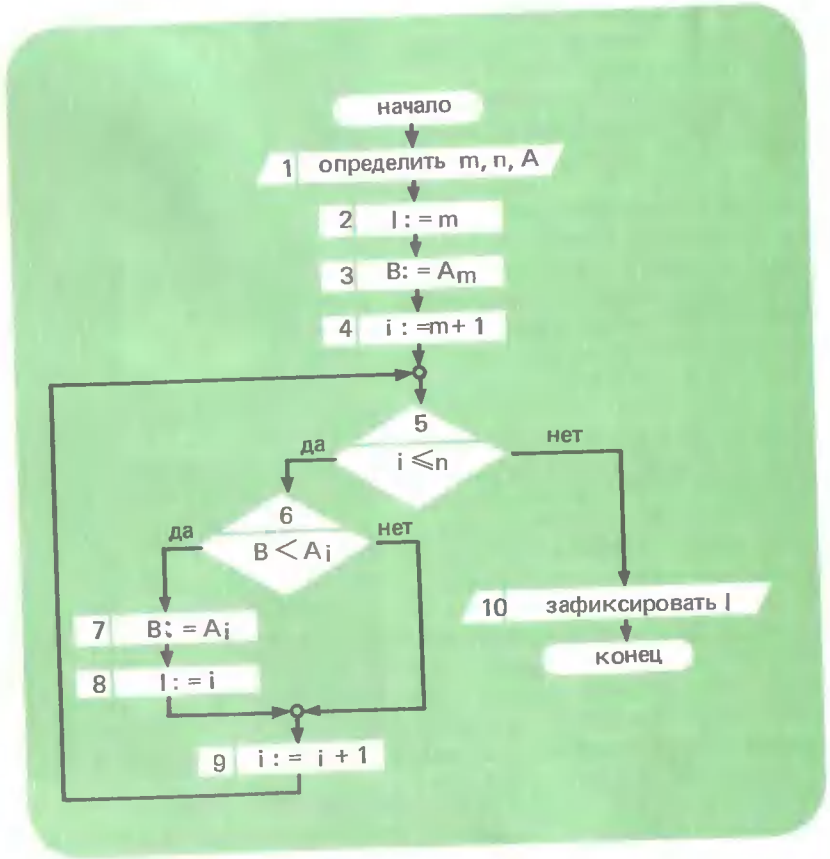


Рис. 21

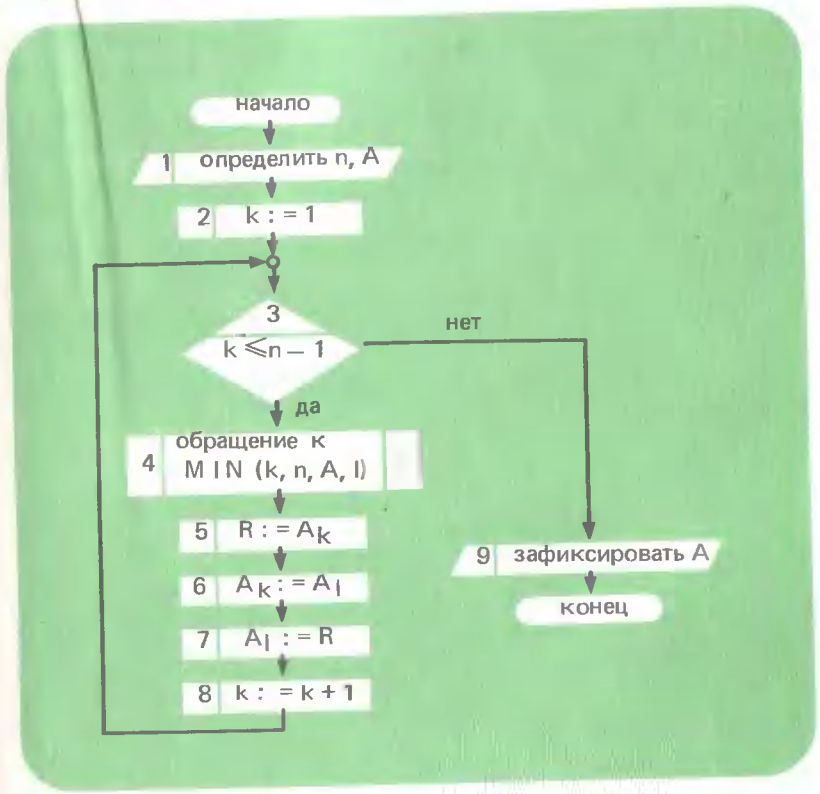


Рис. 22

мера наибольшего элемента одномерного массива A, содержащего элементы, занумерованные от m до n включительно ($m<n$).

2′. Составить схему алгоритма расположения n элементов массива A в порядке возрастания, используя алгоритм решения подзадачи 1′ в качестве вспомогательного.

При решении подзадачи 2′ на первом шаге наибольший элемент массива $A(A_1, A_2, ..., A_n)$ помещается в ячейку с номером n, т. е. $A_n = \max\limits_{i=\overline{1,n}}\{A_i\}$. На втором шаге наибольший элемент массива $(A_1, A_2, ..., A_{n-1})$ помещается в ячейку с номером $n-1$, т. е. $A_{n-1} = \max\limits_{i=\overline{1,n-1}}\{A_i\}$ и т. д.

Схемы алгоритмов решения подзадач 1′ и 2′ представлены на рисунках 23 и 24 соответственно.

Задача 12. Заданы значение B и одномерный массив $A(A_1, A_2, ..., A_n)$, содержащий n элементов. Составить схему алгоритма определения первого элемента массива A меньшего B. Если такой элемент есть, присвоить его значение переменной C.

Решение. Интересующий нас элемент найден, если текущее значение A_i меньше значения B. Тогда, положив $C := A_i$, процесс нужно завершить.

Если текущее значение A_i не меньше значения B, то возможны два исхода: все элементы массива A уже просмотрены, среди них нет меньших B, процесс нужно завершить; не все элементы массива A просмотрены, нужно увеличить значение параметра (индекса) i и переходить к сравнению значения очередного элемента A_i со значением B. Схема алгоритма решения приведена на рисунке 25. Она не является структурированной. Исправить положение можно введением дополнительной переменной k. На рисунке 26 представлена структурированная схема алгоритма решения этой задачи.

Для самостоятельной работы предлагается решить следующие задачи.

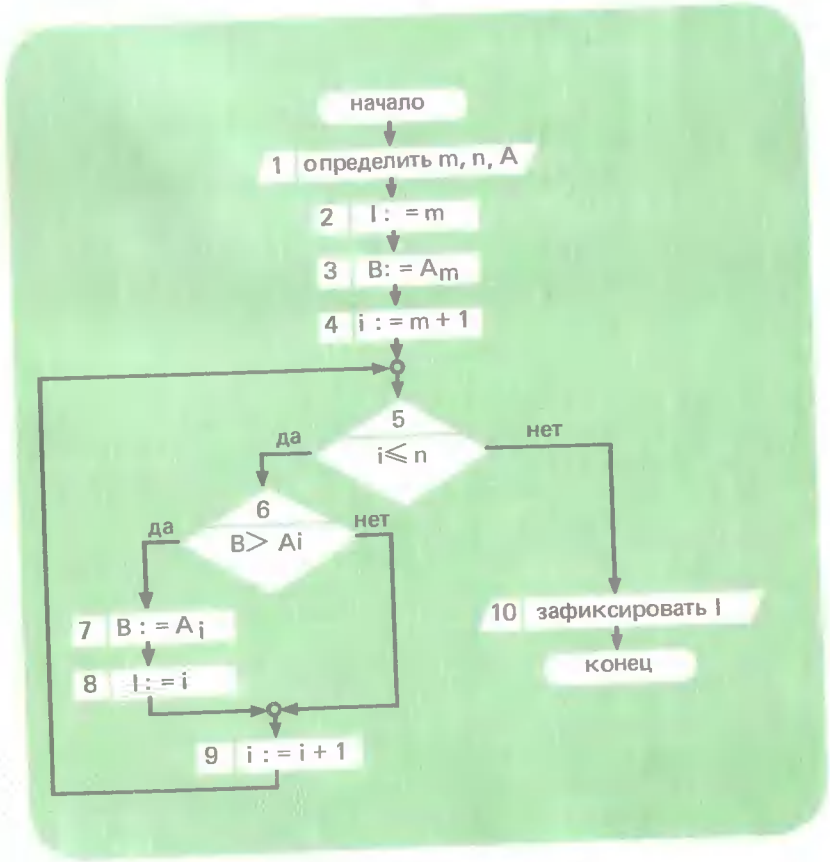


Рис. 23

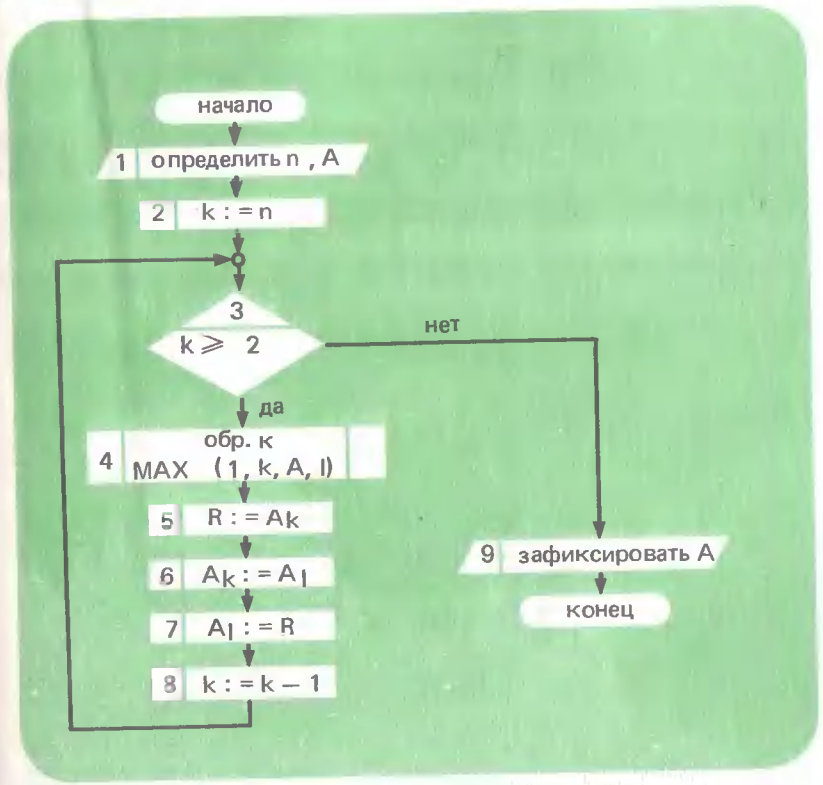


Рис. 24

1. Три вершины $A(x_1;\ y_1)$, $B(x_2;\ y_2)$ и $C(x_3;\ xy_3)$ треугольника заданы своими координатами. Составить схему алгоритма вычисления площади треугольника.

2. Заданы координаты точек $A(x_1;\ y_1)$ и $O(x_0;\ y_0)$. Составить схему алгоритма вычисления координат точки B, симметричной точке A относительно O.

3. Составить схему алгоритма вычисления площади полной поверхности конуса, если известны длина образующей и величина угла наклона ее к плоскости основания.

4. Треугольник задан координатами вершин $A(x_1;\ y_1;\ z_1)$, $B(x_2;\ y_2;\ z_2)$ и $C(x_3;\ y_3;\ z_3)$. Составить схему алгоритма вычисления длины медианы AA_1.

5. Заданы три числа a, b и c. Составить схему алгоритма определения наибольшего из трех заданных чисел.

6. Составить схему алгоритма исследования числа решений системы уравнений

$$\begin{cases} y = a_1x + b_1, \\ y = a_2x + b_2. \end{cases}$$

7. Составить схему алгоритма решения уравнения $ax=b$ относительно x.

8. Составить схему алгоритма решения неравенства $ax>b$ относительно x.

9. Составить схему алгоритма вычисления n первых членов арифметической прогрессии, если заданы первый член прогрессии и ее разность.

10. Составить схему алгоритма вычисления $n!=1\cdot 2\cdot 3\cdot \ldots \cdot n$, где n задано.

11. Составить схему алгоритма вычисления значения функции с точностью h:

а) $y=1-\frac{x^2}{2!}+\frac{x^4}{4!}-\frac{x^6}{6!}+\ldots;$

б) $y=x-\frac{x^3}{3!}+\frac{x^5}{5!}-\frac{x^7}{7!}+\ldots;$

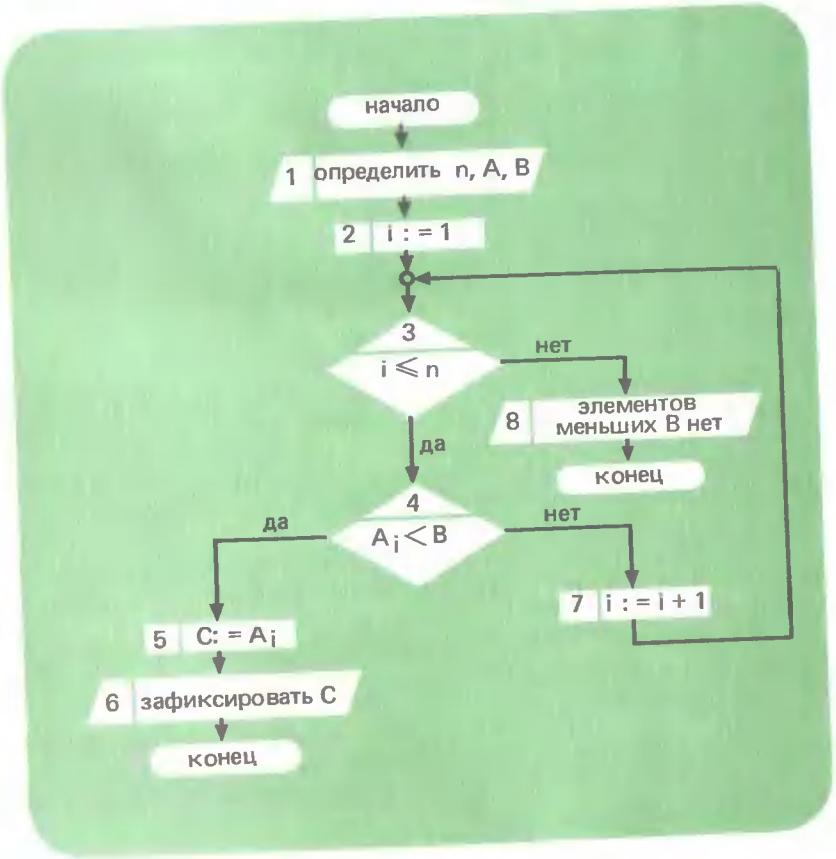


Рис. 25

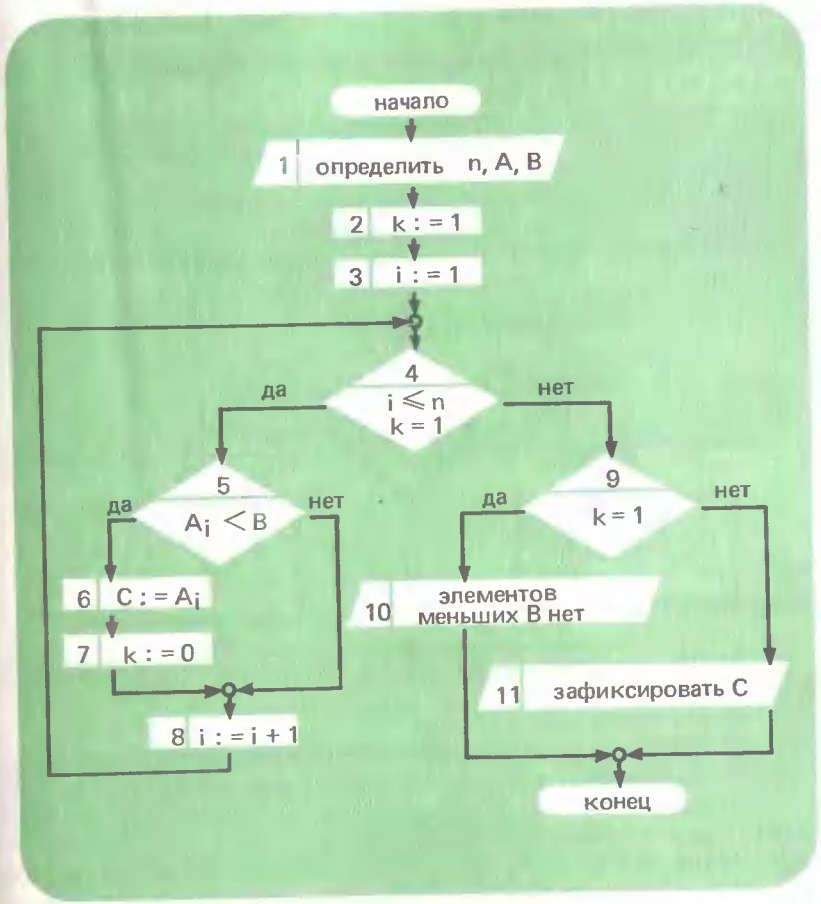


Рис. 26

в) $y=\frac{x}{2}-\frac{x^2}{5}+\frac{x^3}{8}-\frac{x^4}{11}+\ldots,\ |x|<1,$

т. е. суммирование продолжать до тех пор, пока очередное слагаемое a удовлетворяет условию $|a|\geqslant h$.

12. Составить схему алгоритма вычисления значения функций:

а) $y_i=x_ie^{x_i^2}+x_i^2;$

б) $f(x_i)=\frac{x_i^2+2x_i-3}{0{,}5x_i^5+2{,}25_i^2-12{,}5x_i+12},$

где $x_i=1+i\cdot 0{,}1$, $i=0, 1, 2, \ldots, 20$. Результат вычислений поместить в таблице:

I	X	Y

13. Составить схему алгоритма вычисления значений функции

$$y=\begin{cases} a\,lg\,|\sin x-\pi|, & \text{если } |ax|\leqslant 1,\\ \sqrt[3]{a\ln|\pi-x|}, & \text{если } ax>1,\\ 3x^2-6x-a, & \text{если } ax<-1\end{cases}$$

для $x=x_i$, где $x_i=x_0+i\cdot h$, $i=\overline{1,n}$, переменная a при каждом значении x_i принимает последовательно все значения из множества $\{a_1, a_2, \ldots a_n\}$. Значения x_0, h, k, n, a_1, ..., a_n заданы.

14. Составить схемы алгоритмов определения принадлежности точки $X(x_1; y_1)$ заштрихованной области (рис. 27).

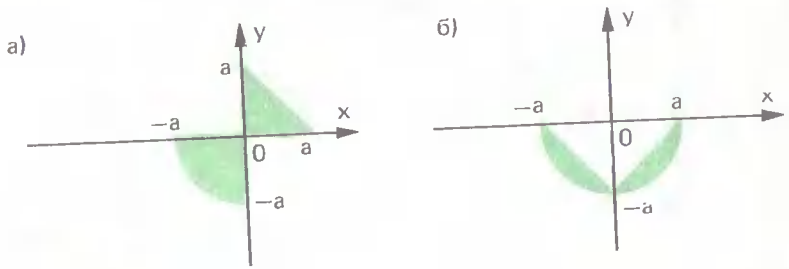


Рис. 27

15. Задан массив $A(A_1, A_2, \ldots, A_n)$. Составить схему алгоритма вычисления:

а) суммы элементов массива;

б) суммы элементов, стоящих на четных местах в массиве;

в) наибольшего элемента массива и его номер в массиве;

г) суммы отрицательных элементов массива и количества положительных элементов.

16. Задан массив $A(A_1, A_2, \ldots, A_n)$. Составить схему алгоритма:

а) определения наибольшего элемента массива и его замену местами с последним элементом;

б) расположения элементов массива в порядке убывания.

17. Задана квадратная матрица

$$A=\begin{pmatrix} A_{11}\,A_{12}\ldots A_{1n}\\ \cdot\;\cdot\;\cdot\;\cdot\;\cdot\;\cdot\;\cdot\;\cdot\\ A_{n1}\,A_{n2}\ldots A_{nn}\end{pmatrix}.$$

Составить схему алгоритма вычисления:

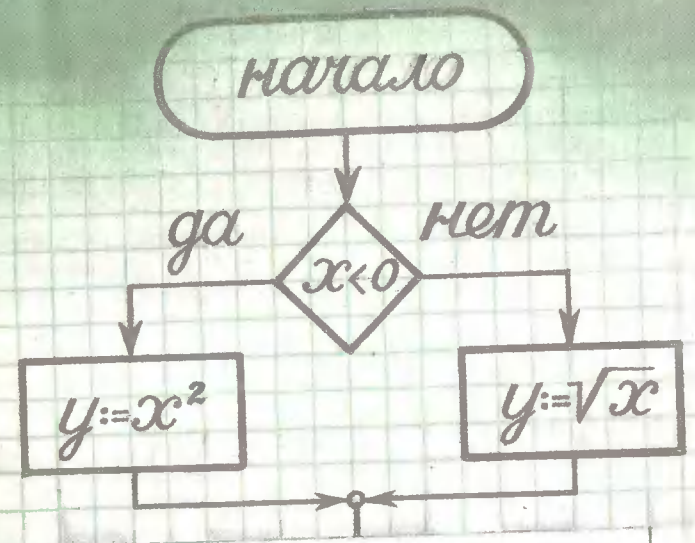
а) суммы элементов, расположенных на главной диагонали;

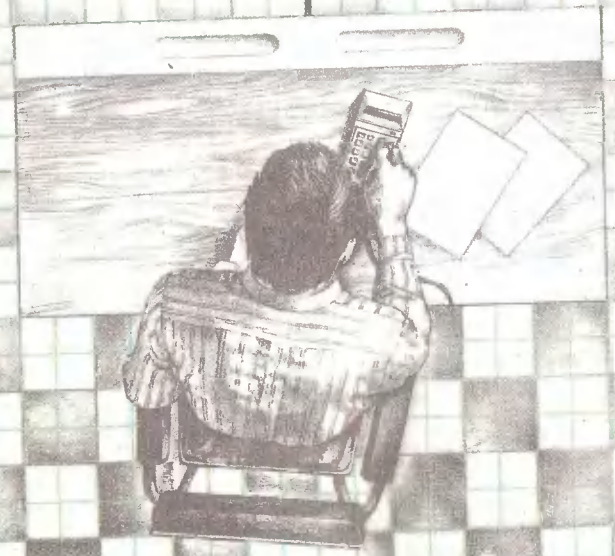
б) суммы элементов, расположенных на диагонали, перпендикулярной главной диагонали;

в) суммы элементов, расположенных под главной диагональю;

г) наибольшего элемента и его координат.

Элементы программирования на микрокалькуляторах





1. Ввод и представление чисел. Выполнение арифметических операций

Внешний вид микрокалькулятора МК-61 показан на первом форзаце книги, на рисунке 28 изображена его верхняя панель.

Переключатель *1* служит для включения микрокалькулятора. На индикаторе *2* высвечиваются числа. Если переключатель *1* поставить в положение «ВКЛ», то на индикаторе высветится 0. Индикатор представляет собой 12-разрядное табло, на котором высвечиваются числа, причем каждая цифра занимает отдельный разряд, а десятичная запятая высвечивается в том же разряде, что и последняя цифра целой части числа.

Переключатель *3* используется при вычислении тригонометрических функций в градусах (Г), градах (ГРД) и радианах (Р). С помощью клавиатуры производится управление работой микрокалькулятора.

Микрокалькулятор работает с восьмиразрядными числами, которые могут быть представлены как в естественной форме записи, так и в форме с плавающей запятой (стандартная запись). Для ввода чисел служат клавиши [1], [2], [3], [4], [5], [6], [7], [8], [9], [0], [·] [/—/], [ВП]. При вводе чисел клавиши с обозначениями цифр и десятичной запятой нажимаются в том порядке, в котором читается число. Для положительных чисел знак «+» не вводится. Для отрицательных чисел знак «—» вводится после набора самого числа нажатием клавиши [/—/]. Для снятия чисел с индикатора служит клавиша [c_x] (сброс с индикатора).

Рассмотрим примеры ввода чисел в естественной форме.

Пример 1. Ввести числа 27,4153; —149,2534.

Чтобы ввести число 27,4153, нужно последовательно нажать клавиши [2][7][·][4][1][5][3]. Чтобы очистить индикатор, необходимо нажать на [c_x]. Для ввода числа —149,2534 нажимаем следующую последовательность клавиш: [1][4][9][·][2][5][3][4][/—/]. На индикаторе получаем

—	1	4	9.	2	5	3	4				

Микрокалькулятор «Электроника МК-61» оперирует также с числами, представленными в форме с плавающей запятой: вида $M \cdot 10^p$, где M — мантисса числа, p — порядок числа.

При вводе чисел в форме с плавающей запятой сначала вводится мантисса как число в естественной форме, затем нажимается клавиша [ВП] — «ввод порядка» и нажимаются клавиши,

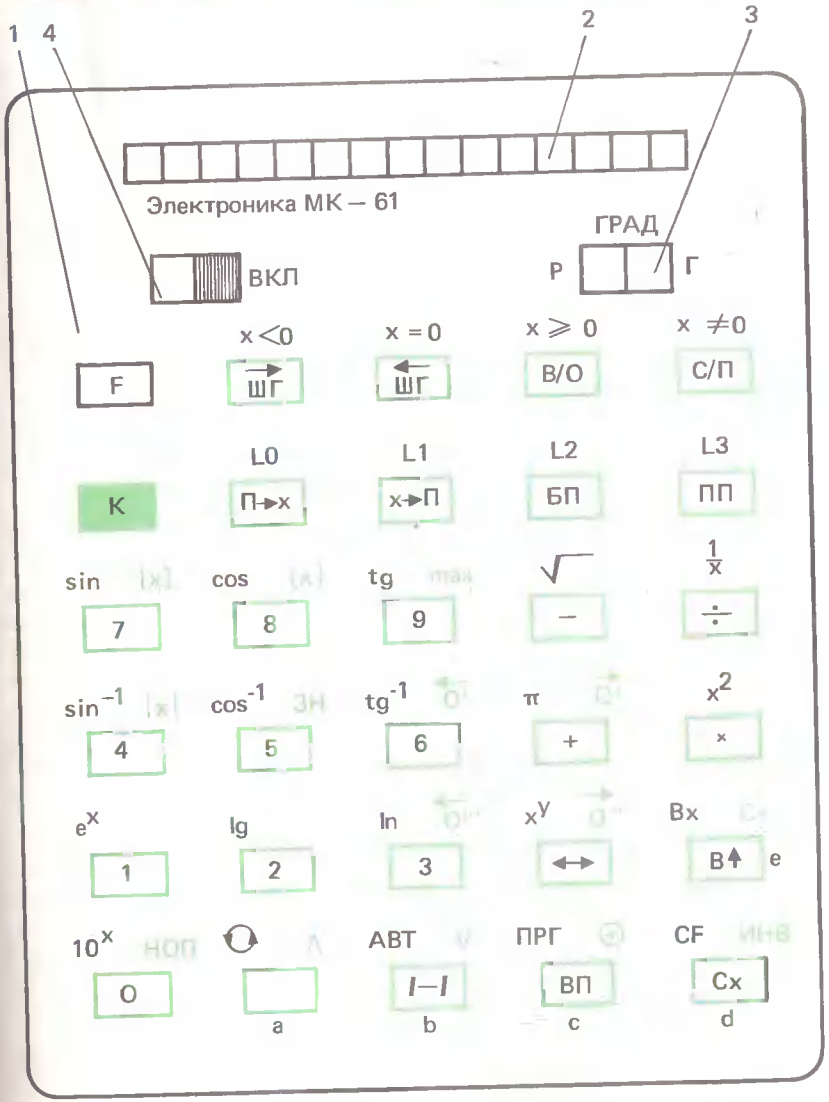
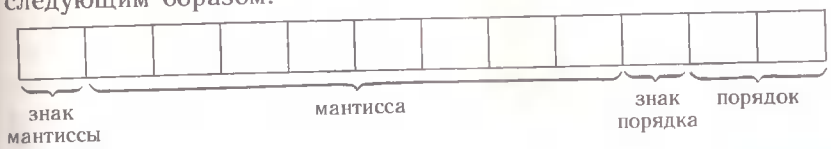


Рис. 28

дающие величину порядка.

На индикаторе числа с плавающей запятой располагаются следующим образом:



Пример 2. Ввести числа $123{,}45 \cdot 10^6$; $-0{,}17 \cdot 10^{21}$; $-12{,}47 \cdot 10^{-5}$.

Для ввода числа $123{,}45 \cdot 10^6$ необходимо нажать следующие клавиши: [1] [2] [3] [·] [4] [5] [ВП] [6]. На индикаторе будет

	1	2	3.	4	5				0	6

Нажатие следующей последовательности клавиш [0] [·] [1] [7] [/—/] [ВП] [2] [1] соответствует вводу числа $-0{,}17 \cdot 10^{21}$.

Чтобы на индикаторе появилось число $-12{,}47 \cdot 10^{-5}$, необходимо нажать следующие клавиши: [1] [2] [·] [4] [7] [/—/] [ВП] [5] [/—/].

В дальнейшем для краткости записи при вводе чисел вместо перечисления всей последовательности клавиш будем сразу писать нужное число. Например, вместо [8] [9] [·] [7] [6] [/—/] [ВП] [3] [2] [/—/] будем писать [$-89{,}76 \cdot 10^{-32}$].

Арифметические операции производятся над двумя числами, одновременно находящимися в микрокалькуляторе. Поэтому в микрокалькуляторе для хранения этих чисел имеются два регистра X и Y. Регистр X связан с индикатором, и поэтому мы всегда видим, какое число находится в регистре X. Для пересылки числа из регистра X в регистр Y существует клавиша [В↑]. При выполнении арифметических операций первое число должно находиться в регистре Y, а второе число — в регистре X. Отсюда следует правило выполнения арифметических операций над двумя числами:

1. Ввести первое число.
2. Нажать клавишу [В↑].
3. Ввести второе число.
4. Нажать клавишу арифметической операции ([+], [−], [×], [÷]).
5. Прочитать результат на индикаторе.

Таким образом, чтобы найти $a+b$, необходимо нажать следующие клавиши: [a] [В↑] [b] [+].

Последовательность клавиш, которые необходимо нажать для получения искомого результата, будем называть *программой* решения задачи.

Пример 3. Вычислить 42,721 : 7,563.

Для вычисления необходимо выполнить следующую программу: [42,721] [В↑] [7,563] [÷]. На индикаторе высветится ответ 5,6486843. Рассмотрим, как располагаются числа при выполнении деления. Вводим первое число 42,721, оно высвечивается на индикаторе, а значит, находится в регистре X. Нажимаем клавишу [В↑], число 42, 721 из регистра X перемещается в регистр Y, но по-прежнему высвечивается на индикаторе. Вводим следующее число 7,563. Это число высвечивается на индикаторе, значит, оно заменило число 42,721 в регистре X. Таким образом, в регистре Y находится число 42,721, в регистре X — число 7,563. Нажимаем клавишу [÷]. На индикаторе появляется число 5,6486843, которое располагается также в регистре X. Какое число будет в регистре Y после выполнения арифметической операции, рассмотрим позже.

Клавиша [В↑] одновременно с пересылкой числа из регистра X в регистр Y нормализует это число, а именно: числа от 1 до 99999999 представляются в естественной форме, числа меньше 1 и больше 99999999 представляются в форме с плавающей запятой, где мантисса числа $1 \leqslant M \leqslant 9$.

Пример 4. Вычислить $0{,}0015 \cdot 0{,}012$.

Для вычисления необходимо выполнить программу: [0,0015] [В↑] [0,012] [×]. На индикаторе читаем ответ $1{,}8 \cdot 10^{-5}$. Проследим за показаниями индикатора. Вводим число 0,0015, оно появляется на индикаторе. Нажимаем клавишу [В↑], число на индикаторе высвечивается в виде $1{,}5 \cdot 10^{-3}$. Микрокалькулятор выдает результат в нормализованном виде.

Пример 5. Вычислить $12{,}5 \cdot 10^3 + 17{,}2 \cdot 10^4$.

Программа вычислений: [$12{,}5 \cdot 10^3$] [В↑] [$17{,}2 \cdot 10^4$] [+]. При выполнении программы проследим за показаниями индикатора. Вводим первое число $12{,}5 \cdot 10^3$, оно высвечивается на индикаторе. Нажимаем клавишу [В↑], число на индикаторе нормализуется, т. е. представляется в виде 12 500. Вводим число $17{,}2 \cdot 10^4$, оно высвечивается на индикаторе. Нажав клавишу [+], получаем ответ 184 500 в нормализованном виде.

В микрокалькуляторе «Электроника МК-61» имеется клавиша [↔], которая позволяет менять местами содержимое регистров X и Y. Проверим это на примерах.

Пример 6. Выполним программу [5,73] [В↑] [9,1]. После ее выполнения число 5,73 находится в регистре Y, а число 9,1 — в регистре X. Нажмем клавишу [↔]. На индикаторе высветится число 5,73, значит, оно находится и в регистре X; число 9,1 перейдет при этом в регистр Y. Еще раз нажмем клавишу [↔], на индикаторе высветится число 9,1, а число 5,73 перейдет в регистр Y.

Упражнения

1. Ввести числа и нормализовать их с помощью микрокалькулятора: $-0{,}12345$; $0{,}000741$; $1{,}5 \cdot 10^6$; $43{,}33 \cdot 10^{-5}$; $-961{,}42 \times 10^{-3}$; $862{,}105 \cdot 10^{-12}$; $145{,}7 \cdot 10^{21}$.

2. Вычислить и сравнить полученный результат с данным: $5738{,}94+978{,}743=6717{,}683$; $3227{,}651-897{,}995=2329{,}656$; $46{,}3522\cdot 18{,}761=869{,}61362$; $23{,}47:37{,}5=6{,}2586666\cdot 10^{-1}$.

3. Обратить в десятичные дроби: $\frac{7}{8}$: $2\frac{5}{7}$; $\frac{11}{15}$; $14\frac{5}{9}$.

2. Левые и правые арифметические формулы

Рассмотрим арифметическое выражение $17\cdot 22+21\cdot 48$. Чтобы вычислить его значение, необходимо использовать известные правила выполнения действий. Получим $17\cdot 22=374$, $21\cdot 48=1008$, $374+1008=1382$. Порядок выполнения действий можно указывать с помощью скобок: $17\cdot 22+21\cdot 48=(17\cdot 22)+(21\cdot 48)$.

Формулы, в которых порядок выполнения действий указан скобками, назовем *скобочными формулами*.

Все скобочные формулы можно разделить на классы формул, отличающиеся порядком следования скобок.

Левой арифметической формулой назовем скобочную формулу, у которой все открывающие скобки стоят слева, т. е. в начале формулы перед первым числом.

Пример 1. Вычислить $((((3{,}8133+1{,}6785)-2{,}915)+7{,}236)-1{,}2157)+2{,}863$.

Порядок выполнения действий указан скобками, значит, это скобочная формула. В этой скобочной формуле все открывающие скобки стоят в начале формулы, поэтому она является левой арифметической формулой. Проведем вычисления по действиям: [3,8133] [В↑] [1,6785] [+], получим 5,4918; [5,4918] [В↑] [2,915] [−], получим 2,5768; [2,5768] [В↑] [7,236] [+], получим 9,8128; [9,8128] [В↑] [1,2157] [−], получим 8,5971; [8,5971] [В↑] [2,863] [+], получим 11,4601. При выполнении действий замечаем, что результат каждого предыдущего действия является первым числом при выполнении следующего действия. Так как результаты каждого действия высвечиваются на индикаторе, то программу вычисления можно записать в следующем виде: [3,8133] [В↑] [1,6785] [+] [В↑] [2,915] [−] [В↑] [7,236] [+] [В↑] [1,2157] [−] [В↑] [2,863] [+].

Эту программу можно сократить, если учесть следующее: при вводе нового числа на индикатор старое число, находящееся на индикаторе, автоматически переводится в регистр Y, если оно является результатом выполнения арифметической операции.

Итак, программа примет окончательный вид: [3,8133] [В↑] [1,6785] [+] [2,915] [−] [7,236] [+] [1,2157] [−] [2,863] [+].

Пример 2. Вычислить $(((3{,}8\cdot 46{,}0544):7{,}196)\cdot 3{,}1913)-51{,}978$.

Программа вычислений: [3,8] [В↑] [46,0544] [×] [7,196] [÷] [3,1913] [×] [51,978] [−]. Ответ. 25,634416.

Рассмотрим правило построения программ для вычисления левых арифметических формул:

1. Ввести первое число формулы.
2. Нажать клавишу [В↑].
3. Ввести следующее число формулы.
4. Нажать клавишу арифметической операции, стоящей в формуле перед этим числом.
5. Если выполнена последняя операция, то перейти к пункту 6, иначе к пункту 3.
6. На индикаторе прочитать ответ.

Правой арифметической формулой назовем скобочную формулу, у которой все закрывающие скобки стоят в конце формулы, т. е. после последнего числа.

Пример 3. Вычислить $45{,}6:(12{,}3-(74{,}1-(9{,}63:8{,}52)))$.

Порядок выполнения действий указан скобками, поэтому формула является скобочной. Все закрывающие скобки стоят после последнего числа, поэтому это правая арифметическая формула. Проведем вычисления по действиям [9,63] [В↑] [8,52] [÷], получим 1,1302816; [74,1] [В↑] [1,1302816] [−], получим 72,969718; [12,3] [В↑] [72,969718] [−], получим −60,669718; [45,6] [В↑] [−60,669718] [÷], получим $-7{,}5161054\cdot 10^{-1}$.

При выполнении действий замечаем, что результат каждого предыдущего действия есть второе число при выполнении следующего действия. Так, результатом первого действия является число 1,1302816. Оно высвечивается на индикаторе. Если теперь ввести следующее число 74,1, то оно высветится на индикаторе и расположится в регистре X, число 1,1302816 будет переведено в регистр Y. Итак, в микрокалькуляторе находятся два числа: 74,1 в регистре X и 1,1302816 в регистре Y. Над ними нужно выполнить операцию вычитания. Для вычитания первое число 74,1 должно находится в регистре Y, второе число 1,1302816 в регистре X. Тогда для выполнения действия необходимо нажать клавишу [↔], чтобы числа 74,1 и 1,1302816 поменялись местами, после чего можно производить вычитание. Проанализировав выполнение остальных действий, можно записать программу вычислений: [9,63] [В↑] [8,52] [÷] [74,1] [↔] [−] [12,3] [↔] [−] [45,6] [↔] [÷].

По аналогии с остальными действиями запишем первое, тогда программа примет вид: [8,52] [В↑] [9,63] [↔] [÷] [74,1] [↔] [−] [12,3] [↔] [−] [45,6] [↔] [÷].

Пример 4. Вычислить 167,9 — (47,5 + (13,4 · (0,7 + (15,8 — (22,1 : : 0,9))))).

Данная формула является правой арифметической формулой. Напишем программу вычислений: [0,9] [B↑] [22,1] [↔] [÷] [15,8] [↔] [—] [0,7] [↔] [+] [13,4] [↔] [×] [47,5] [↔] [+] [167,9] [↔] [—].

Анализируя данную программу, видим, что при выполнении коммутативных операций (сложение и умножение) клавиша [↔] является лишней. Поэтому программу можно сократить: [0,9] [B↑] [22,1] [↔] [÷] [15,8] [↔] [—] [0,7] [+] [13,4] [×] [47,5] [+] [167,9] [↔] [—]. Ответ. 228,34444.

Рассмотрим правило составления программ для вычисления правых арифметических формул:

1. Ввести последнее число формулы.
2. Нажать клавишу [B↑].
3. Ввести число, предшествующее введенному.
4. Если операция, стоящая после введенного числа, некоммутативная, то нажать клавишу [↔], иначе перейти к пункту 5.
5. Нажать клавишу арифметической операции.
6. Если выполнена последняя арифметическая операция, то перейти к пункту 7, иначе к пункту 3.
7. На индикаторе прочитать ответ.

Упражнения

Вычислить:

1. (((29723,69 : 33,3412) — 46,3381) · 34,05) + 23,62 = 28801,39.
2. (((410,613 + 699,591) : 44,1003) — 36,638) : 44,183 = $= -2{,}594544 \cdot 10^{-1}$.
3. ((((46,3881 · 5,6045) + 64,058) + 45,04) · 5,126) — 97,26 = = 1794,644.
4. 54,1096 + (32,2531 : (4,54349 + 5,4108)) = 57,34972.
5. 4,2856 · (52,928 + (236,344 — (237,66 : 28,15))) = 1203,522.
6. 45,3224 : (44,8981 — (2,1109 · (3,2192 + 0,4136))) = 1,217374.
7. ((((17,1 — 9,8) : 47,5) + 13,4) · 0,7) — 18,2 = —8,712422.
8. (((((18,965 — 7,41) + 24,6) · 3,7) : 26,7547) — 20) × 0,5 = —7,5.

3. Простейшие арифметические формулы

Скобочную формулу, в которой сначала идут все открывающие, а затем все закрывающие скобки, назовем *простейшей арифметической формулой*.

Примеры простейших арифметических формул:

$(k + (((c \cdot (a + b)) : d) - m)) - n$; $(((m - (c : (a : b))) - k) - d) + m$.

Из примеров видно, что результат каждого предыдущего действия является либо первым, либо вторым числом следующего действия, т. е. результат предыдущего действия обязательно входит как число в следующее действие.

Рассмотрим правило составления программ для вычисления простейших арифметических формул:

1. Записать программу первого действия.
2. Ввести число следующего действия.
3. Если выполняемая арифметическая операция коммутативная, то перейти к пункту 5, иначе к пункту 4.
4. Если первое число выполняемой арифметической операции находится в регистре X, то нажать клавишу [↔], иначе перейти к пункту 5.
5. Нажать клавишу арифметической операции.
6. Если выполнена последняя арифметическая операция, то перейти к пункту 7, иначе к пункту 2.
7. На индикаторе прочитать ответ.

Пример 1. Вычислить 323,48 — ((252,45 + 715,91) : 17,2).

Формула является простейшей арифметической. Согласно правилу составим программу: [252,45] [B↑] [715,91] [+] [17,2] [÷] [323,48] [↔] [—]. Ответ: 267,18.

Пример 2. Вычислить 1,42 — (1,522 + ((8,76 — 3,141) · 4,97)).

Формула является простейшей арифметической. Так как в формуле довольно много скобок, то для вычислений можно над арифметическими операциями поставить номер действия. Согласно правилу составим программу: [8,76] [B↑] [3,141] [—] [4,97] [×] [1,522] [+] [1,42] [↔] [—]. Ответ. —28,02843.

Из определения простейшей арифметической формулы следует, что левые и правые арифметические формулы являются простейшими арифметическими формулами.

Скобочные формулы, у которых открывающие и закрывающие скобки чередуются, назовем *арифметическими формулами*.

При составлении программ для вычисления арифметических формул можно пользоваться следующим правилом:

1. Арифметическую формулу разбить на части так, чтобы каждая часть представляла собой простейшую арифметическую формулу.
2. Составить программы для каждой простейшей арифметической формулы.
3. После выполнения каждой программы на бумаге записать результат.
4. Из полученных результатов составить арифметическую формулу.
5. Если полученная арифметическая формула простейшая арифметическая, то перейти к пункту 6, иначе к пункту 1.

6. Составить программу окончательной формулы и после ее выполнения получить ответ.

Пример 3. Вычислить

$$\frac{3{,}65\cdot(558{,}78:6{,}95-40{,}02)}{222{,}68:(880{,}08-26{,}6\cdot 32{,}8)}+\frac{197{,}08+32{,}7\cdot 257{,}42}{4501{,}68:127{,}5-269{,}87}.$$

1. Арифметическую формулу разбиваем на части так, чтобы каждая часть представляла собой простейшую арифметическую формулу.

I. $3{,}65\cdot(558{,}78:6{,}95-40{,}02)$;
II. $222{,}68:(880{,}08-26{,}6\cdot 32{,}8)$;
III. $197{,}08+32{,}7\cdot 257{,}42$;
IV. $4501{,}68:127{,}5-269{,}87$.

2. Составляем программы для каждой простейшей арифметической формулы:

I. [558,78] [В↑] [6,95] [÷] [40,02] [−] [3,65] [×];
II. [26,6] [В↑] [32,8] [×] [880,08] [↔] [−] [222,68] [↔] [÷];
III. [32,7] [В↑] [257,42] [×] [197,08] [+];
IV. [4501,68] [В↑] [127,5] [÷] [269,87] [−].

3. Выполняем каждую программу и записываем результаты:

I. 147,387; II. 29,3; III. 8614,714; IV. −234,5627.

4. Из полученных результатов составляем арифметическую формулу: $(147{,}387:29{,}3)+(8614{,}714:(-234{,}5627))$.

5. Полученная формула не является простейшей арифметической формулой, поэтому снова разбиваем ее на части так, чтобы каждая часть была простейшей арифметической формулой:

I. $147{,}387:29{,}3$;
II. $8614{,}714:(-234{,}5627)$.

6. Составляем программы для каждой части:

I. [147,387] [В↑] [29,3] [÷];
II. [8614,714] [В↑] [−234,5627] [÷].

7. Выполняем эти программы и записываем результаты:

I. 5,030273; II. −36,7267.

8. Из результатов составляем арифметическую формулу: $5{,}030273+(-36{,}7267)$.

9. Получена простейшая арифметическая формула, составляем для нее программу:

[5,030273] [В↑] [−36,7267] [+].

10. Выполняем ее и получаем ответ: −31,69642.

Итак, при вычислениях простейших арифметических формул результат получают на индикаторе микрокалькулятора, при вычислениях арифметических формул необходимо записывать на бумаге промежуточные результаты.

Упражнения

Написать программу и вычислить.

1. $66{,}4578:((22{,}2819+25{,}963)\cdot 0{,}4687-4{,}1883)+63{,}97$.
2. $17{,}1528:((32{,}9426:6{,}3526+41{,}83):5{,}781-7{,}936)$.
3. $9{,}236\cdot(3{,}412+5{,}405\cdot((9{,}591-6{,}38):8{,}945)$.
4. $1{,}109-41{,}368\cdot(54{,}928+36{,}344)+37{,}66:4{,}2523+7{,}2467$.
5. $29{,}426\cdot 10{,}613-9{,}591:6{,}638+41{,}833:64{,}058\cdot 25{,}7$.
6. $\dfrac{37{,}92-15{,}87+16{,}2}{18{,}3\cdot 2{,}7-3{,}145}-\dfrac{127{,}86:66{,}74-27{,}4}{3{,}97\cdot 2{,}82:0{,}18}$.

4. Запись и вызов чисел из регистров памяти. Вычисления с использованием регистров памяти. Стековая память

В микрокалькуляторе МК-61 имеется 15 регистров памяти. Одновременно в каждом из регистров может находиться одно число. Регистры памяти (РП) имеют номера 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e. В дальнейшем условимся запись вида $x\to$РПd читать следующим образом: число x засылается в регистр памяти с номером d.

Засылка чисел в регистры памяти производится следующим образом: число вводится в микрокалькулятор, т. е. число находится на индикаторе или, что то же самое, в регистре X. Затем нажимается клавиша [x→П] (из регистра X в регистр памяти) и клавиша с обозначением номера регистра памяти.

Пример 1. Заслать числа 17,59; −13,92; 0,15; $1{,}47\cdot 10^{-3}$ в регистры памяти 0, 1, 2, 3, соответственно.

Программа засылки:

[17,59] [x→П] [0] [−13,92] [x→П] [1] [0,15] [x→П] [2] [$1{,}47\cdot 10^{-3}$] [x→П] [3].

Числа в регистрах памяти хранятся до тех пор, пока в тот же регистр памяти не будет послано другое число. Следует помнить, что выключение микрокалькулятора стирает все числа в регистрах памяти.

Чтобы вызвать число из регистра памяти на индикатор, необходимо нажать клавишу [П→x] и клавишу с номером регистра памяти.

Пример 2. Вызвать на индикатор числа из регистров памяти 0, 1, 2, 3 (см. пример 1).

[П→x] [0], на индикаторе появится число 17,59; [П→x] [1], на индикатор будет выведено число −13,92; [П→x] [2], на индикаторе число $1{,}5\cdot 10^{-1}$; [П→x] [3], на индикаторе число $1{,}47\times 10^{-3}$.

Регистры памяти удобно использовать при вычислениях ариф-

метических формул, так как промежуточные результаты вычислений можно не записывать на бумаге, а заносить их в регистры памяти.

При выполнении арифметических операций над двумя числами не нужно числа разделять клавишей [В↑], если хотя бы одно из чисел вызывается из регистра памяти.

Пример 3. Вычислить $(19{,}46 \overset{1}{-} 8{,}79)^4(17{,}27 \overset{2}{-} 3{,}45)^5(9{,}81 \overset{3}{+} 10{,}17)$.

Установим порядок выполнения действий и составим план вычислений. Выполним 1-е действие и результат занесем в РПа, выполним 2-е действие и результат занесем в РПb, выполним третье действие и результат занесем в РПс. Затем выполним 4-е и 5-е действия. Исходя из такого плана, получим следующую программу: [19,46] [В↑] [8,79] [−] [x→П] [a] [17,27] [В↑] [3,45] [−] [x→П] [b] [9,81] [В↑] [10,17] [+] [x→П] [c] [П→x] [a] [П→x] [b] [×] [П→x] [c] [×]. Ответ. 2946, 2388.

В этом примере можно установить другой порядок выполнения действий, что даст возможность использовать только один регистр памяти: $(19{,}46 \overset{4}{-} 8{,}79)^5(17{,}27 \overset{2}{-} 3{,}45)^3(9{,}81 \overset{1}{+} 10{,}17)$. В результате получим следующую программу: [9,81] [В↑] [10,17] [+] [x→П] [a] [17,27] [В↑] [3,45] [−] [П→x] [a] [×] [x→П] [a] [19,46] [В↑] [8,79] [−] [П→x] [a] [×]. Очевидно, что и сама программа стала короче.

При составлении программ следует стремиться к тому, чтобы программы использовали как можно меньше регистров памяти и были короче.

Пример 4. Вычислить $\dfrac{(15{,}28 + 17{,}93) \cdot 7{,}2}{272{,}4 : (49{,}5 - 16{,}8)} - \dfrac{(3{,}97 + 12{,}26) - (0{,}57 \cdot 4{,}2)}{7{,}6 \cdot 12{,}8 + 24{,}1}$.

Эта формула арифметическая. Разобьем ее на части так, чтобы каждая часть представляла собой простейшую арифметическую формулу: *a*) $(15{,}28 + 17{,}93) \cdot 7{,}2$; *b*) $272{,}4 : (49{,}5 - 16{,}8)$; *c*) $3{,}97 + 12{,}26$; *d*) $0{,}57 \cdot 4{,}2$; *e*) $7{,}6 \cdot 12{,}8 + 24{,}1$.

Используя обозначения частей формулы, запишем исходную формулу в виде $a \overset{3}{:} b \overset{4}{-} (c \overset{1}{-} d) \overset{2}{:} e$, где цифрами укажем порядок действий. Отсюда видим, что сначала надо вычислить *d* и результат занести, например, в РП1. Потом найти $c - d$ и результат занести в РП1. Затем вычислить *e*, но перед выполнением деления нажать клавишу [↔], результат деления занести в РП1. Вычислить *b* и результат занести в РП2, вычислить *a* и разделить на *b*, вычесть число из РП1. В результате получаем следующую программу: [0.57] [В↑] [4.2] [×] [x→П] [1] [3,97] [В↑] [12,26] [+] [П→x] [1] [−] [x→П] [1] [7,6] [В↑] [12,8] [×] [24.1] [+] [П→x] [1] [↔] [÷] [x→П] [1] [49.5] [В↑] [16.8] [−] [272,4] [↔] [÷] [x→П]

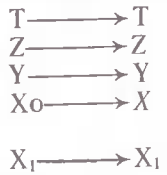
[2] [15.28] [В↑] [17.93] [+] [7.2] [×] [П→x] [2] [÷] [П→x] [1] [−].

Ответ: 28,589984.

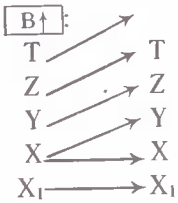
В МК-61 имеется еще один вид памяти — стековая память. Она состоит из четырех регистров X, Y, Z, T. Туда же относят регистр XI (регистр предыдущего результата).

Рассмотрим все возможные случаи передвижения информации в регистрах стека.

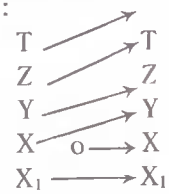
1. Ввод чисел после нажатия клавиш [В↑] или [C_x]:



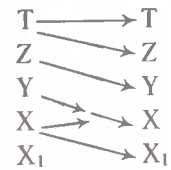
2. Нажатие клавиши [В↑]:



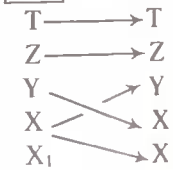
3. Ввод числа после выполнения арифметической операции, вычисления значения элементарных функций, засылки и вызова числа из регистра памяти:



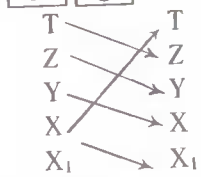
4. Нажатие клавиш [+], [−], [×], [÷]:



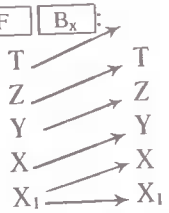
5. Нажатие клавиши [↔]:



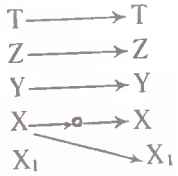
6. Нажатие клавиш [F] [↻]:



7. Нажатие клавиш [F] [Bx]:



8. Вычисление значений элементарных функций, т. е. после нажатия клавиш [F] [sin], [F] [cos], [F] [tg], [F] [sin⁻¹], [F] [cos⁻¹], [F] [tg⁻¹], [F] [x²], [F] [√], [F] [1/x], [F] [eˣ], [F] [10ˣ], [F] [ln], [F] [lg], [F] [xʸ], [K] [[x]], [K] [{x}], [K] [|x|], [K] [зн]:



Чтобы быстрее усвоить передвижение информации в регистрах стека, на первых порах следует рисовать таблицы.

Пример 5. Вычислить $(27{,}5 \cdot 31{,}2) + (72{,}8 : 22{,}1)$.

Программа вычислений приведена в таблице 1.

Таблица 1

Программа	Регистры стека			
	X	Y	Z	T
[27.5]	27.5	—	—	—
[B↑]	27.5	27.5	—	—
[31.2]	31.2	27.5	—	—
[×]	858	—	—	—
[72,8]	72.8	858	—	—
[B↑]	72.8	72.8	858	—
[22.1]	22.1	72.8	858	—
[÷]	3.2941176	858	—	—
[+]	861.29412	—	—	—

Пример 6. Вычислить $\dfrac{716{,}5}{32{,}4 + 41{,}1} - \dfrac{(47{,}8 - 21{,}3)(72{,}1 - 69{,}5)}{3{,}6 \cdot 7{,}2}$.

Программа вычислений и передвижение информации в регистрах стека приведены в таблице 2.

Таблица 2

Программа	Регистры стека			
	X	Y	Z	T
[716.5]	716.5	—	—	—
[B↑]	716.5	716.5	—	—
[32.4]	32.4	716.5	—	—
[B↑]	32.4	32.4	716.5	—
[41.1]	41.1	32.4	716.5	—
[+]	73.5	716.5	—	—
[÷]	9.7482993	—	—	—
[47.8]	47.8	9.7482993	—	—
[B↑]	47.8	47.8	9.7482993	—
[21.3]	21.3	47.8	9.7482993	—
[—]	26.5	9.7482993	—	—
[72.1]	72.1	26.5	9.7482993	—
[B↑]	72.1	72.1	26.5	9.7482993
[69.5]	69.5	72.1	26.5	9.7482993
[—]	2.6	26.5	9.7482993	9.7482993
[×]	68.9	9.7482993	9.7482993	9.7482993
[3.6]	3.6	68.9	9.7482993	9.7482993
[B↑]	3.6	3.6	68.9	9.7482993
[7.2]	7.2	3.6	68.9	9.7482993
[×]	25.92	68.9	9.7482993	9.7482993
[÷]	2.658179	9.7482993	9.7482993	9.7482993
[—]	7.0901203	9.7482993	9.7482993	9.7482993

Если при вычислениях не хватает регистров стека, то следует использовать другие регистры памяти.

Упражнения

Вычислить:

1. $\dfrac{0{,}3125 : 0{,}125 + 2{,}5 \cdot 6{,}31}{(4{,}525 - 2{,}5 : 10{,}1) : 0{,}302 + 136{,}7}$.
2. $\dfrac{158{,}47 : 6{,}32 + 21{,}6 : 15{,}6}{102{,}58 - (216{,}8 : 8{,}71 - 51{,}62 : 78{,}4)}$.
3. $\dfrac{19{,}604 + 1{,}52 \cdot (35{,}8 : 2{,}7 - 30{,}4 : 11{,}05)}{11{,}809 : 14{,}6 + 0{,}904}$.

4. $\dfrac{2{,}098\cdot 3{,}21-4{,}29\cdot 3{,}07}{5{,}19\cdot 2{,}9}+\dfrac{33{,}1776:5{,}12+6{,}44}{6{,}72-2{,}89}$.

5. $\dfrac{6{,}2\cdot 27{,}8+13{,}15-3{,}89}{2{,}5+43{,}9\cdot 8{,}3-817{,}59}-\dfrac{35{,}8+84{,}6\cdot 9{,}2-548{,}4}{92{,}34-12{,}03+27{,}8}$.

5. Общие свойства вычисления значений элементарных функций

Каждая клавиша МК-61 имеет двойную или тройную символику. Рассмотрим вычисление значений элементарных функций, написанных над клавишами желтым цветом: $\sin x$, $\cos x$, $\operatorname{tg} x$, $\sin^{-1} x$, $\cos^{-1} x$, $\operatorname{tg}^{-1} x$, e^x, 10^x, $\ln x$, $\lg x$, $\sqrt{x}$, $\frac{1}{x}$, π, x^2, x^y, где $\sin^{-1} x$, $\cos^{-1} x$, $\operatorname{tg}^{-1} x$ — обратные тригонометрические функции. Чтобы перейти на второе назначение клавиши, предварительно необходимо нажать префиксную клавишу [F], которая тоже желтого цвета.

Итак, для вычисления значений функции $f=\{\sin, \cos, \operatorname{tg}, \sin^{-1}, \cos^{-1}, \operatorname{tg}^{-1}, e^x, 10^x, \ln x, \lg x, \sqrt{x}, \frac{1}{x}, x^2\}$ необходимо выполнить программу [x] [F] [f], где [f] клавиша, над которой записана нужная элементарная функция.

При вычислении значений элементарных функций полученные результаты необходимо округлять.

Пример 1. Вычислить $7{,}5^2$.

Программа: [7,5] [F] [x^2]. Ответ. 56,25.

Пример 2. Вычислить $\frac{1}{8}$.

Программа: [8] [F] [1/x]. Ответ. 0,125.

Пример 3. Вычислить $\sqrt{12{,}75}$.

Программа: [12,75] [F] [√]. Ответ. 3,5707142.

Пример 4. Вычислить $10^{0.6}$.

Программа: [0,6] [F] [10^x]. Ответ. 3,9810715.

Пример 5. Вычислить e^1.

Программа: [1] [F] [e^x]. Ответ. 2,7182818.

Пример 6. Вычислить $\lg 5{,}7$.

Программа: [5,7] [F] [lg]. Ответ. 0,75587482.

Пример 7. Вычислить $\ln 14{,}6$.

Программа: [14,6] [F] [ln]. Ответ. 2,6810215.

Пример 8. Вычислить $\dfrac{1}{0{,}7^2+\sqrt{5{,}4}}$.

Программа: [0,7] [F] [x^2] [B↑] [5,4] [F] [√] [+] [F] [1/x]. Ответ. 0,35539254.

Для вычисления значений элементарных функций x^2, $\frac{1}{x}$, $\sqrt{x}$, e^x, 10^x, $\ln x$, $\lg x$ в микрокалькулятор необходимо вводить только одно число, поэтому эти операции называются одноместными. При вычислении этих функций число вводится в регистр X, затем дается команда микрокалькулятору вычислить значение необходимой функции, и результат опять высвечивается на индикаторе, т. е. в регистре X. Итак, вычисление функций не затрагивает регистр Y.

Пример 9. Вычислить $\dfrac{\ln 37{,}5+\lg 41{,}8}{e^{2{,}9}}-\dfrac{1}{\sqrt{23{,}8}-\sqrt{17{,}3}}$.

Программа: [37,5] [F] [ln] [41,8] [F] [lg] [+] [2,9] [F] [e^x] [÷] [23,8] [F] [√] [17,3] [F] [√] [−] [F] [1/x] [−]. Ответ. −1,1018133.

Вычисление x^y — это двухместная операция, так как необходимо вводить два числа x и y. Поэтому при вычислении x^y заняты регистры X и Y. Программа вычисления x^y: [y] [B↑] [x] [F] [x^y].

Пример 10. Вычислить $\sqrt[5]{53{,}8}$.

Программа: [5] [F] [1/x] [53,8] [F] [x^y]. Ответ. 2,2189956.

Пример 11. Вычислить

$$\frac{9{,}87\cdot 13{,}2-\sqrt[7]{8{,}25}}{24{,}6+49{,}8}-\frac{\sqrt[3]{7{,}2}+\sqrt[5]{2{,}7^3}}{e^{0{,}3}+e^{2{,}2}}.$$

Программа: [0,3] [F] [e^x] [2,2] [F] [e^x] [+] [x→П] [1] [3] [F] [1/x] [7,2] [F] [x^y] [x→П] [2] [3] [B↑] [5] [÷] [2,7] [F] [x^y] [П→x] [2] [+] [П→x] [1] [÷] [x→П] [1] [7] [F] [1/x] [8,25] [F] [x^y] [x→П] [2] [9.87] [B↑] [13.2] [×] [П→x] [2] [−] [x→П] [2] [24,6] [B↑] [49,8] [+] [П→x] [2] [↔] [÷] [П→x] [1] [−]. Ответ. 1,3719201.

Упражнения

Вычислить:

1. $\dfrac{10^x-x^3}{\sqrt{e^{x^2}-1}}$, при $x=1{,}3$.

2. $\dfrac{\ln 17{,}3+6{,}9^4\sqrt{e^{0{,}4}}-\sqrt{45{,}89}}{\sqrt{84{,}32}+\ln(28{,}4:7{,}2+248{,}1)}$.

3. $\dfrac{16{,}2^2-\sqrt{11{,}8}:14{,}6+0{,}9}{\sqrt[3]{35{,}8}\cdot 11{,}05+\ln 196{,}4}$.

4. $\dfrac{\lg 27{,}3:\lg 31{,}2-\ln 17{,}1:\ln 42{,}8}{\sqrt[8]{2{,}7^3}+\sqrt[6]{3{,}1^2}}$.

6. Особенности вычисления тригонометрических функций

Вычислять значения тригонометрических функций можно в градусах, радианах и градах. Для этого необходимо следить за положением переключателя *3* (см. рис. 28). МК-61 позволяет непосредственно вычислять значения $\sin x$, $\cos x$, $\operatorname{tg} x$, $\arcsin x$, $\arccos x$, $\operatorname{arctg} x$, причем это одноместные операции.

Программа вычисления этих функций: [x] [F] [f], где x — значение аргумента функции, заданное в соответствующих единицах измерения, f — вычисляемая тригонометрическая функция.

МК-61 позволяет также вызывать на индикатор число π. Для этого необходимо выполнить программу [F] [π] и на индикаторе высветится 3,1415926.

Пример 1. Вычислить sin 0,72.

Так как аргумент задан в радианах, то переключатель *3* необходимо поставить в положение Р и выполнить программу: [0,72] [F] [sin]. Получим ответ 0,65938472.

Пример 2. Вычислить $\operatorname{ctg} 27°$.

Для вычисления можно воспользоваться формулой $\operatorname{ctg} 27° = \frac{1}{\operatorname{tg} 27°}$. Так как аргумент задан в градусах, то переключатель *3* необходимо поставить в положение Г. Программа: [27] [F] [tg] [F] [1/x]. Ответ. 1,9626105.

Пример 3. Вычислить $\cos \frac{3\pi}{4}$.

Переключатель *3* поставить в положение Р. Программа: [3] [B↑] [F] [π] [×] [4] [÷] [F] [cos]. Ответ. —0,70710676.

Пример 4. Вычислить arcsin 0,3.

При вычислении обратных тригонометрических функций переключатель *3* нужно ставить в то положение, в каких единицах измерения мы хотим получить ответ. В данном примере угол нужно получить в градусах, поэтому переключатель *3* поставим в положение Г и выполним программу [0,3] [F] [sin⁻¹]. Ответ. 17,457604°.

Пример 5. Вычислить $\frac{\sin 15° + \operatorname{tg} 32° \cdot \cos 24°}{\operatorname{ctg} 73° - \cos 52°}$.

Переключатель *3* должен стоять в положении Г. Программа: [73] [F] [tg] [F] [1/x] [52] [F] [cos] [—] [x→П] [a] [32] [F] [tg] [24] [F] [cos] [×] [15] [F] [sin] [+] [П→x] [a] [÷]. Ответ. —2,6769375.

Программу вычислений можно записать по-другому, если использовать стек: [15] [F] [sin] [32] [F] [tg] [24] [F] [cos] [×] [+] [73] [F] [tg] [F] [1/x] [52] [F] [cos] [—] [÷].

Пример 6. Вычислить $\operatorname{tg} \frac{3\pi}{5} + \operatorname{ctg} 41° \cos \frac{\pi}{7} - \sin 15°$.

Если в формуле аргументы заданы и в градусах, и в радианах, то переключатель *3* необходимо ставить в нужное положение перед каждой вычисляемой тригонометрической функцией. Запишем программу: Р←Г [3] [B↑] [F] [π] [×] [5] [÷] [F] [tg] [x→П] [0] [F] [π] [B↑] [7] [÷] [F] [cos] Р→Г [41] [F] [tg] [F] [1/x] [×] [П→x] [0] [+] [15] [F] [sin] [—]. Ответ. —2,3000571.

Иногда приходится вычислять значения тригонометрических функций, когда угол задан в градусах и минутах. В таких случаях минуты необходимо перевести в доли градуса. МК-61 позволяет производить этот перевод автоматически, для этого используются клавиши [→0'], [←0']. На микрокалькуляторе эти обозначения нанесены синим цветом, т. е. это третье назначение клавиш [6] и [+]. Чтобы перейти на это назначение, необходимо предварительно нажать префиксную клавишу [K], которая тоже закрашена синим цветом.

Пример 7. Выразить 32°16′ в градусах.

Программа: [32,16] [K] [→0']. Ответ. 32,266666°.

Пример 8. Выразить 20°30,45′ в градусах и долях градуса.

Программа: [20,3045] [K] [→0']. Ответ. 20,5075°.

Можно производить и обратную операцию, переводить градусы и доли градуса в градусы, минуты и доли минуты.

Пример 9. Выразить 47,2143° в градусах и минутах.

Программа: [47,2143] [K] [←0']. На индикаторе высвечивается 47,12858. Нужно читать 47°12,858′.

Пример 10. Вычислить sin 15°42,2′.

Программа: [15,422] [K] [→0'] [F] [sin]. Ответ. 0,27065643.

Пример 11. Вычислить arccos 0,7 и ответ выразить в градусах и минутах.

Программа: [0,7] [F] [cos⁻¹] [K] [←0']. Ответ. 45°34,3797′.

Перевод градусов и долей градуса в градусы и минуты и обратная операция являются одноместными.

Упражнения

Вычислить:

1. $\arcsin 0{,}7 + \arccos 0{,}3 - \operatorname{arctg} 1{,}5$ (ответ получить в градусах и минутах);

2. $\dfrac{\sin\frac{\pi}{3}+\cos\frac{\pi}{3}+\operatorname{tg}\frac{\pi}{3}}{\sin\frac{\pi}{6}\cos\frac{\pi}{6}\operatorname{tg}\frac{\pi}{6}}$;

3. $\sqrt{\dfrac{\cos 41°17'-\sin 20°13'}{\operatorname{tg} 52°24'}}$;

4. $\dfrac{\sin 40°\cos 20°-\sin 20°\cos 40°}{\operatorname{tg}^2 35°}$;

5. $\operatorname{ctg}^2 36°17'+\sqrt{\operatorname{tg} 23°}$.

7. Решение задач с использованием элементарных функций

Операции выделения целой и дробной части числа, определения знака числа, нахождения модуля числа, определения случайных величин на отрезке от 0 до 1, перевод часов, минут, секунд в часы и доли часа и обратная операция являются одноместными операциями и выполняются с использованием префиксной клавиши [K]. Все эти операции очень часто используются при программировании, но могут использоваться и при решении задач обычным способом.

Пример 1. Определить целую часть результата вычислений

$$\frac{37{,}8\cdot 42{,}4-12{,}5\cdot 17{,}8}{23{,}6}.$$

Программа: [37.8] [B↑] [42.4] [×] [12.5] [B↑] [17.8] [×] [—] [23.6] [÷] [K] [|x|]. Ответ. 58.

Пример 2. Определить дробную часть числа ln 5,76.

Программа: [5,76] [F] [ln] [K] [{x}]. Ответ. 0,7509374.

Пример 3. Определить |cos 200°|.

Программа: [200] [F] [cos] [K] [|x|]. Ответ. 0,93969264.

Пример 4. Определить знак результата вычислений ln 2,02 — — ln 4,5.

Программа: [2,02] [F] [ln] [4,5] [F] [ln] [—] [K] [ЗН]. Ответ. —1. Это значит, что знак полученного числа отрицательный. Если знак числа положительный, то на индикаторе высвечивается 1.

Пример 5. Выразить 6 ч 15 мин 40 с в долях часа.

Программа: [6,1540] [K] [0'''] . Ответ. 6,261111 ч.

Пример 6. Выразить 18,5022 часа в часах, минутах, секундах.

Программа: [18,5022] [K] [0'''] . Ответ. 18 ч 30 мин 7,92 с.

Если нажать на клавиши [K] [Сч], на индикаторе появится любое случайное число из промежутка]0; 1[.

Операция нахождения наибольшего из двух чисел является двухместной и производится по программе [a] [B↑] [b] [K] [max], где a и b — числа.

Пример 7. Определить, какое из двух чисел больше: 0,345 или 2,726.

Программа: [0,345] [B↑] [2,726] [K] [max]. Ответ. 2,726.

Пример 8. Какое из чисел больше: $\frac{3}{8}$ или $\frac{1}{5}$?

Программа: [3] [B↑] [8] [÷] [5] [F] [1/x] [—] [K] [ЗН]. Ответ. 1. Значит, разность $\frac{3}{8}-\frac{1}{5}$ положительна, а из этого следует, что $\frac{3}{8}>\frac{1}{5}$.

Замечание.

При работе с микрокалькулятором в обычном режиме на индикаторе может высвечиваться ЕГГОГ, что обозначает невозможность выполнения действия. Это может произойти в следующих случаях: деление на 0; при вычислении $\frac{1}{x}$, если $x=0$; при вычислении $\ln x$, $\lg x$, $\sqrt{x}$, если $x\leqslant 0$; при возведении x в степень y, если $x=0$; при вычислении $\operatorname{tg} x$, если $x=\frac{\pi}{2}+\pi k$; при вычислении $\arcsin x$, $\arccos x$, если $|x|>1$; при переводе временных и угловых величин, если значения минут или секунд $\geqslant 60$; если в результате вычислений получается число, модуль которого больше, чем число $9{,}999999\cdot 10^{99}$.

Упражнения.

1. Вычислить $([13{,}44]+[15{,}62])\cdot\{7{,}43\}$.
2. Перевести в часы, минуты и секунды 7,23 ч, 46,75 ч, 21,35 ч.
3. Найти $\left|\dfrac{x^2-23x+5}{x-1{,}2}\right|$, если $x=0{,}3$; 3,4. (Указание: предварительно x заслать в какой-нибудь регистр памяти.)
4. Определить, какое из чисел больше: $\frac{17}{26}$ или $\frac{5}{6}$. (Указание: написать две программы. Одну с использованием клавиши [max], другую — [ЗН].)

8. Решение задач

Задача 1. В треугольнике ABC известны две стороны: $a=12{,}6$, $b=7{,}5$ и угол между ними $\gamma=32°$. Найти сторону c, углы α и β, площадь S.

Решение. По теореме косинусов имеем $c=\sqrt{a^2+b^2-2ab\cos\gamma}$. Можем записать следующую программу вычисления c: [12,6] [B↑] [7,5] [×] [32] [F] [cos] [×] [2] [×] [/—/] [12,6] [F] [x²] [+] [7,5] [F] [x²] [+] [F] [√]. Ответ. $c=7{,}398$.

Площадь треугольника $S=\frac{1}{2}ab\sin\gamma$. Программа вычисления S: [12,6] [B↑] [7,5] [×] [32] [F] [sin] [×] [2] [÷]. Ответ. $S=25{,}039$. По теореме синусов можно найти угол α: $\frac{a}{\sin\alpha}=\frac{c}{\sin\gamma}$, $\sin\alpha=\frac{a\sin\gamma}{c}$, $\alpha=\arcsin\frac{a\sin\gamma}{c}$. Программа вычисления α: [32] [F] [sin] [12,6] [×] [7,398] [÷] [F] [$\sin^{-1}$], $\alpha=64{,}493822°$ или, если нажать [K] [$\overleftarrow{0}$], то $\alpha\approx64°30'$. Угол β найдем из равенства $\alpha+\beta+\gamma=180°$; $\beta=180°-\alpha-\gamma$. Программа вычислений β: [180] [B↑] [32] [−] [64,49] [−]. Ответ. 83,51° или 83°30′.

Задача 2. Написать программу вычисления площади S треугольника ABC, если известны три его стороны a, b, c.

Решение. Площадь S находится по формуле Герона

$$S=\sqrt{p(p-a)(p-b)(p-c)},\ \text{где}\ p=\frac{a+b+c}{2}.$$

Программа: [a] [B↑] [b] [+] [c] [+] [2] [÷] [x→П] [0] [a] [−] [x→П] [1] [П→x] [0] [b] [−] [П→x] [1] [×] [x→П] [1] [П→x] [0] [c] [−] [П→x] [1] [×] [П→x] [0] [×] [F] [√].

Задача 3. Прямоугольный треугольник задан катетами a и b. Найти радиус R описанной и r вписанной окружностей треугольника.

Решение. Радиусы описанной и вписанной окружностей вычисляются по формулам: $R=\frac{abc}{4S}$, $r=\frac{2S}{a+b+c}$. Неизвестные величины S и c определяются по формулам: $S=\frac{1}{2}ab$, $c=\sqrt{a^2+b^2}$. Программы вычислений S и c: [a] [B↑] [b] [×] [2] [÷] [x→П] [1] [a] [F] [x^2] [b] [F] [x^2] [+] [F] [√] [x→П] [2]. Программа вычисления R: [a] [B↑] [b] [×] [П→x] [2] [×] [4] [÷] [П→x] [1] [÷]. Программа вычисления r: [a] [B↑] [b] [+] [П→x] [2] [+] [x→П] [3] [2] [П→x] [1] [×] [П→x] [3] [÷].

Задача 4. Решить неравенство

$$\frac{\sqrt{x-1{,}5}(x^2-7{,}29)(5{,}7-x)}{(x-3{,}4)\sqrt{16{,}2-x}}>0.$$

Решение. Область определения функции

$$\begin{cases}x-1{,}5\geqslant0,\\ x-3{,}4\neq0,\\ 16{,}2-x>0,\end{cases}\Leftrightarrow\begin{cases}x\geqslant1{,}5,\\ x\neq3{,}4,\\ x<16{,}2,\end{cases}\quad[1{,}5;\ 3{,}4[\cup]3{,}4;\ 16{,}2[.$$

Находим промежутки знакопостоянства выражения, стоящего в левой части неравенства, и наносим их на числовую ось.

$$x^2-7{,}29=0,\ |x|=\sqrt{7{,}29},\ x_1=2{,}7,\ x_2=-2{,}7,$$
$$5{,}7-x=0,\ x=5{,}7.$$

Для определения знака функции на каждом из промежутков выбираем любое значение x из данного промежутка и вычисления проводим по следующей программе, подставляя вместо x выбранное значение. [x] [B↑] [1,5] [−] [F] [√] [x] [F] [x^2] [7,29] [−] [×] [5,7] [B↑] [x] [−] [×] [x] [B↑] [3,4] [−] [16,2] [B↑] [x] [−] [F] [√] [×] [÷] [K] [ЗН].

Из промежутка [1,5; 2,7[возьмем $x=2$, проведем вычисления по программе, получим 1, т. е. на этом промежутке неравенство принимает знак «+». Теперь из промежутка]2,7; 3,4[выбираем $x=3$ и после выполнения программы получаем «−» и т. д. Решением неравенства будет]1,5; 2,7[∪]3,4; 5,7[.

Задача 5. Вычислить первые пять членов последовательности $x_n=2+\frac{3}{n}$.

Решение. Вычисления проводим по следующей программе, подставляя $n=1, 2, 3, 4, 5$: [3] [B↑] [n] [÷] [2] [+]. Получим $x_1=5$; $x_2=3{,}5$; $x_3=3$; $x_4=2{,}75$; $x_5=2{,}6$.

Задача 6. Экспериментально доказать, что $\lim\limits_{n\to\infty}\frac{3n-1}{n}=3$, взяв $\varepsilon=0{,}1$.

Решение. Напомним определение предела последовательности. Число $x(x=3)$ называется пределом последовательности $\left(x_n=\frac{3n-1}{n}\right)$, если для любого $\varepsilon>0$ ($\varepsilon=0{,}1>0$) при всех достаточно больших номерах n выполняется неравенство $|x-x_n|<\varepsilon$. Поэтому необходимо найти номер n, начиная с которого будет выполняться неравенство $|3-x_n|<0{,}1$. Вычислим несколько членов этой последовательности по программе [n] [B↑] [3] [×] [1] [−] [n] [÷] [3] [−] [K] [|x|].

$|3-x_1|=1$; $|3-x_2|=0{,}5$; $|3-x_3|=0{,}3333334$; $|3-x_4|=0{,}25$; $|3-x_5|=0{,}2$; $|3-x_6|=0{,}1666667$; $|3-x_7|=0{,}1428572$; $|3-x_8|=0{,}125$; $|3-x_9|=0{,}1111112$; $|3-x_{10}|=0{,}1$; $|3-x_{11}|=0{,}0909091$; $|3-x_{12}|=0{,}083334$; $|3-x_{13}|=0{,}0769231$; $|3-x_{14}|=0{,}0714286$.

Видим, что модуль разности все время уменьшается и начиная с номера 11 $|3-x_{11}|<0{,}1$.

9. Программирование на микрокалькуляторе. Программная память. Понятие команды. Адрес и код команды. Составление линейных программ

Рассмотрим на примере составление линейных программ.

Задача. Написать программу вычисления длины гипотенузы прямоугольного треугольника, если известны его катеты a и b.

Решение. Длина гипотенузы вычисляется по формуле $c = \sqrt{a^2+b^2}$. Поэтому для получения ответа на микрокалькуляторе необходимо выполнить следующую программу: [a] [F] [x²] [b] [F] [x²] [+] [F] [√], где вместо [a] и [b] ввести конкретные числа, выражающие длины катетов a и b.

Изменим немного условие задачи. Пусть необходимо найти длины гипотенуз нескольких треугольников. Тогда эту программу надо выполнить несколько раз, всякий раз подставляя вместо a и b необходимые числа. Таким образом, последовательность нажатия клавиш будет несколько меняться в зависимости от чисел a и b. Чтобы этого не происходило, мы можем договориться, что предварительно числа a и b будут занесены в регистры памяти, например $a \to$ РП1, $b \to$ РП2. Тогда программа вычислений примет вид: [П→x] [1] [F] [x²] [П→x] [2] [F] [x²] [+] [F] [√]. Чтобы найти гипотенузы, например у десяти треугольников, необходимо 10 раз нажать указанные клавиши, предварительно заслав каждый раз значения a и b в соответствующие регистры памяти. Работу можно существенно упростить, если занести составленную программу в программную память микрокалькулятора, поручив ему выполнять эту программу автоматически.

Программная память микрокалькулятора состоит из 105 ячеек. Каждая ячейка имеет свой номер: 00, 01, 02, ..., 104. В каждую ячейку программной памяти можно записать только одну команду, а некоторые команды, которые мы рассмотрим позже, будут занимать две ячейки.

Ячейки программной памяти связаны между собой подобно кольцевому стеку, поворачиваемому на один шаг после ввода очередной команды. Поэтому при вводе более 105 команд программная память переполняется, а лишние команды записываются вместо ранее записанных команд.

Чтобы записать программу в программную память, микрокалькулятор необходимо перевести в режим «Программирование». Для этого нажимают клавиши [F] [ПРГ], при этом в правом углу индикатора высветится 00. Это означает, что счетчик ячеек программной памяти установлен в начальное положение и ячейка с номером 00 свободна, — туда можно заносить команду. Нажимаем первую клавишу программы [П→x]. На индикаторе ничего не изменилось. Нажимаем клавишу [1], на индикаторе в левом углу появилось число 61. Это код команды [П→x] [1]. Одновременно с появлением числа 61 изменилось на 01 показание счетчика ячеек программной памяти. В дальнейшем номера ячеек, высвечиваемые на индикаторе, будем называть адресом команды. Нажимаем клавишу [F] программы — на индикаторе никаких изменений. Значит, одна клавиша [F] команды не образует. Нажимаем клавишу [x²]. На индикаторе появляется код 22, причем код предыдущей команды сдвинулся вправо. Счетчик адресов команд показывает 03. Нажимая последовательно все клавиши программы, мы видим, что на индикаторе высвечиваются коды последних трех команд и адрес ячейки программной памяти, куда можно заносить следующую команду. Поскольку на индикаторе высвечиваются адреса и коды, то в дальнейшем программы будем записывать в виде таблицы (табл. 3).

Таблица 3

Адрес команды	Команда	Код команды
00	[П→x] [1]	61
01	[F] [x²]	22
02	[П→x] [2]	62
03	[F] [x²]	22
04	[+]	10
05	[F] [√]	21
06	[С/П]	50

В ячейки программной памяти программа записывается в виде последовательности кодов команд. После выполнения последней команды извлечения квадратного корня машина должна остановиться. Для этого используется команда [С/П]. После ввода всей программы в программную память микрокалькулятор переводят в режим «Работа» нажатием клавиши [F] [АВТ].

Счет по программе производится следующим образом. Исходные числовые данные заносятся в соответствующие регистры памяти. Пусть $a=3$, $b=4$. Тогда $3 \to$ РП1, $4 \to$ РП2. Затем счетчик адресов команд устанавливаем на номер 00, так как с него начинается наша программа. Для этого существует клавиша [В/О]. Затем пускаем программу для автоматического просчета нажатием клавиши [С/П]. Индикатор начнет слегка мигать, что свидетельствует о выполнении программы. После окончания счета на индикаторе высветится число 5.

Чтобы найти гипотенузу другого треугольника, необходимо ввести новые числовые данные в регистры памяти 1 и 2 и запустить программу на выполнение нажатием клавиш [В/О] [С/П].

Выключение микрокалькулятора стирает программу в программной памяти.

Упражнения

Написать программы решения следующих задач:

1. Вычислить расстояние между двумя точками, заданными своими координатами $A(x_1, y_1)$, $B(x_2, y_2)$ по формуле

$$d=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}.$$

2. Вычислить k-й член последовательности $x_n=|4-2n|$.

3. Вычислить объем шара, если известен радиус R.

4. В треугольнике известны три стороны a, b, c. Найти радиус вписанной и описанной окружностей.

5. Составить таблицу значений функции $y=\operatorname{tg} 3x+\cos x-\sqrt{x}$ на промежутке [0; 1] с шагом 0,1.

10. Ввод, редактирование и отладка программы

Итак, программа составлена и введена в программную память микрокалькулятора. По ней можно считать и получать какие-то результаты. Однако встает вопрос: правильно ли написана программа и получим ли мы по нашей программе тот ответ, который требуется. Программисты говорят в этом случае, что необходимо отладить программу, т. е. исправить ошибки, если они есть. Для этого составляют контрольный пример. Вернемся к задаче о нахождении гипотенузы треугольника. Мы взяли $a=3$, $b=4$, получили $c=5$. Правильный ли это ответ? Для проверки достаточно просчитать вручную $c=\sqrt{3^2+4^2}=\sqrt{9+16}=\sqrt{25}=5$. Итак, ответ, выданный по программе и полученный при ручном просчете, совпадают, поэтому программу можно считать работоспособной. Если ответы не совпадают, это означает, что в программе допущены ошибки.

Чтобы исправить ошибку, в программе ставят счетчик адресов команд в нулевое положение клавишей [В/О] (если программа записана с адреса 00) и нажимают клавишу [ПП] («пошаговый проход» программы). При нажатии на клавишу [ПП] происходит выполнение одной команды программы. Ее результат высвечивается на индикаторе, при этом содержимое счетчика адресов команд увеличивается на единицу. На некотором шаге обнаружится ошибка, т. е. на индикаторе появится не то число, какое должно было бы появиться.

В месте, где найдена ошибка, переходят в режим «Программирование» ([F][ПРГ]). На индикаторе в левом углу высвечивается код последней выполненной команды.

Если код не совпадает с кодом, записанным в самой программе, то следует с помощью клавиши [←шг] вернуться на одну команду назад и нажать нужные клавиши, при этом в левом углу

появится правильный код команды. Если нужно просмотреть коды команд, которые записаны далее, используют клавишу [→шг], которая при каждом нажатии увеличивает счетчик адресов на единицу. Далее включают режим «Работа» ([F][АВТ]) и просматривают программу до конца с помощью клавиши [ПП]. Если обнаружена лишняя команда, ее можно стереть в программной памяти с помощью команды [К][НОП] («нет операции»). После просмотра всей программы до конца опять просчитывают по ней контрольный пример. Если ответ снова не совпал — значит программа составлена неправильно. В этом случае всю работу надо начинать сначала.

11. Команды перехода. Организация передачи управления

До сих пор мы рассматривали линейные программы, которые заносились в программную память с адреса 00. Вообще программу можно записывать, начиная с любого адреса. Для этого существует клавиша [БП] («безусловный переход»).

Задача 1. Написать программу нахождения суммы первых n членов арифметической прогрессии, если известен первый член a и разность d. Программу занести с адреса 17 (табл. 4).

Решение: Сумма первых n членов арифметической прогрессии вычисляется по формуле $S=\frac{2a+d(n-1)}{2}n$. Пусть $a\to$РП1, $d\to$РП2, $n\to$РП3.

Таблица 4

Адрес команды	Команда	Код команды
17	[2]	02
18	[П→x][1]	61
19	[×]	12
20	[П→x][3]	63
21	[1]	01
22	[—]	11
23	[П→x][2]	62
24	[×]	12
25	[+]	10
26	[2]	02
27	[÷]	13
28	[П→x][3]	63
29	[×]	12
30	[С/П]	50

Чтобы выйти на адрес 17, необходимо нажать клавиши [БП] и клавиши [1][7], указывающие номер адреса. Затем следует перейти в режим «Программирование» — ([F][ПРГ]). На индикаторе высветится в правом углу 17. Теперь можно заносить программу.

При составлении разветвляющихся и циклических программ необходимы команды безусловного и условного переходов. Команда безусловного перехода в программной памяти занимает две ячейки: в одной записывается команда [БП] (код 51), во второй адрес, на который необходимо перейти, или, как говорят программисты, передать управление — «адрес перехода».

Таблица 5

Адрес каманды	Команда	Код команды
47	БП	51
48	3 1	31

В таблице 5 приведен фрагмент программы, который реализует переход на выполнение команды, записанной по адресу 31.

На МК-61 с помощью команды условного перехода можно проверять следующие условия: $x<0$, $x\geqslant 0$, $x=0$, $x\neq 0$, т. е. сравнить с нулем числа, находящиеся в регистре X. Команда условного перехода в программной памяти занимает также две ячейки. В одной проверка условия [F] [x<0] ([F] [x⩾0], [F] [x=0], [F] [x≠0]), во второй — адрес перехода, если условие не выполняется.

Таблица 6 представляет собой фрагмент программы, содержащей команду условного перехода.

Таблица 6

Адрес команды	Команда	Код команды
24	F x⩾0	59
25	6 3	63
26	П→x 7	67
. . . .	. . . . .	. . . .
63	÷	13
. . . .	. . . . .	. . . .

По команде 24 число на индикаторе сравнивается с нулем, если оно больше либо равно нулю, то выполняется команда 26 и далее, если число на индикаторе меньше нуля, то выполняется команда перехода 25, которая указывает, что нужно перейти на выполнение команды 63, т. е. перейти к делению.

Задача 2. Написать программу вычисления значения

$$y=\begin{cases}\dfrac{\sqrt{x}-5x}{2}, & \text{если } x\geqslant 0,\\ 3x, & \text{если } x<0.\end{cases}$$

Решение. Схема алгоритма решения задачи приведена на рисунке 29. Пусть $x\to$РП1, ответ будет на индикаторе.

Таблица 7

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
00	П→x 1	61	07	—	11
01	F x⩾0	59	08	2	02
02	1 1	11	09	÷	13
03	F √	21	10	С/П	50
04	П→x 1	61	11	3	03
05	5	05	12	×	12
06	×	12	13	С/П	50

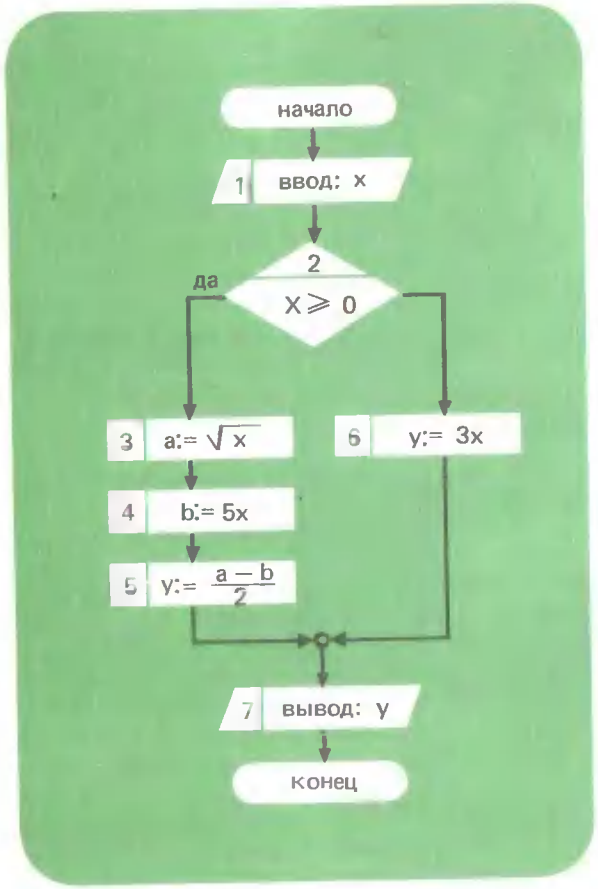


Рис. 29

При отладке данной программы (табл. 7) необходимо взять два числовых значения для x, одно, например, $4>0$, второе — $-6<0$, чтобы проверить программу по всем возможным путям.

Задача 3. Составить обучающую программу для проверки таблицы умножения.

Решение. Суть обучающей программы будет состоять в следующем. Ученик вводит в микрокалькулятор два числа $a\to$РП1, $b\to$РП2, а также подсчитанное в уме произведение $p=a\cdot b\to$РП3. Микрокалькулятор должен проверить, правильно ли ученик подсчитал в уме. Для этого произведение, полученное учеником, сравнивается с произведением, найденным микрокалькулятором. Если они равны, на индикаторе появляется оценка «5», если нет, то на индикаторе появляется сообщение «ошибка» (ЕГГОГ).

Из всего сказанного следует схема алгоритма (рис. 30) и программа (табл. 8) решения задачи.

Таблица 8

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
00	П→x 1	61	06	0 9	09
01	П→x 2	62	07	5	05
02	×	12	08	С/П	50
03	П→x 3	63	09	0	00
04	—	11	10	÷	13
05	F x=0	5E	11	С/П	50

Замечание. При составлении программ авторы преследуют цель объяснить, как составляются программы, поэтому программы в некоторых задачах могут быть составлены нерационально. Если учесть все возможности микрокалькулятора, то программы можно сократить.

12. Написание программ для разветвляющихся алгоритмов. Кодирование нечисловых ответов

Задача 1. Составить схему алгоритма и написать программу решения уравнения $ax=b$.

Решение. Схема алгоритма решения представлена на рисунке 31. Блоки 5 и 6 представляют собой не результат вычисления, а предложения «уравнение имеет бесчисленное множество решений» и «уравнение не имеет решений» соответственно. При составлении программ на микрокалькуляторах такие предложения зашифровывают некоторыми числами. Для этого введем переменную k, которая будет принимать значения

$$k=\begin{cases}1, \text{ если } x=\dfrac{b}{a},\\ 2, \text{ если уравнение}\\ \text{не имеет решений,}\\ 3, \text{ если } x \text{ — любое число.}\end{cases}$$

Пусть $a\to$РП1, $b\to$РП2, $k\to$РП3. Тогда программа примет вид:

Таблица 9

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
00	П→x 1	61	08	С/П	50
01	F x≠0	57	09	П→x 2	62
02	0 9	09	10	F x=0	5E
03	П→x 2	62	11	1 4	14
04	П→x 1	61	12	3	03
05	÷	13	13	С/П	50
06	x→П 4	44	14	2	02
07	1	01	15	С/П	50

Пусть имеем уравнение $0\cdot x=4\cdot 0\to$РП1, $4\to$РП2, В/О, С/П. На индикаторе появилось 2. Это означает, что уравнение не имеет решений.

Пусть имеем уравнение $0\cdot x=0$. $0\to$РП1, $0\to$РП2, В/О, С/П. На индикаторе появилось 3. Это означает, что x — любое число.

Пусть $5x=100$. $5\to$РП1, $100\to$РП2, В/О, С/П. На индикаторе появилось 1. Это означает, что значение x необходимо вызвать из РП4. Получаем $x=20$.

Задача 2. Составить схему алгоритма и написать программу вычисления значения функции $y=\sqrt{x-1}+\frac{1}{x-2}$.

Решение. Схема алгоритма приведена на рисунке 32.

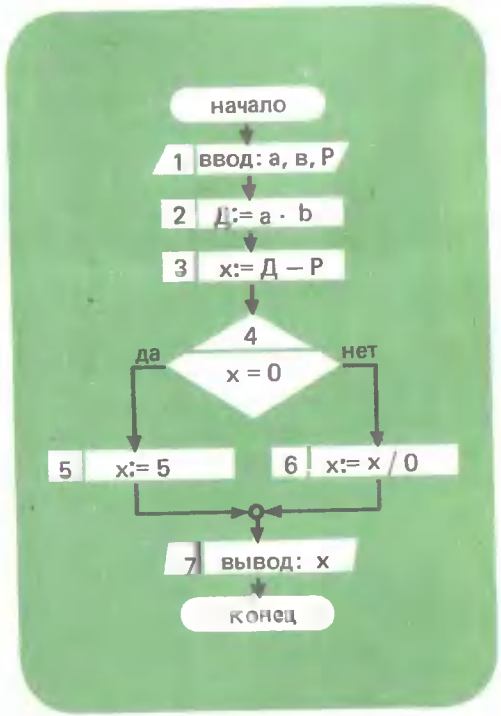


Рис. 30

Пусть $x\to$РП1. Предложение «функция не определена» зашифруем словом ЕГГОГ, которое высвечивается при выполнении некорректной операции. Введем в программу команды 0 F 1/x. Таким образом, получим следующую программу (табл. 10):

Таблица 10

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
00	П→x 1	61	09	F x≠0	57
01	1	01	10	1 4	14
02	—	11	11	F 1/x	23
03	F x≥0	59	12	+	10
04	1 4	14	13	С/П	50
05	F √	21	14	0	00
06	П→x 1	61	15	F 1/x	23
07	2	02	16	С/П	50
08	—	11			

Задача 3. Составить схему алгоритма и написать программу определения, каким днем недели было 12 августа 1983 г.

Решение. Введем обозначения: Д (день) = 12, М (месяц) = = 8, Г (год) = 1983. Тогда день недели определяется по формуле

$$К=\left(Д+[2{,}6М'+0{,}4]+\left[\frac{5Г'}{4}\right]+X\right)-$$
$$-7\left[\frac{Д+[2{,}6М'+0{,}4]+\left[\frac{5Г'}{4}\right]+X}{7}\right]$$

где $\left.\begin{matrix}М'=М+9\\Г'=Г-1\end{matrix}\right\}$, если $М<3$ и $\left.\begin{matrix}М'=М-3\\Г'=Г\end{matrix}\right\}$, если $М\geqslant 3$.

Для старого стиля, который был до 14 февраля 1918 г, $X=0$, для нового стиля $X=1$.

В ответе может получиться одно из чисел 0 — воскресенье, 1 — понедельник, 2 — вторник, 3 — среда, 4 — четверг, 5 — пятница, 6 — суббота. Схема алгоритма решения представлена на

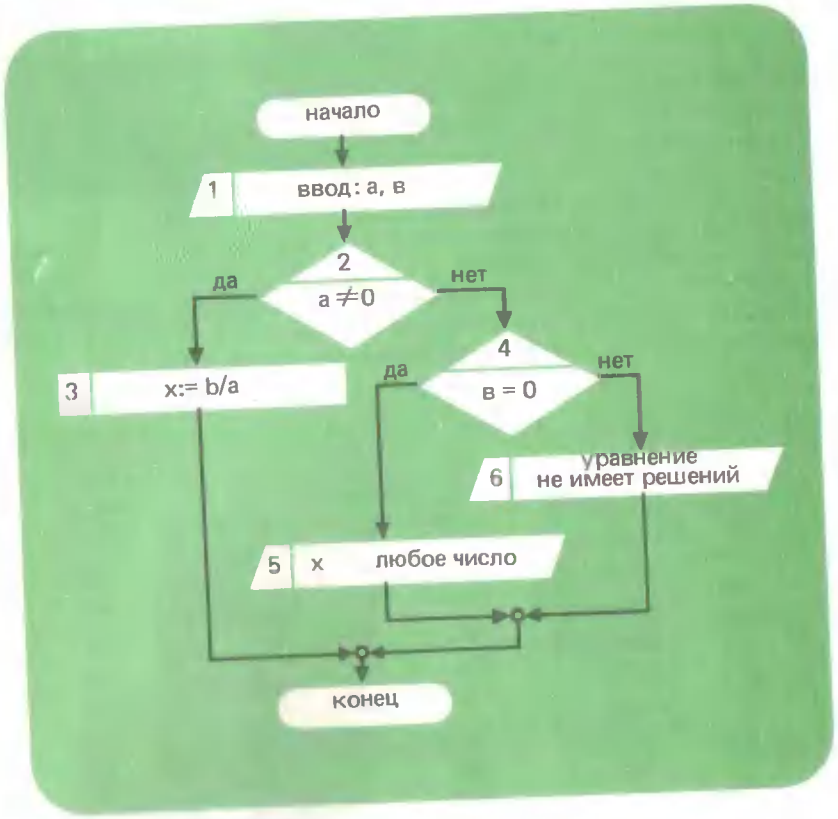


Рис. 31

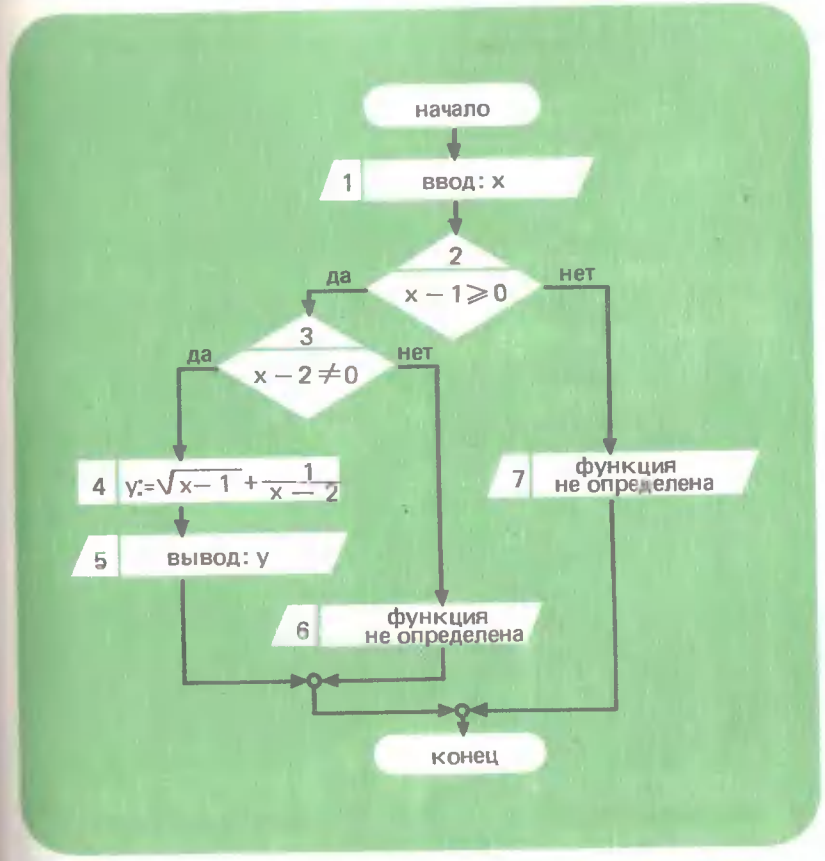


Рис. 32

рисунке 33, программа — в таблице 11. Распределим регистры памяти. Д→РП0, М→РП1, Г→РП2, Х→РП3, М′→РП4, Г′→РП5, А→РП6, В→РП7, С→РП8, 1918→РПа, 2,6→РПb, 0,4→РПс, К→индикатор.

Таблица 11

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
00	П→x a	6—	06	1 8	18
01	П→x 2	62	07	2	02
02	—	11	08	П→x 1	61
03	F x⩾0	59	09	—	11
04	2 2	22	10	F x⩾0	59
05	F x=0	5E	11	2 2	22

Продолжение

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
12	1	01	42	П→x 4	64
13	4	04	43	П→x b	61
14	П→x 0	60	44	×	12
15	—	11	45	П→x c	6[
16	F x⩾0	59	46	+	10
17	2 2	22	47	K [ ]	34
18	0	00	48	x→П 6	46
19	x→П 3	43	49	П→x 5	65
20	БП	51	50	5	05
21	2 4	24	51	×	12
22	1	01	52	4	04
23	x→П 3	43	53	÷	13
24	П→x 1	61	54	K [ ]	34
25	3	03	55	x→П 7	47
26	—	11	56	П→x 6	66
27	F x<0	5[	57	+	10
28	3 9	39	58	П→x 0	60
29	9	09	59	+	10
30	П→x 1	61	60	П→x 3	63
31	+	10	61	+	10
32	x→П 4	44	62	x→П 8	48
33	П→x 2	62	63	7	07
34	1	01	64	÷	13
35	—	11	65	K [ ]	34
36	x→П 5	45	66	7	07
37	БП	51	67	×	12
38	4 2	42	68	П→x 8	68
39	x→П 4	44	69	—	11
40	П→x 2	62	70	/—/	01
41	x→П 5	45	71	С/П	50

После выполнения программы на индикаторе появилось число 5, это значит, что 12 августа 1983 г.— пятница.

Задача 4. Составить схему алгоритма и написать программу игры «нуль или единица».

Решение. Играющий вводит число 0 или 1 на индикатор и запускает программу с помощью клавиш [В/О] [С/П]. Машина с помощью генератора случайных чисел выбирает число. Если

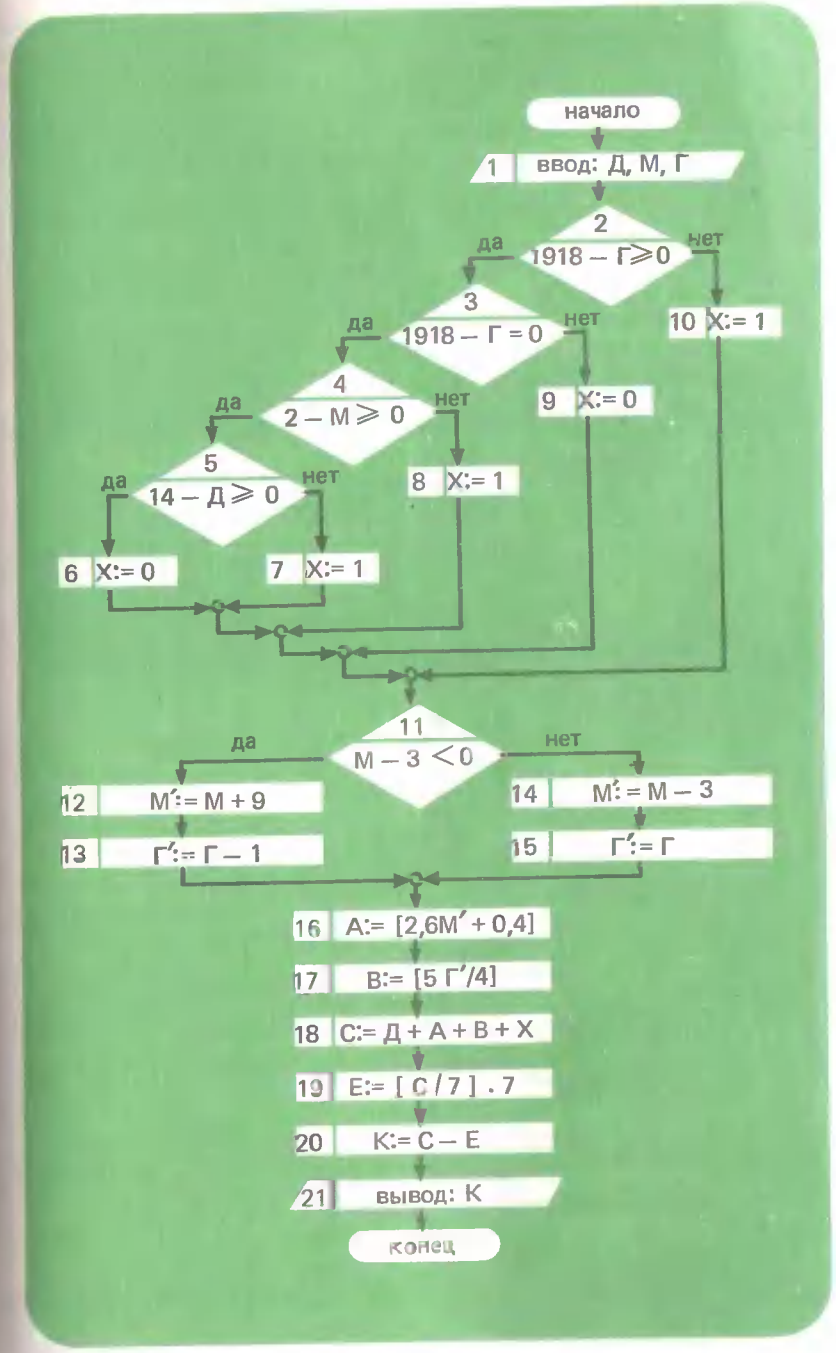


Рис. 33

оно меньше 0,5, то на индикаторе высвечивается 0. Если оно больше либо равно 0,5, то на индикаторе высвечивается 1. При этом если ответ машины совпал с числом, введенным играющим, то машине присуждается 1 очко, если не совпал,— 0 очков. Кроме этого, машина должна считать, сколько раз была проведена игра. Количество сыгранных партий заносим в РП0, а число выигранных машиной партий — в РП1.

Составляем схему алгоритма (рис. 34) и программу (табл. 12). Распределяем регистры памяти: x — число, вводимое играющим,→ индикатор, S→РП0 (число сыгранных партий), B→РП1

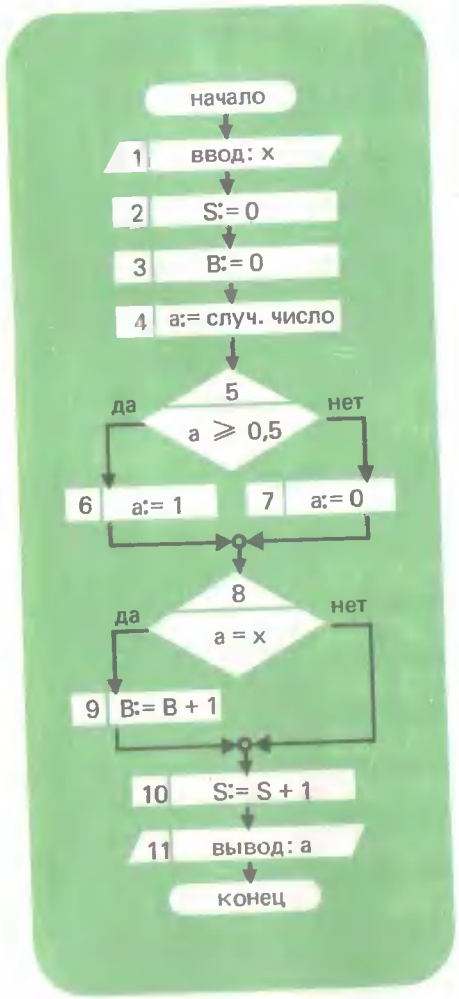


Рис. 34

Таблица 12

Адрес команды	Команда	Код команды
00	x→П 3	43
01	К С₄	3L
02	П→x 4	64
03	—	11
04	F x⩾0	59
05	·1 0	10
06	1	01
07	x→П 2	42
08	БП	51
09	1 2	12
10	0	00
11	x→П 2	42
12	П→x 3	63
13	—	11
14	F x=0	5E
15	2 0	20
16	1	01
17	П→x 1	61
18	+	10
19	x→П 1	41
20	1	01
21	П→x 0	60
22	+	10
23	x→П 0	40
24	П→x 2	62
25	С/П	50

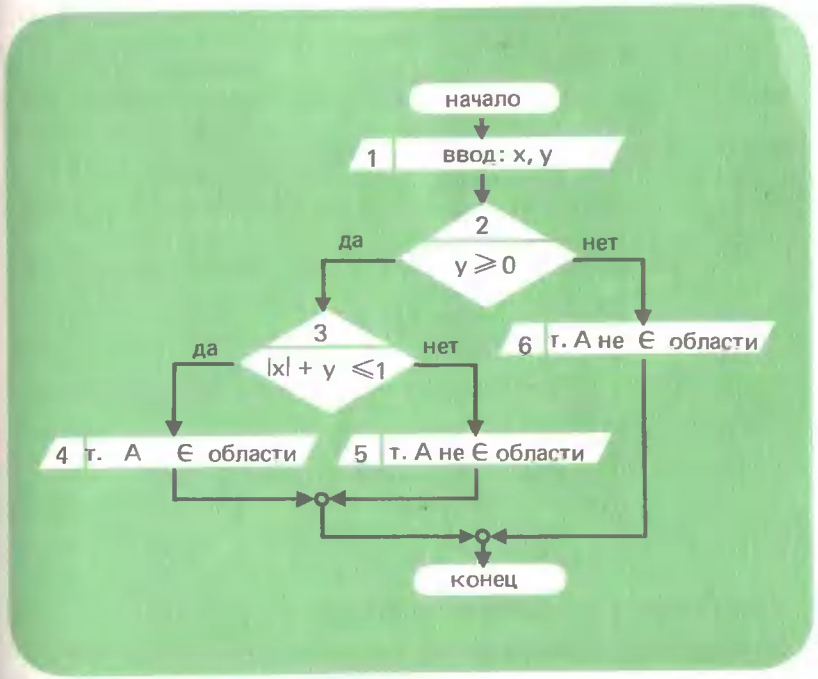


Рис. 35

(число выигранных машиной партий), a→РП2 (число полученное машиной), 0,5→РП4.

Перед началом игры необходимо очистить регистры 0, 1, 2, 3, а в РП4 поставить число 0,5.

Играющий вводит число в индикатор и пускает программу нажатием клавиши [В/О] [С/П], машина выдает на индикаторе свое число.

Играющий опять вводит в индикатор свое число и пускает программу.

Этот процесс повторяется несколько раз. Чтобы узнать, сколько раз была сыграна игра, надо нажать клавиши [П→x] [0], а чтобы узнать, сколько раз машина угадала число,— клавиши [П→x] [1].

Задача 5. Точка A задана координатами x и y. Определить, принадлежит ли точка A области, ограниченной линиями

$$\begin{cases} y \geqslant 0, \\ |x| + y \leqslant 1. \end{cases}$$

Решение: Схема алгоритма решения приведена на рисунке 35, программа в таблице 13. Пусть x→РП1, y→РП2.

Показания индикатора: $\begin{cases} 0, \text{ «Точка } A \text{ не принадлежит области»} \\ 1, \text{ «Точка } A \text{ принадлежит области»} \end{cases}$

Таблица 13

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
00	П→x 2	62	08	/—/	0L
01	F x≥0	59	09	F x≥0	59
02	1 3	13	10	1 3	13
03	П→x 1	61	11	1	01
04	K |x|	31	12	С/П	50
05	+	10	13	0	00
06	1	01	14	С/П	50
07	—	11			

Упражнения

1. Решить квадратное уравнение $ax^2+bx+c=0$.
2. Точка A задана координатами x и y. Определить принадлежит ли точка A внутренней области круга $x^2+y^2=R^2$.
3. Решить уравнение $ax^3+bx=0$.
4. Вычислить значения функции $y=\frac{2x^2+\sqrt{x^3+1}}{2}$.
5. Вычислить степень $y=a^x$, где x — целое число, $a\neq 0$;

$$a^x=\begin{cases}1, \text{ если } x=0,\\ a^x, \text{ если } x>0,\\ \frac{1}{a^{-x}}, \text{ если } x<0.\end{cases}$$

13. Составление программ для циклических алгоритмов

При разработке циклических схем мы отмечали, что имеются базовые структуры цикл-ПОКА и цикл-ДО.

Цикл-ПОКА при написании программ реализуется с помощью команд условного и безусловного переходов.

Схема цикла-ПОКА (см. рис. 6) реализуется следующим фрагментом программы (табл. 14):

Таблица 14

Адрес команды	Команда	Адрес команды	Команда
.	.	20	серия
.	.	....	.....
.	.	25	серия
17	проверка условия	26	безусловный переход
18	адрес команды, идущей после безусловного перехода	27	на проверку условия, т. е. команду 17.
		28	.....
19	серия	....	.....

Задача 1. Написать программу вычисления значения функции $y=1-\frac{1}{2}+\frac{1}{3}-\frac{1}{4}+\frac{1}{5}-\frac{1}{6}+\ldots$. Суммирование продолжать до тех пор, пока очередное слагаемое по абсолютной величине не станет меньше заданного числа h.

Решение. Схема алгоритма решения этой задачи нами была составлена ранее (см. рис. 16), воспользуемся ею и составим программу (табл. 15). Пусть $h\to$РП1, $y\to$РП2, $n\to$РП3, $a\to$РП4.

Таблица 15

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
00	0	00	16	1	01
01	x→П 2	42	17	+	10
02	1	01	18	x→П 3	43
03	x→П 3	43	19	1	01
04	1	44	20	—	11
05	x→П 4	44	21	П→x 4	64
06	K |x|	31	22	/—/	0L
07	П→x 1	61	23	×	12
08	—	11	24	П→x 3	63
09	F x≥0	59	25	÷	13
10	2 9	29	26	x→П 4	44
11	П→x 4	64	27	БП	51
12	П→x 2	62	28	0 6	06
13	+	10	29	П→x 2	62
14	x→П 2	42	30	С/П	50
15	П→x 3	63			

Программа запускается нажатием клавиш В/О С/П, при предварительной засылке значения $h\to$РП1.

Задача 2. Написать программу нахождения наибольшего общего делителя (НОД) двух чисел N и M.

Решение. Схема алгоритма решения этой задачи составлена ранее (см. рис. 15), напишем программу (табл. 16).

Пусть $N\to$РП1, $M\to$РП2, $x\to$РП3, $y\to$РП4.

Значения N и M заносим в регистры памяти 1 и 2 соответственно и пускаем программу клавишами В/О С/П.

Цикл-ДО (схему см. на рис. 7) применяется в тех случаях, когда известно число повторений цикла, выраженное некоторым натуральным числом. Для реализации этой базовой структуры в микрокалькуляторе имеется специальная команда, которая называется «командой организации цикла». В программной памяти

Таблица 16

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
00	П→x 1	61	10	1 6	15
01	x→П 3	43	11	/—/	0L
02	П→x 2	62	12	x→П 3	43
03	x→П 4	44	13	БП	51
04	П→x 4	64	14	0 4	04
05	П→x 3	63	15	x→П 4	44
06	—	11	16	БП	51
07	F x≠0	57	17	0 4	04
08	1 8	18	18	П→x 3	63
09	F x<0	5[	19	С/П	50

эта команда занимает две ячейки. Первая ячейка: F L0 (F L1, F L2, F L3); вторая ячейка: адрес перехода. При организации цикла с помощью этой команды переменная, которая задает число повторений цикла, должна находиться в РП0, если в команде есть клавиши F L0 (либо в РП1 для F L1, в РП2 для F L2, в РП3 для F L3). При выполнении команды организации цикла происходит обращение к регистру, номер которого входит в команду. При каждом обращении из содержимого этого регистра вычитается 1, новое содержимое этого регистра сравнивается с нулем. Если содержимое регистра не равно нулю, то выполняется переход на адрес, записанный во второй ячейке команды. Если содержимое регистра равно нулю, то выполняется следующая по программе команда, например, как это сделано в таблице 17.

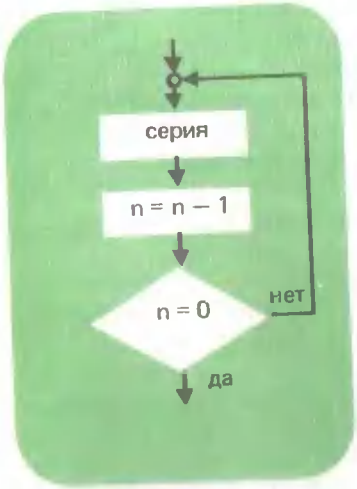


Рис. 36

Таблица 17

22	F	L0
23	1	6
24	П→x	2
. . . .	. . . . .	

При выполнении этого фрагмента программы произойдет следующее: из содержимого РП0 вычитается 1, получается некоторое число. Это число сравнивают с нулем. Если оно не равно нулю, то произойдет выполне-

Таблица 18

Адрес команды	Команда	Код команды
00	0	00
01	x→П 1	41
02	П→x 0	60
03	x→П 2	42
04	П→x 2	62
05	F x^2	22
06	4	04
07	×	12
08	П→x 2	62
09	1	01
10	5	05
11	×	12
12	6	06
13	—	11
14	÷	13
15	П→x 1	61
16	+	10
17	x→П 1	41
18	F L2	58
19	0 4	04
20	П→x 1	61
21	С/П	50

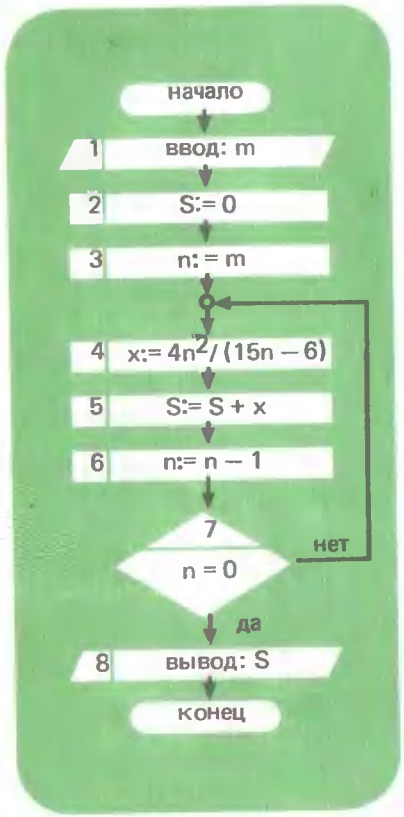


Рис. 37

ние 23-й строки, т. е. переход на команду по адресу 16. Если число равно нулю, то будет выполняться команда по адресу 24 и далее по программе.

Из всего сказанного следует, что схему базовой структуры цикла-ДО применительно к микрокалькулятору можно изобразить следующим образом (рис. 36).

Задача 3. Написать программу нахождения суммы первых m членов последовательности $x_n = \frac{4n^2}{15n-6}$.

Решение. Схема алгоритма решения задачи была составлена ранее (см. рис. 13). Однако, учитывая действие команды организации цикла, схему следует переделать (рис. 37).

Программа приведена в таблице 18.

Распределим регистры памяти: $m \to$ РП0, $S \to$ РП1, $n \to$ РП2, $x \to$ РП3.

Перед пуском программы значение m засылаем в РП0.

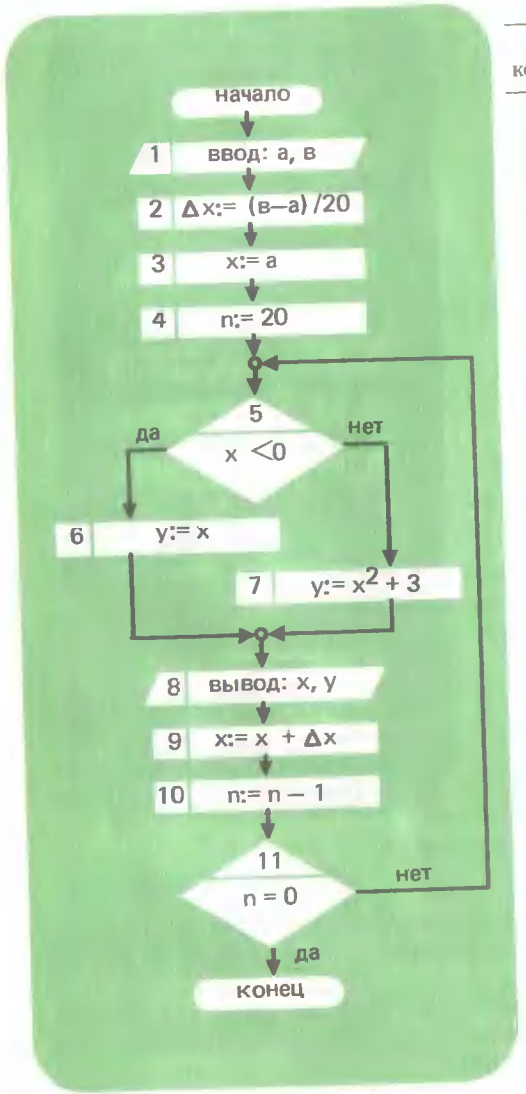


Рис. 38

Таблица 19

Адрес команды	Команда	Код команды
00	П→x 2	62
01	П→x 1	61
02	—	11
03	2	02
04	0	00
05	÷	13
06	x→П 4	44
07	П→x 1	61
08	x→П 3	43
09	2	02
10	0	00
11	x→П 0	40
12	П→x 3	63
13	F x<0	5[
14	1 7	17
15	БП	51
16	2 0	20
17	F x^2	22
18	3	03
19	+	10
20	С/П	50
21	П→x 3	63
22	П→x 4	64
23	+	10
24	x→П 3	43
25	F L0	5Г
26	1 3	13
27	0	00
28	F 1/x	23
29	С/П	50

Задача 4. Написать программу вычисления 20 значений функции

$$y=\begin{cases} x^2+3, & \text{если } x\geqslant 0,\\ x, & \text{если } x<0,\end{cases} \quad \text{причем } x\in[a,\ b].$$

Решение. Схема алгоритма решения представлена на рисунке 38. Распределим регистры памяти: $a\to$РП1, $b\to$РП2, $n\to$РП0, $x\to$РП3, $\Delta x\to$РП4.

Перед пуском программы (табл. 19) необходимо заслать числовые значения $a\to$РП1, $b\to$РП2. Нажать клавиши [В/О], [С/П]. После останова записать значение и пустить программу далее на выполнение [С/П]. После получения 20 значений на индикаторе высветится ЕГГОГ, что означает «конец вычислений».

Задача 5. Написать программу нахождения натуральных чисел x и y, удовлетворяющих следующим условиям:

$$\begin{cases} ax+by=c,\\ 0<x\leqslant b,\\ 0<y\leqslant a,\\ a,\ b,\ c \text{ — натуральные числа.}\end{cases}$$

Решение. Из уравнения $ax+by=c$ выразим x: $x=(c-by)/a$. Подставляя вместо y числа 1, 2, ... и проверяя, является ли x целым числом, а также удовлетворяют ли x, y заданным условиям ($0<x\leqslant b$, $0<y\leqslant a$), найдем решение. Для получения структурированной схемы введем переменную k:

$$k=\begin{cases} 1, & \text{нет решений,}\\ 2, & \text{решение найдено,}\\ 0, & \text{необходимо продолжать поиск решения.}\end{cases}$$

Схема алгоритма решения задачи приведена на рисунке 39, программа в таблице 20. Распределим регистры памяти: $a\to$РПа, $b\to$РПb, $c\to$РПc, $x\to$РП1, $y\to$РП2, $d\to$РПd, $k\to$РП0.

Таблица 20

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
00	0	00	16	—	11
01	x→П 0	40	17	/—/	0L
02	1	01	18	П→x a	6—
03	x→П 2	42	19	÷	13
04	П→x 0	60	20	x→П 1	41
05	F x=0	5E	21	K [x]	34
06	4 4	44	22	П→x 1	61
07	П→x a	6—	23	—	11
08	П→x 2	62	24	F x=0	5E
09	—	11	25	3 9	39
10	F x≥0	59	26	П→x b	6L
11	3 5	35	27	П→x 1	61
12	П→x 2	62	28	—	11
13	П→x b	6L	29	F x≥0	59
14	×	12	30	3 5	35
15	П→x c	6[	31	2	02

Продолжение

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
32	x→П 0	40	44	0 4	04
33	БП	51	45	П→x 0	60
34	0 4	04	46	1	01
35	1	01	47	—	11
36	x→П 0	40	48	F x≠0	57
37	БП	51	49	5 3	53
38	0 4	04	50	П→x 2	62
39	П→x 2	62	51	П→x 1	61
40	1	01	52	С/П	50
41	+	10	53	0	00
42	x→П 2	42	54	F 1/x	23
43	БП	51	55	С/П	50

Если на индикаторе высветилось число, то это значение x. Чтобы получить значение y, надо нажать клавишу [↔]. Если на индикаторе высветилось ЕГГОГ, это означает, что не существует таких x и y, которые удовлетворяют данным условиям.

Перед пуском программы на выполнение необходимо заслать численные значения a, b, c соответственно в регистры памяти a, b, c.

Упражнения

1. Написать программу, определяющую, является ли число c членом геометрической прогрессии, заданной первым членом b_1 и знаменателем q.

2. Написать программу угадывания дня и месяца, если задано число $A=12x+31y$, где x — день, y — месяц.

3. Написать программу, определяющую, является ли число членом арифметической прогрессии, если она задана вторым a_2 и четвертым a_4 членами.

14. Подпрограммы и обращение к ним

В процессе составления программы может появиться необходимость проведения для разных ее участков однотипных вычислений с различными исходными данными. В этом случае удобно записывать соответствующую группу команд в программе только один раз, а затем использовать их для тех участков программы, где необходимы вычисления, задаваемые данной группой команд. Такой подход реализуется путем использования подпрограмм.

Подпрограмма — самостоятельная программная единица. Она составляется так же, как и обычная программа, но заканчивается операцией В/О (возврат из подпрограммы).

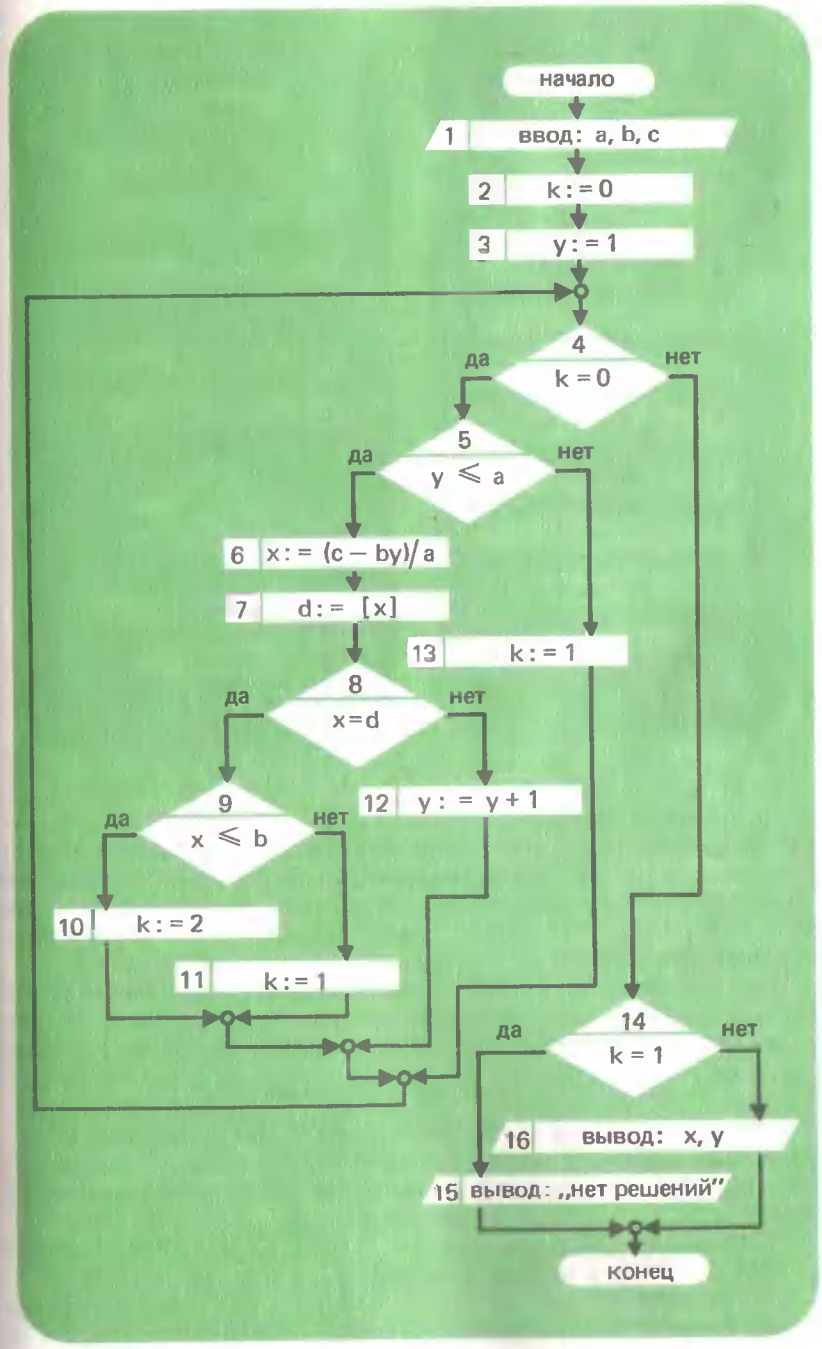


Рис. 39

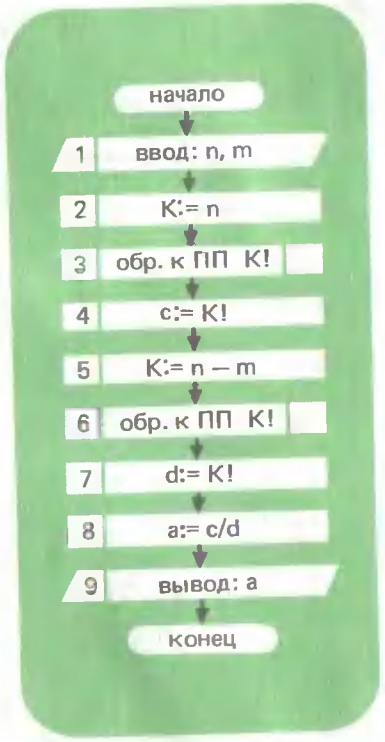


Рис. 40

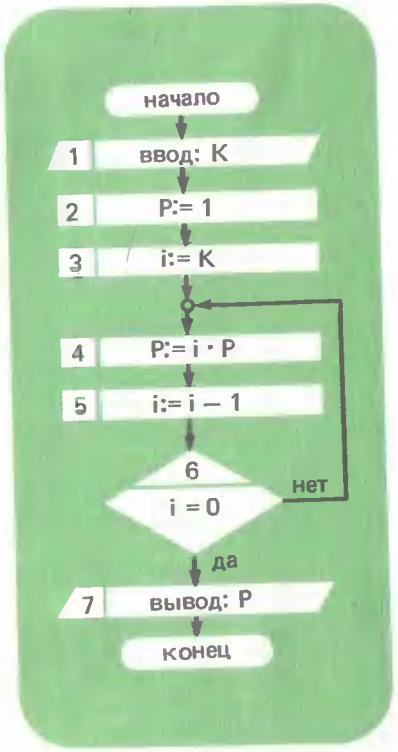


Рис. 41

В тех участках основной программы, где требуются результаты вычислений по подпрограммам, пишут команду обращения к подпрограмме. Она занимает две ячейки: первая — ПП (переход к подпрограмме), вторая указывает адрес, с которого начинается подпрограмма.

При составлении общей схемы алгоритма решения задачи схема алгоритма подпрограммы составляется отдельно.

Задача 1. Написать программу вычисления числа размещений из n элементов по m $A_n^m = \frac{n!}{(n-m)!} = \frac{1 \cdot 2 \cdot 3 \cdot \ldots \cdot n}{1 \cdot 2 \cdot 3 \cdot \ldots \cdot (n-m)}$.

Решение. Вычисление факториалов предусмотрим в подпрограмме. Тогда основная схема алгоритма будет очень простой (рис. 40). Схема алгоритма подпрограммы показана на рисунке 41.

Распределим регистры памяти: $n \to$ РПа, $m \to$ РПb, $k \to$ РП0, $p \to$ РП1, $i \to$ РП2, $c \to$ РПc, $d \to$ РПd.

Составление программы можно начинать как с основной программы, так и с подпрограммы. Данную программу начнем с основной (табл. 21).

Таблица 21

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
00	П→x a	6—	15	x→П 1	41
01	П→x b	6L	16	П→x 0	60
02	—	11	17	x→П 2	42
03	x→П 0	40	18	П→x 2	62
04	ПП	53	19	П→x 1	61
05	1 4	14	20	×	12
06	x→П d	4Г	21	x→П 1	41
07	П→x a	6—	22	П→x 2	62
08	x→П 0	40	23	1	01
09	ПП	53	24	—	11
10	1 4	14	25	F L2	58
11	П→x d	6Г	26	1 8	18
12	÷	13	27	П→x 1	61
13	С/П	50	28	В/О	52
14	1	01			

Для пуска программы необходимо ввести численные значения $n \to$ РПа, $m \to$ РПb и очистить остальные регистры памяти 0, 1, 2, c, d. После просчета на индикаторе высвечивается результат.

Задача 2. Написать программу нахождения НОД трех чисел.

Решение. Мы уже составляли программу нахождения НОД двух чисел. Поэтому целесообразно воспользоваться ею как подпрограммой. Тогда основная схема примет вид, показанный на рисунке 42.

Из программы, составленной ранее для НОД двух чисел, нам необходимы команды с адреса 04 по адрес 19. Чтобы подпрограмму не менять, распределим регистры памяти, учитывая, что $x \to$ РП3, $y \to$ РП4. Тогда $a \to$ РПа, b→РПb, $c \to$ РПc. Напишем всю программу, начиная с адреса 04, с которого идет подпрограмма, изменив команду С/П по адресу 19 на команду В/О (табл. 22).

Перед пуском программы необходимо заслать численные значения a, b, c. Поскольку основная программа начинается с адреса 20, то и пуск программы на счет необходимо начинать с адреса 20, т. е. БП 20 С/П. После выполнения программы результат высветится на индикаторе.

Задача 3. Заданы две точки $A(x_1, y_1)$, $B(x_2, y_2)$. Написать программу вычисления полярных координат этих точек: $\rho = \sqrt{x^2 + y^2}$, $\varphi = \operatorname{arctg} \frac{y}{x}$.

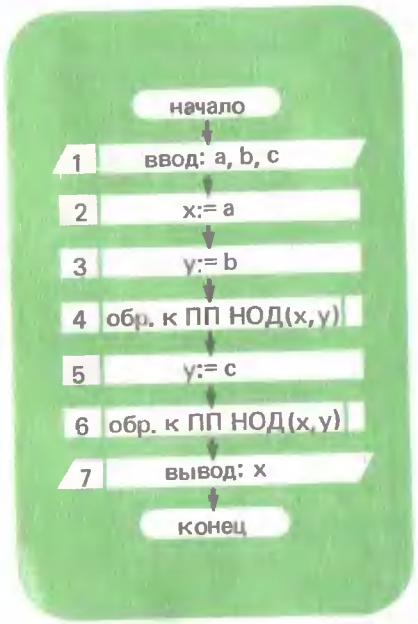


Рис. 42

Рис. 43

Таблица 22

Адрес команды	Команда	Код команды
04	П→x 4	64
05	П→x 3	63
06	—	11
07	F x≠0	57
08	1 8	18
09	F x<0	5[
10	1 5	15
11	/—/	0L
12	x→П 3	43
13	БП	51
14	0 4	04
15	x→П 4	44
16	БП	51
17	0 4	04
18	П→x 3	63
19	В/О	52
20	П→x a	6—
21	x→П 3	43
22	П→x b	6L
23	x→П 4	44
24	ПП	53
25	0 4	04
26	П→x C	6[
27	x→П 4	44
28	ПП	53
29	0 4	04
30	П→x 3	63
31	С/П	50

Таблица 23

Адрес команды	Команда	Код команды
00	П→x 4	64
01	x→П 0	40
02	П→x 5	65
03	x→П 1	41
04	ПП	53
05	2 1	21

Адрес команды	Команда	Код команды
06	П→x 2	62
07	x→П a	4—
08	П→x 3	63
09	x→П b	41
10	П→x 6	66
11	x→П 0	40
12	П→x 7	67
13	x→П 1	41
14	ПП	53
15	2 1	21
16	П→x 2	62
17	x→П c	4[
18	П→x 3	63
19	x→П d	4Г
20	С/П	50
21	П→x 0	60
22	F x^2	22
23	П→x 1	61
24	F x^2	22
25	+	10
26	F √	21
27	x→П 2	42
28	П→x 1	61
29	П→x 0	60
30	÷	13
31	F tg^{-1}	1L
32	x→П 3	43
33	В/0	52

Рис. 44

Решение: Вычисление полярных координат предусмотрим в подпрограмме. Схема подпрограммы изображена на рисунке 43. Схема основной программы дана на рисунке 44. Распределим регистры памяти: $x \to$ РП0, $y \to$ РП1, $\rho \to$ РП2, $\varphi \to$ РП3, $x_1 \to$ РП4, $y_1 \to$ РП5, $x_2 \to$ РП6, $y^2 \to$ РП7, $a \to$ РПa, $b \to$ РПb, $c \to$ РПc, $d \to$ РПd. Программа приведена в таблице 23.

Перед пуском программы (табл. 23) координаты точек A и B необходимо послать в 4, 5, 6, 7 РП. Пуск программы осуществляется с помощью клавиши В/О С/П. Ответ ($\rho \to$ РПa, $\varphi_1 \to$ РПb, $\rho_2 \to$ РПc, $\varphi_2 \to$ РПd) необходимо вызвать из соответствующих регистров памяти.

15. Организация косвенных переходов. Косвенный вызов и засылка чисел в регистры памяти

Рассмотрим команды косвенного вызова и косвенной засылки. Для команды косвенного вызова необходимо нажать три клавиши: [К] [П→x] и клавишу, обозначающую номер регистра памяти. При этом если номер регистра памяти от 0 до 3, то при косвенном вызове содержимое регистра памяти уменьшается на единицу.

Если номер регистра памяти от 4 до 6, то по команде косвенного вызова содержимое регистра памяти увеличивается на единицу.

Если номер регистра памяти 7, 8, 9, a, b, c, d, e, то по команде косвенного вызова содержимое регистра памяти остается без изменения. Изменение числа при команде косвенного вызова является модификацией числа.

Если в адресуемом регистре памяти находится целое число, не превосходящее 15, то по команде косвенного вызова число в регистре памяти модифицируется. На индикатор вызывается число из регистра памяти, номер которого равен модифицированному числу.

При команде косвенного вызова микрокалькулятор числа 10, 11, 12, 13, 14 воспринимает как номера регистров памяти a, b, c, d, e.

Аналогично работает команда косвенной засылки чисел в регистры памяти. Команда косвенной засылки реализуется клавишами [К] [x→П] [N], где [N] — номер адресуемого регистра памяти. По команде [К] [x→П] [N] происходит модификация содержимого регистра N и засылка числа с индикатора по модифицированному адресу.

Команды косвенного вызова и косвенной засылки дают возможность писать довольно короткие программы при работе с массивами.

Задача 1. Написать программу суммирования чисел, образующих массив $x=\{7{,}2;\ 5{,}1;\ 3{,}4;\ 15{,}2;\ 7{,}5;\ 10{,}9;\ 14{,}2;\ 29{,}1;\ 18{,}9;\ 17{,}1;\ 20{,}4\}$.

Решение. Распределим эти числа по регистрам памяти: 7,2→РП1, 5,1→РП2, 3,4→РП3, 15,2→РП4, 7,5→РП5, 10,9→→РП6, 14,2→РП7, 29,1→РП8, 18,9→РП9, 17,1→РПa, 20,4→→РПb. В РП0 поместим число 12 и используем его для косвенного вызова чисел. В РП0 поместили число 12, так как в массиве 11 чисел, и последнее число массива находится в РПb, номер которого 11. При первом выполнении команды [К] [П→x] [0] число 12 в РП0 модифицируется (12−1=11) и произойдет вызов числа из РП11 на индикатор. При повторном выполнении команды [К] [П→x] [0] в РП0 будет модифицироваться уже число

Таблица 25

Адрес команды	Команда	Код команды
00	[0]	00
01	[x→П] [e]	4E
02	[1]	01
03	[2]	02
04	[x→П] [0]	40
05	[П→x] [e]	6E
06	[К] [П→x] [0]	Г0
07	[+]	10
08	[x→П] [e]	4E
09	[П→x] [0]	60
10	[F] [x=0]	5E
11	[0] [5]	05
12	[П→x] [e]	6E
13	[С/П]	50

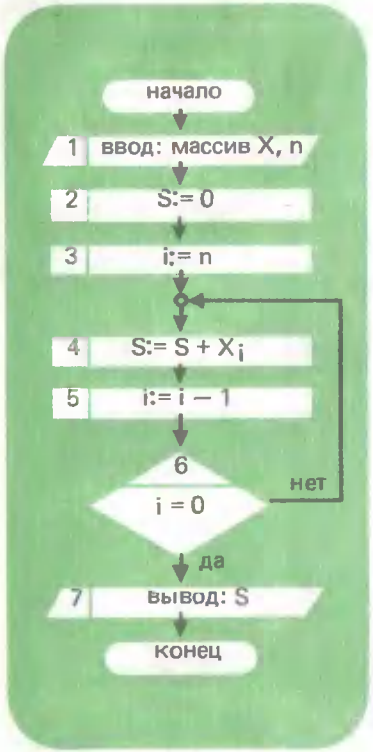


Рис. 45

11(11−1=10) и произойдет вызов числа из РП10 (РПa) на индикатор, и так до тех пор, пока в РП0 не станет число 0. Сумму чисел будем накапливать в РПe, поэтому первоначально зашлем туда число 0. Схема алгоритма представлена на рисунке 45, программа — в таблице 25.

На индикаторе читаем ответ:149.

Задача 2. Написать программу нахождения наибольшего элемента массива $y=\{3{,}7;\ 10{,}5;\ 14{,}2;\ 9{,}8;\ 11{,}6;\ 27{,}1;\ 21{,}3;\ 25{,}2;\ 17{,}4;\ 8{,}9\}$.

Решение. Распределим элементы массива по регистрам памяти: 3,7→РП1, 10,5→РП2, 14,2→РП3, 9,8→РП4, 11,6→РП5, 27,1→РП6, 21,3→РП7, 25,2→РП8, 17,4→РП9, 8,9→РПa. Всего элементов в массиве 10. Элементы в массиве будем сравнивать попарно, начиная с последнего, а затем наибольшее из двух чисел будем сравнивать со следующим числом, и так до тех пор, пока не переберем весь массив. Сравнение удобно начинать с последнего элемента, так как, послав в РП0 число, равное количеству элементов в массиве, и обращаясь к команде косвенного вызова, мы можем вызывать последовательно числа из всех необходимых регистров памяти. Итак, 10→РП0.

Схема алгоритма приведена на рисунке 46, а программа — в таблице 26.

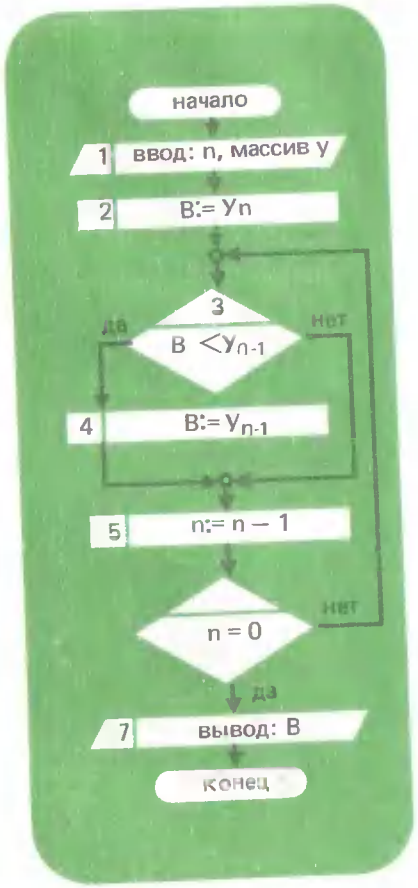


Рис. 46

Таблица 26

Адрес команды	Команда	Код команды
00	П→x a	6—
01	x→П b	4L
02	П→x b	6L
03	K П→x 0	Г0
04	K max	36
05	x→П b	4L
06	П→x 0	60
07	1	01
08	—	11
09	F x=0	5E
10	0 2	02
11	П→x b	6L
12	С/П	50

На индикаторе высветится ответ: 27,1.

Упражнения

1. Составить программу упорядочения элементов массива по возрастанию.

2. Написать программу подсчета числа положительных чисел массива.

3. Написать программу замены отрицательных чисел массива нулями.

4. Написать программу подсчета числа элементов массива, удовлетворяющих условию $A \leqslant x \leqslant B$.

16. Команды косвенного безусловного и условного переходов

В микрокалькуляторе «Электроника МК-61», кроме прямых безусловных и условных переходов, возможны и косвенные безусловные и условные переходы.

Команда косвенного безусловного перехода выполняется клавишами [K] [БП] [N], где N — номер регистра памяти. Эта команда в программной памяти задается одним адресом. При исполнении этой команды происходит модификация числа, хранящегося в регистре N, и переход на команду, адрес которой равен модифицированному числу в регистре N. Модификация чисел происходит точно так же, как и при косвенной записи и косвенном вызове чисел из регистров памяти. Так, если N равно 0, 1, 2, 3, то из числа, находящегося в РПN, вычитается 1, если N=4, 5, 6, то к числу прибавляется 1, если N=7, 8, 9, a, b, c, d, e, то содержимое регистра N не изменяется.

Команда косвенного условного перехода выполняется клавишами [K] [x⊗0] [N], где [x⊗0] — это одна из клавиш [x<0], [x=0], [x⩾0], [x≠0], а [N] — номер регистра памяти. В программной памяти эта команда занимает одну ячейку.

При выполнении этой команды происходит проверка содержимого регистра X на выполнение заданного условия. Если условие **не выполняется**, то происходит модификация числа, хранящегося в регистре N, и осуществляется переход к выполнению команды, адрес которой равен модифицированному числу. Если условие **выполняется**, то осуществляется выполнение команды, записанной после команды косвенного условного перехода, при этом модификация числа в регистре N не происходит.

Задача 1. Вычислить значение функции

$$y=\begin{cases} 5x^2+7|x|-10, & \text{если } x<-1, \\ \dfrac{|x|+4}{x-3}, & \text{если } -1\leqslant x<1, \\ x^2+2x+3, & \text{если } x\geqslant 1. \end{cases}$$

Решение. Составим схему алгоритма решения задачи (рис. 47).

В схеме есть два блока проверки условий: $a<0$ и $a\geqslant 0$. Эти условия в программе запишем с помощью команд косвенного условного перехода. Для проверки условия $a<0$ используем команду [K] [x<0] [1]. Здесь взят первый регистр памяти, значит, предварительно в первый регистр памяти необходимо заслать число на единицу большее, чем адрес команды, выполняемой по стрелке «нет».

Для проверки условия $a\geqslant 0$ воспользуемся командой [K] [x⩾0] [4]. Здесь задействован РП4, поэтому предварительно в него нужно послать число на единицу меньшее, чем адрес команды, выполняемой после проверки этого условия по стрелке «нет».

Пока мы не знаем адреса команд, которые необходимо заслать в РП1 и в РП4. Это мы сделаем после написания программы (таблица 27). Пусть $x\to$РПе.

Таблица 27

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
00	П→x e	6E	20	К x≥0 4	94
01	1	01	21	П→x e	6E
02	+	10	22	F x^2	22
03	К x<0 1	11	23	П→x e	6E
04	П→x e	6E	24	2	02
05	F x^2	22	25	×	12
06	5	05	26	3	03
07	×	12	27	+	10
08	П→x e	6E	28	+	10
09	К |x|	31	29	С/П	50
10	7	07	30	П→x e	6E
11	×	12	31	К |x|	31
12	1	01	32	4	04
13	0	00	33	+	10
14	—	11	34	П→x e	6E
15	+	10	35	3	03
16	С/П	50	36	—	11
17	П→x e	6E	37	÷	13
18	1	01	38	С/П	50
19	—	11			

Итак, если условие $a<0$ не выполняется, то осуществляется переход на адрес 17. Отсюда следует, что в ПР1, который задействован в команде [К] [x<4] [1], необходимо записать число 18, поскольку при модификации оно уменьшится на единицу.

Если условие $x\geqslant 0$ не выполняется, то нужно перейти на адрес 30. В команде [К] [x≥0] [4] задействован РП4, в который запишем число 29, так как при модификации это число увеличится на единицу.

Итак: $x\to$РПe, 18→РП1, 29→РП4.

Задача. 2. В прямоугольной системе координат задана точка $A(x, y)$. Написать программу определения номера четверти, которой принадлежит точка A. Будем считать, что

$$n=\begin{cases}1, \text{ если } x\geqslant 0,\ y\geqslant 0;\\ 2, \text{ если } x<0,\ y\geqslant 0;\\ 3, \text{ если } x<0,\ y<0;\\ 4, \text{ если } x\geqslant 0,\ y<0.\end{cases}$$

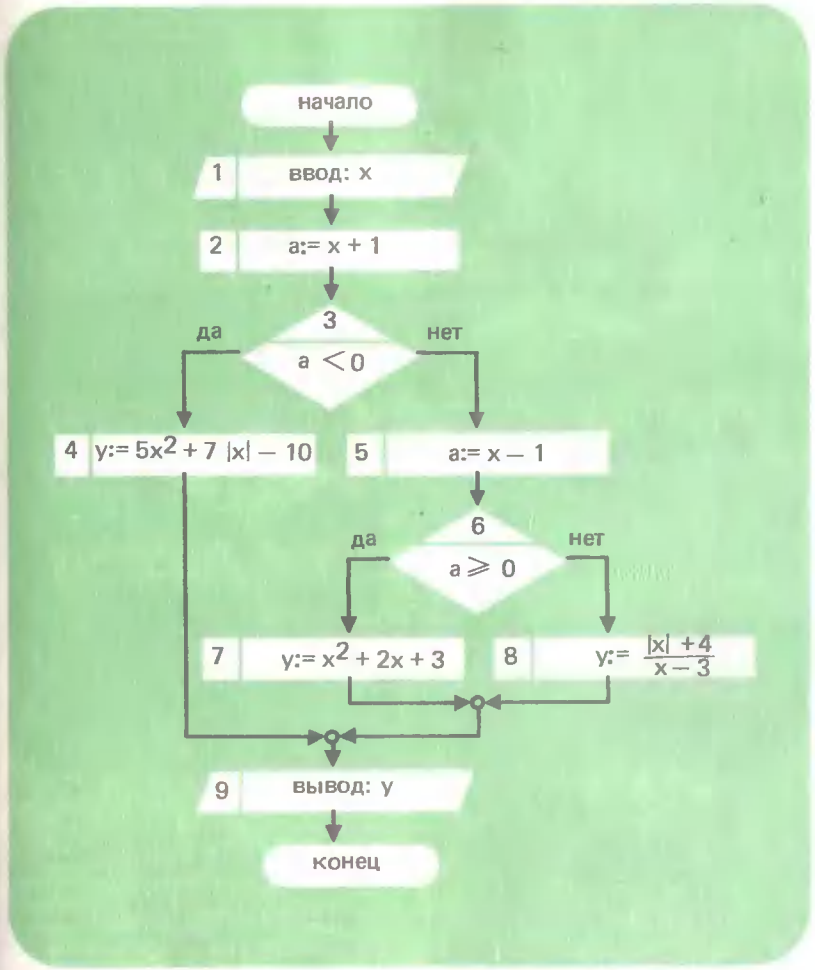


Рис. 47

Решение. Запишем программу, используя команды косвенных условных переходов, начиная с адреса 09. Пусть $x\to$РПe, $y\to$РПd, схема алгоритма решения изображена на рисунке 48.

Из текста программы (табл. 28) следует, что в РП1 надо занести число 18, так как при $x\geqslant 0$ необходимо перейти на адрес 17; в РП2 — число 16, так как при $y\geqslant 0$ необходимо перейти на адрес 15; в РП3 — число 22, так как при $y<0$ необходимо перейти на адрес 21. Это можно сделать вручную, но лучше, если это будет делать сам микрокалькулятор. Поэтому дополним программу командами, которые запишем, начиная с адреса 00 (табл. 29).

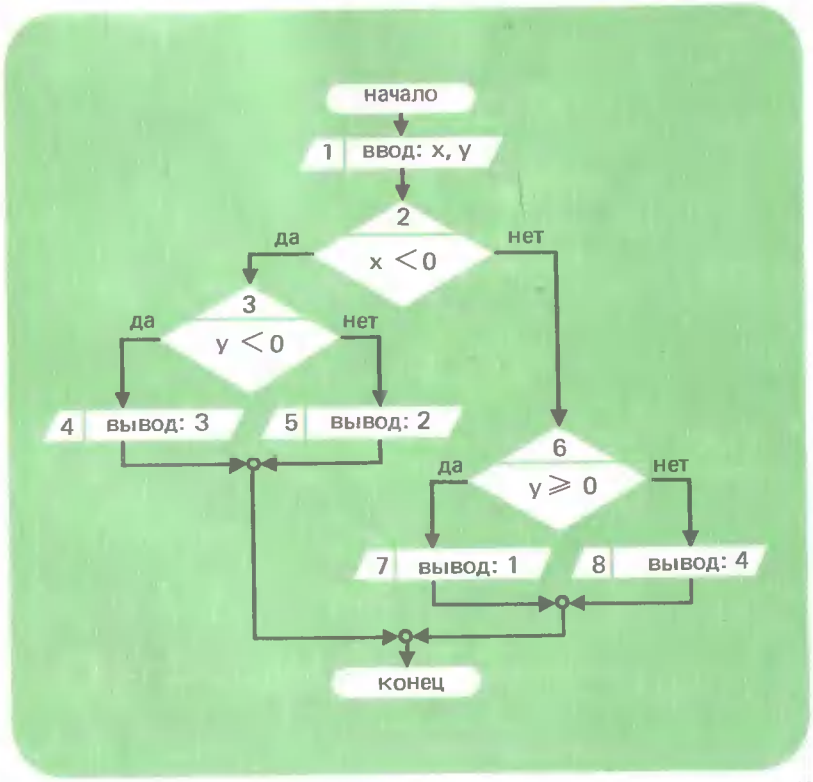


Рис. 48

Таблица 28

Адрес команды	Команда	Код команды
09	П→x e	6E
10	К x<0 1	[1
11	П→x d	6Г
12	К x<0 2	[2
13	3	03
14	С/П	50
15	2	02
16	С/П	50
17	П→x d	6Г
18	К x⩾0 3	93
19	1	01
20	С/П	50
21	4	04
22	С/П	50

Таблица 29

Адрес команды	Команда	Код команды
00	1	01
01	8	08
02	x→П 1	41
03	1	01
04	6	06
05	x→П 2	42
06	2	02
07	2	02
08	x→П 3	43

После этого пуск программы производят нажатием клавиши В/О С/П.

Задача 3. Задан массив Y чисел, среди которых имеется число 0. Просуммировать элементы массива, стоящие после нулевого элемента. $Y=\{2{,}5;\ 3{,}4;\ 7{,}2;\ 0;\ 5{,}3;\ 11{,}7;\ 2{,}8;\ 8{,}1\}$.

Решение. Распределим регистры памяти: 2,5→РП5; 3,4→РП6; 7,2→РП7; 0→РП8; 5,3→РП9; 11,7→РПa; 2,8→РПb. Для составления программы понадобится новая команда — команда косвенной индикации. С помощью этой команды производится модификация содержимого адресуемого регистра и вызов в регистр X содержимого того регистра, который соответствует модифицированному коду. Для команды косвенной индикации возьмем РП4, для команды косвенного условного перехода — РПd, для команды безусловного косвенного перехода — РПe, значение суммы в РП0 (табл. 30). Схема алгоритма решения задачи дана на рисунке 49.

Таблица 30.

Адрес команды	Команда	Код команды
00	К П→x 4	Г4
01	F x=0	5E
02	0 0	00
03	П→x 4	64
04	1	01
05	2	02
06	—	11
07	К x<0 d	[Г
08	П→x 0	60
09	К П→x 4	Г4
10	+	10
11	x→П 0	40
12	К БП e	8E
13	П→x 0	60
14	С/П	50

Разместим элементы массива в регистры 5—c, 0→РП0, 12→РПd, 3→РПe, 4→РП4. Так как элементы массива располагаются, начиная с 5-го РП, то начальное значение $i=4$. Это значение посылаем в РП4. Всего в массиве 8 элементов, поэтому конечное

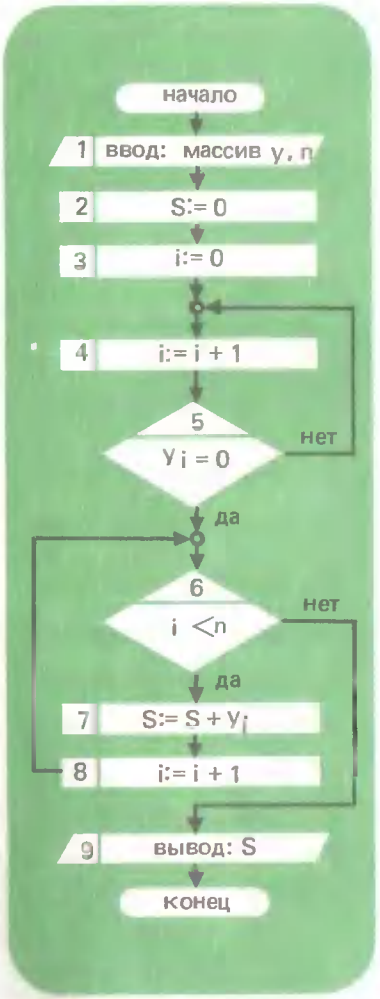


Рис. 49

значение $i=4+8=12$, это значение засылаем в РПd. В программе вместо команды [К][БП][е] можно было использовать команду [БП][0][3], но это увеличило бы объем программы.

Задача 4. Даны два массива $A=\{3{,}2;\ 5{,}1;\ -2{,}7;\ 4{,}9;\ 8{,}1\}$ и $B=\{-0{,}5;\ -13{,}1;\ 19{,}7;\ 48{,}6;\ -27{,}4\}$. Написать программу обмена местами массивов A и B.

Решение. Схема алгоритма изображена на рисунке 50. Распределим регистры памяти следующим образом: для чисел массива А отведем РП1—5; для чисел массива В—РП6—а, РП0—для хранения числа с, РПb и РПc — для косвенной индикации чисел массивов А и В соответственно, РПd — для косвенного условного перехода. Программа приведена в таблице 31.

В команде косвенного условного перехода, если $x\neq 0$, то должен осуществляться переход на команду 02. Поэтому перед пуском программы засылаем 02 в РПd.

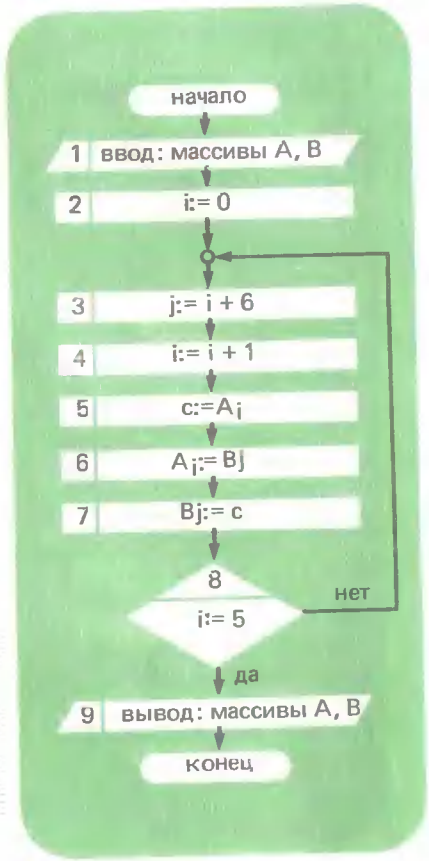


Рис. 50

Таблица 31

Адрес команды	Команда	Код команды
00	[0]	00
01	[x→П][b]	4L
02	[П→x][b]	6L
03	[6]	06
04	[+]	10
05	[x→П][c]	4[
06	[П→x][b]	6L
07	[1]	01
08	[+]	10
09	[x→П][b]	4L
10	[К][П→x][b]	ГL
11	[x→П][0]	40
12	[К][П→x][c]	Г[
13	[К][x→П][b]	LL
14	[П→x][0]	60
15	[К][x→П][c]	L[
16	[П→x][b]	6L
17	[5]	05
18	[—]	11
19	[К][x=0][d]	ЕГ
20	[С/П]	50

Вывод полученных массивов A и B производится вручную.

Задача 5. Написать программу вычисления значения многочлена $y=28{,}3x^7+32{,}8x^6-17{,}5x^5-47{,}2x^4+21{,}6x^3-36{,}2x^2+50{,}3x-45{,}8$ при любом x.

Решение. Запишем многочлен n-й степени в общем виде: $y=a_nx^n+a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\ldots+a_2x^2+a_1x+a_0$. Обычно значения многочленов такого вида вычисляются по схеме Горнера: $y=((\ldots((a_nx+a_{n-1})x+a_{n-2})x+\ldots+a_2)x+a_1)x+a_0$.

Предположим, что коэффициенты многочлена образуют массив А, состоящий из $n+1$ элемента.

В нашем примере степень многочлена $n=7$, поэтому в массиве A будет 8 чисел. Отведем для чисел массива с 1 по 8 регистры памяти, т. е. 28,3→РП8, 32,8→РП7; —17,5→РП6; —47,2→РП5; 21,6→РП4; —36,2→РП3; 50,3→РП2; —45,8→РП1. Для косвенной индикации элементов массива А отведем РП0. При косвенной индикации содержимое регистра 0 каждый раз уменьшается на 1. В массиве 8 чисел, поэтому в РП0 занесем число $8+1=9$. Значение x→РП9, y→РПа, РПb — для косвенного условного перехода. Программа приведена в таблице 32, схема алгоритма — на рисунке 51.

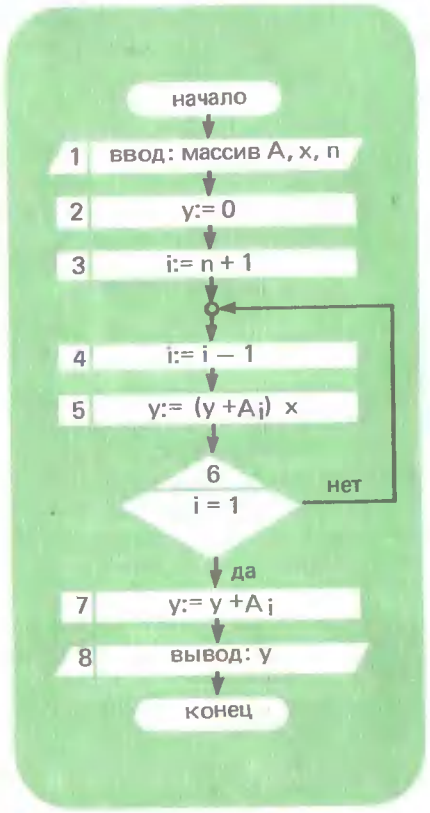


Рис. 51

Таблица 32

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
00	[К][П→x][0]	Г0	07	[1]	01
01	[П→x][a]	6—	08	[—]	11
02	[+]	10	09	[К][x=0][b]	EL
03	[П→x][9]	69	10	[К][П→x][0]	Г0
04	[×]	12	11	[П→x][a]	6—
05	[x→П][a]	4—	12	[+]	10
06	[П→x][0]	60	13	[С/П]	50

Перед пуском программы необходимо $y=0 \to$ РПа, $0 \to$ РПb, $9 \to$ РП0, $x \to$ РП9. Ответ высвечивается на индикаторе. Если $x=1$, то $y=-13{,}7$.

Упражнения

Написать программы решения следующих задач:

1. Найти наименьший элемент массива A.
2. Расположить элементы массива A по убыванию.
3. Найти среднее арифметическое элементов массива A.

17. Задачи на моделирование

Рассмотрим задачи, решение которых требует большого объема вычислений. Для решения таких задач разрабатывается сначала алгоритмическая модель исследуемого процесса. Поэтому назовем такие задачи «задачами на моделирование».

Задача 1. При $t_0=0$ материальная точка находилась в точке $x=4{,}2$. Скорость движения материальной точки есть функция от времени t, т. е. $v=3{,}7t^2-1{,}6$. Найти положение точки через $T=10$ с.

Решение. Пусть материальная точка движется по прямой с постоянной скоростью v. Тогда путь s, пройденный точкой за время t, равен $s=v \cdot t$. Если x_0 — координата материальной точки в начальный момент времени t_0, то в момент времени $T=t_0+t$ координата $x=x_0+vT$.

В нашем примере скорость есть функция $v=f(t)$. Поступим следующим образом. Разобьем отрезок времени t точками t_0, t_1, ..., t_n на n отрезков, причем $t_0=0$, $t_n=t_0+t=T$. Величину i-го отрезка обозначим $\Delta t_i=t_i-t_{i-1}$. На каждом из отрезков Δt_i скорость тоже является функцией. Однако, учитывая, что величины отрезков Δt_i достаточно малы, сделаем допущение, что скорость v на каждом отрезке Δt_i постоянна и равна одному какому-то значению функции $v=f_i$ на этом отрезке. Тогда координата x материальной точки в момент времени $T=t_0+t$ будет $x \approx x_0+f_0\Delta t_0+f_1\Delta t_1+\ldots+f_n\Delta t_n$.

Естественно, что таким образом x будет найдено приближенно. Приближение, вообще говоря, будет тем точнее, чем меньше величина отрезка Δt_i.

Для облегчения вычислений все Δt_i возьмем одинаковыми. За постоянную скорость $v=f_i$ на i-м отрезке возьмем значение функции f в одном из концов i-го отрезка. Тогда $x \approx x_0+\sum_{i=0}^{n-1} f_i\Delta t$, или $x \approx x_0+\sum_{i=1}^{n} f_i\Delta t$.

Общая схема алгоритма вычисления приведена на рисунке 52.

Обычно вычисление $v=f$ осуществляется в подпрограмме, что позволяет при переходе к решению аналогичных задач не менять основную программу, а изменять подпрограмму в зависимости от заданной функции f.

Подпрограмма для нашего примера приведена в таблице 33. Распределим регистры памяти: $x_0 \to$ РП1, $t_0 \to$ РП2, $T \to$ РП3, $n \to$ РП4 (n выбирается произвольно), $t \to$ РП5, $x \to$ РП6, $\Delta t \to$ РП7, $i \to$ РП8, $v \to$ РП9.

По адресу 14 записана команда косвенного перехода на подпрограмму. Она реализуется клавишами [К] [ПП] [N], где N — номер регистра памяти. В программной памяти она занимает одну ячейку. При выполнении этой команды производится модификация числа, находящегося в РПN, и переход к исполнению команды по адресу, равному модифицированному числу.

В нашем примере подпрограмма начинается с адреса 32. Поэтому 32 → РПе, так как e входит в команду косвенного перехода на подпрограмму.

Перед выполнением программы необходимо ввести данные согласно распределенным регистрам памяти: $x_0=4{,}2 \to$ РП1, $t_0=0 \to$ РП2, $T=10 \to$ РП3, пусть $n=20 \to$ РП4, 32 → РПе. Пуск программы на выполнение производится клавишами [В/О] [С/П]. Ответ высвечивается на индикаторе: 1315,575.

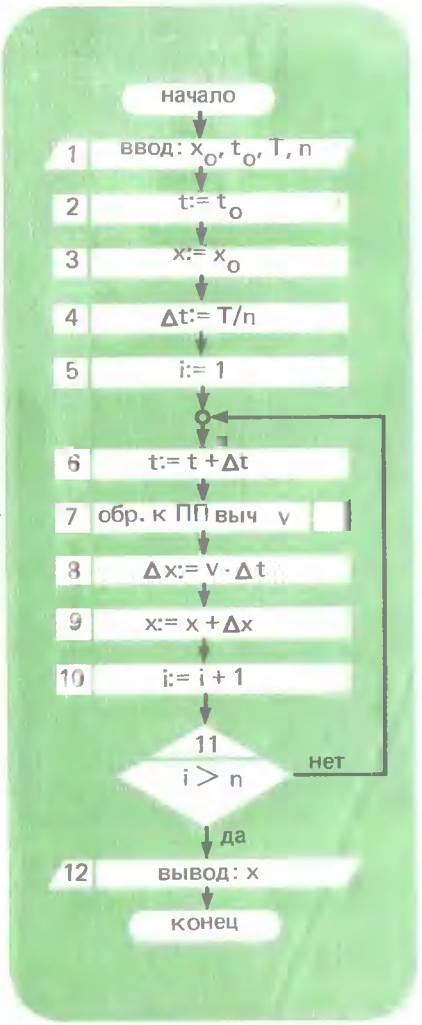


Рис. 52

Таблица 33

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
00	[П→x] [2]	62	05	[П→x] [4]	64
01	[x→П] [5]	45	06	[÷]	13
02	[П→x] [1]	61	07	[x→П] [7]	47
03	[x→П] [6]	46	08	[1]	01
04	[П→x] [3]	63	09	[x→П] [8]	48

Продолжение

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
10	П→x 5	65	27	/—/	0L
11	П→x 7	67	28	F x<0	5[
12	+	10	29	1 0	10
13	x→П 5	45	30	П→x 6	66
14	К ПП e	—E	31	С/П	50
15	П→x 9	69	32	П→x 5	65
16	П→x 7	67	33	F x^2	22
17	×	12	34	3	03
18	П→x 6	66	35	,	0—
19	+	10	36	7	07
20	x→П 6	46	37	×	12
21	П→x 8	68	38	1	01
22	1	01	39	,	0—
23	+	10	40	6	06
24	x→П 8	48	41	—	11
25	П→x 4	64	42	x→П 9	49
26	—	11	43	В/О	52

Задача 2. В 1975 г. в городе было 2 000 000 жителей. Прирост населения постоянный и составляет 1,2 % в год. Определить, в каком году население города достигнет 3 000 000.

Таблица 34

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
00	П→x b	6L	13	П→x c	6[
01	x→П e	4E	14	×	12
02	П→x a	6—	15	1	01
03	x→П 0	40	16	0	00
04	П→x e	6E	17	0	00
05	П→x d	6Г	18	÷	13
06	—	11	19	П→x e	6E
07	К x<0 9	[9	20	+	10
08	П→x 0	60	21	x→П e	4E
09	1	01	22	К БП 8	88
10	+	10	23	П→x 0	60
11	x→П 0	40	24	С/П	50
12	П→x b	6L			

Решение. Обозначим $a=1975$, $b=2\,000\,000$, $c=1{,}2$, $d=3\,000\,000$. Схема алгоритма представлена на рисунке 53. Распределим регистры памяти: $d\to$РПd, $c\to$РПc, $e\to$РП1, $a\to$РПa, $b\to$РПb, $i\to$РП0. Программа приведена в таблице 34. Для команды косвенного условного перехода используем РП9, для косвенного безусловного перехода — РП8.

Перед пуском программы заносим исходные данные в регистры памяти: 1975→РПa, 2000000→РПb, 1,2→РПc, 3000000→РПd, 23→РП9, 4→РП8. Ответ высветится на индикаторе: 2016.

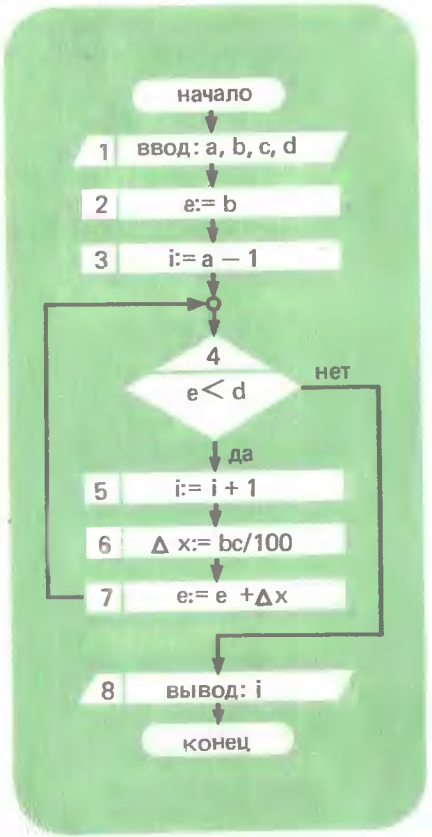


Рис. 53

Упражнения

1. Скорость движущегося тела возрастает обратно пропорционально пройденному пути. В начальный момент движения тело находилось на расстоянии 5 м от начала отсчета пути и имело скорость $v=20$ м/с. Написать программу определения положения и скорости тела через 8 с после начала движения.

2. Скорость движущегося тела $v=\sqrt{\frac{1}{3}x^2+\frac{1}{5}t^2}$ м/с, где t — время от начала движения, отсчитываемое в секундах; x — абсцисса тела в момент времени t, отсчитываемая в метрах. В начальный момент $x_0=5$ м. Написать программу вычисления времени, через которое тело достигнет точки $x=10$ м.

3. Сумма 10 000 р. положена в сберегательную кассу, при этом прирост составляет 2 % в год и считается непрерывным. Написать программу определения промежутка времени, через который сумма достигнет 15 000 р.

4. Через сосуд емкостью 100 л, наполненный 50 %-ным раствором соли, непрерывно протекает вода, причем в 1 мин вливается 5 л воды и такое же количество раствора выливается. Написать программу, определяющую, через какой промежуток времени раствор соли в сосуде станет 25 %-ным.

18. Вычисление определенных интегралов по формулам прямоугольников, трапеций

Пусть f — непрерывная на отрезке [a, b] функция. Если F — первообразная для f, то

$$\int_a^b f(x)dx = F(b) - F(a).$$

Для многих функций первообразную нельзя выразить через элементарные функции. По существу, это означает, что первообразная неизвестна. Это приводит к необходимости приближенного вычисления определенного интеграла.

Пусть задана на отрезке $[a, b]$ непрерывная функция $f(x)$. Разобьем отрезок $[a, b]$ на n отрезков точками $x_0 = a$, x_1, ..., $x_n = b$. Обозначим через Δx_i длину i-го отрезка. Составим сумму $\sum_{i=0}^{n-1} f(\delta_i)\Delta x_i$, где δ_i взята на i-м отрезке. Такую сумму называют интегральной суммой функции $f(x)$ на отрезке $[a, b]$.

Предел интегральных сумм при стремлении к нулю наибольшей длины отрезков Δx_i есть, согласно определению, интеграл $\int_a^b f(x)dx$. Следовательно, каждая интегральная сумма $\sum_{i=0}^{n-1} f(\delta_i)\Delta x_i$ приближенно равна интегралу $\int_a^b f(x)dx$ тем точнее, чем меньше длины отрезков Δx_i. Для облегчения вычисления отрезки Δx_i берут равными, т. е. $\Delta x = \frac{b-a}{n}$, а значение функции $f(\delta_i)$ берут в одном из концов отрезка $f(x_i)$ или $f(x_{i+1})$. Итак,

$$\int_a^b f(x)dx \approx \frac{b-a}{n} \sum_{i=1}^{n-1} f(x_i) (*)$$ или

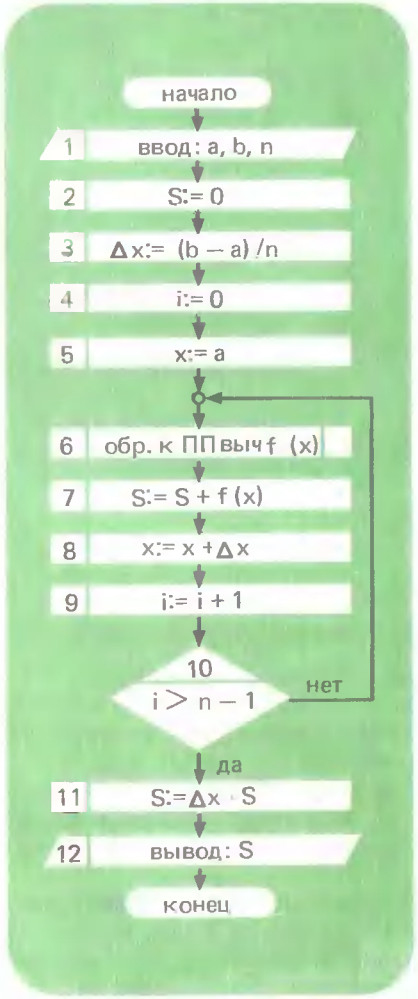


Рис. 54

$$\int_a^b f(x)dx \approx \frac{b-a}{n} \sum_{i=1}^{n} f(x_i) (**).$$ Эти формулы называют формулами прямоугольников. Схема алгоритма вычисления по формуле (*) приведена на рисунке 54. Для формулы (**) в данной схеме необходимо изменить значение i от 1 до n и блок 8 поставить после блока 5.

Лучшим приближением будет полусумма указанных выражений, т. е.

$$\int_a^b f(x)dx \approx \frac{1}{2}\left(\frac{b-a}{n}\sum_{i=0}^{n-1} f(x_i) + \frac{b-a}{n}\sum_{i=1}^{n} f(x_i)\right).$$

Раскроем скобки, получим:

$$\int_a^b f(x)dx \approx \frac{b-a}{n}\left(\frac{f(x_0)+f(x_n)}{2} + f(x_1) + \ldots + f(x_{n-1})\right)$$

Эту формулу называют формулой трапеций, схема алгоритма приведена на рисунке 55.

Задача 1. Вычислить приближенное значение интеграла $\int_1^2 \sqrt{x}\ln x\,dx$, используя формулу прямоугольников.

Решение: Разобьем отрезок [1; 2] на 20 частей. Для составления программы (табл. 35) воспользуемся схемой алгоритма, изображенной на рисунке 54. Распределим регистры памяти: a→РП1, b→→РП2, n→РП3, Δx→РП4, S→РП5, x→РП6, i→РП7, $f(x)$→РП8.

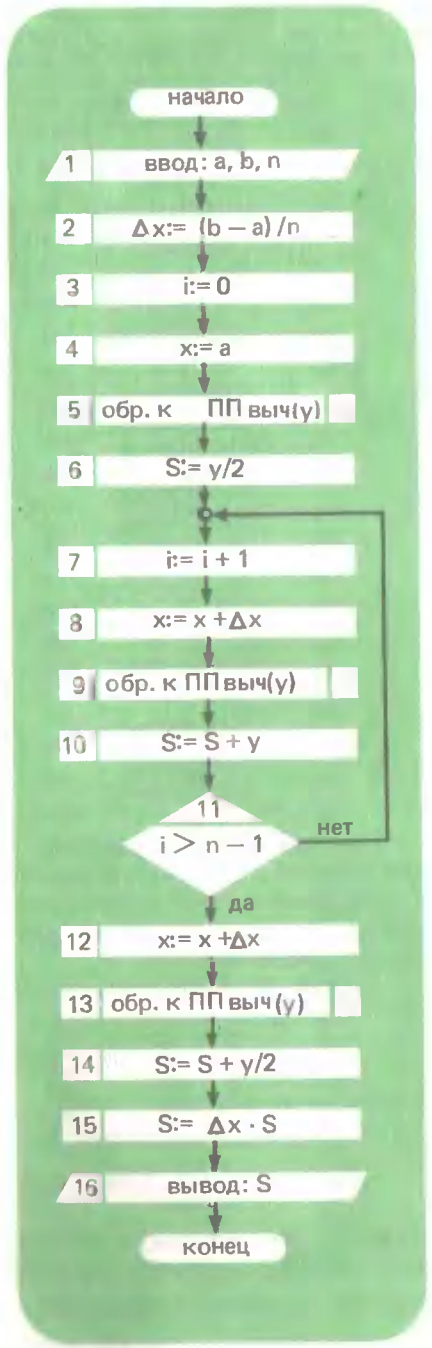


Рис. 55

Таблица 35

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
00	0	00	21	П→x 7	67
01	x→П 5	45	22	1	01
02	П→x 2	62	23	+	10
03	П→x 1	61	24	x→П 7	47
04	—	11	25	П→x 3	63
05	П→x 3	63	26	1	01
06	÷	13	27	—	11
07	x→П 4	44	28	П→x 7	67
08	0	00	29	—	11
09	x→П 7	47	30	К x<0 d	[Г
10	П→x 1	61	31	П→x 4	64
11	x→П 6	46	32	П→x 5	65
12	К ПП e	—E	33	×	12
13	П→x 8	68	34	С/П	50
14	П→x 5	65	35	П→x 6	66
15	+	10	36	F √	21
16	x→П 5	45	37	П→x 6	66
17	П→x 6	66	38	F ln	18
18	П→x 4	64	39	×	12
19	+	10	40	x→П 8	48
20	x→П 6	46	41	В/О	52

Подпрограмма начинается с команды 35, значит, 35→РПе, для команды косвенного условного перехода использован РПd, в него необходимо занести число 12. Перед пуском программы нужно разнести исходные данные: 1→РП1, 2→РП2, 20→РП3. Ответ. 0,46986014.

Задача 2. Вычислить по формуле трапеций определенный интеграл

$$\int_{1,2}^{2,2} x^2 e^{0,5x} dx$$

с точностью $\alpha = 0,01$.

Решение. Определим n, используя формулу погрешности

$$\alpha = \frac{(b-a)^3}{12n^2} M_2, \quad \text{где } M_2 = \max_{a \leqslant x \leqslant b} |f''(x)|.$$

Находим, что максимум второй производной на [1,2; 2,2] равен $M_2 \approx 22,862$. Подставим это значение в формулу для α:

$$0,01 = \frac{(2,2-1,2)^3}{12n^2} \cdot 22,862.$$

Отсюда находим, что $n \approx \sqrt{\frac{22,862}{0,12}} \approx 14$. Итак, все исходные данные известны, распределим регистры памяти: a→РП1, b→РП2, n→РП3, Δx→РП4, S→РП5, x→РП6, i→РП7, $f(x)$→РП8 и составим программу (табл. 36).

Таблица 36

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
00	П→x 2	62	28	П→x 3	63
01	П→x 1	61	29	1	01
02	—	11	30	—	11
03	П→x 3	63	31	П→x 7	67
04	÷	13	32	—	11
05	x→П 4	44	33	К x<0 d	[Г
06	0	00	34	П→x 6	66
07	x→П 7	47	35	П→x 4	64
08	П→x 1	61	36	+	10
09	x→П 6	46	37	x→П 6	46
10	К ПП e	—E	38	К ПП e	—E
11	П→x 8	68	-39	П→x 8	68
12	2	02	40	2	02
13	÷	13	41	÷	13
14	x→П 5	45	42	П→x 5	65
15	П→x 7	67	43	+	10
16	1	01	44	П→x 4	64
17	+	10	45	×	12
18	x→П 7	47	46	С/П	50
19	П→x 6	66	47	П→x 6	66
20	П→x 4	64	48	2	02
21	+	10	49	÷	13
22	x→П 6	46	50	F e^x	16
23	К ПП e	—E	51	П→x 6	66
24	П→x 5	65	52	F x^2	22
25	П→x 8	68	53	×	12
26	+	10	54	x→П 8	48
27	x→П 5	45	55	В/О	52

Разнесем исходные данные по регистрам памяти: 1,2→РП1, 2,2→РП2, 14→РП3, 15→РПd, 47→РПе. Результат следует округлить до 0,01, получим $\int_{1,2}^{2,2} x^2 e^{0,5x} dx \approx 8,46$.

При вычислении других интегралов в таблицах 35 и 36 следует изменить только подпрограммы.

Упражнения

Вычислить определенные интегралы приближенно, используя формулы прямоугольников и трапеций:

1. $\int\limits_3^4 \frac{e^x}{\lg x+1}dx.$
2. $\int\limits_1^{1,5} \frac{x^2-2}{10^x}dx.$
3. $\int\limits_{0,5}^1 \frac{\sin x\cos x}{x}dx.$
4. $\int\limits_1^2 \left(\frac{\sqrt{x}}{\lg x}+\ln x\right)dx.$

19. Решение трансцендентных уравнений методом деления отрезка пополам

Пусть задано уравнение $f(x)=0$. Решение уравнения методом деления отрезка пополам состоит из двух этапов: отделения корней и уточнения корней.

Говорят, что корни уравнения $f(x)=0$ отделены, если на числовой оси указаны отрезки, в каждом из которых содержится по одному корню.

Рассмотрим графический метод отделения корней. Строят график функции $y=f(x)$. Абсциссы точек пересечения графика функции $y=f(x)$ с осью абсцисс являются корнями уравнения $f(x)=0$. Так как графики строятся по нескольким точкам, то абсциссы точек пересечения графика с осью абсцисс являются грубым приближением. Можно сказать, что корень находится возле указанной точки, т. е. в некотором интервале.

Если построение графика $y=f(x)$ вызывает затруднения, то уравнение $f(x)=0$ преобразовывают к виду $f_1(x)=f_2(x)$. Строят графики функций $y_1=f_1(x)$ и $y_2=f_2(x)$. Абсциссы точек пересечения этих графиков являются корнями уравнения $f(x)=0$.

При отделении корней следует использовать теорему: если функция $y=f(x)$ определена и непрерывна на отрезке $[a, b]$ и принимает на концах отрезка значения противоположных знаков, т. е. $f(a)f(b)<0$, то внутри отрезка $[a, b]$ существует по крайней мере один корень уравнения $f(x)=0$.

Пусть корень c уравнения $f(x)=0$ уже отделен и принадлежит отрезку $[a, b]$ длиной $\delta=|b-a|$. Найдем его с заданной точностью h. Пусть x_0 произвольная точка отрезка $[a, b]$. Так как x_0 и корень c принадлежат одному отрезку $[a, b]$, то

$$|x-c|<|b-a|.$$

Таким образом, можно любую точку x_0 из отрезка $[a, b]$ взять за приближение корня c с точностью $\delta=|b-a|$.

В качестве нулевого приближения выберем точку $x_0=\frac{a+b}{2}$. Эта точка разделит отрезок $[a, b]$ пополам: $[a, x_0]$ и $[x_0, b]$. Если $f(x_0)\neq 0$, то корень уравнения $f(x)=0$ будет принадлежать тому из отрезков, на концах которого функция $y=f(x)$ принимает значения разных знаков.

Пусть корень уравнения принадлежит отрезку $[a, x_0]$, длина которого $\frac{\delta}{2}$. Тогда за следующее приближение выберем точку $x_1=\frac{a+x_0}{2}$. Ясно, что $|c-x_1|<\frac{\delta}{2}$. Тогда x_1 тоже разделит отрезок $[a, x_0]$ пополам: $[a, x_1]$ и $[x_1, x_0]$. Пусть $f(x_1)f(x_0)<0$. Тогда выбираем следующее приближение $x_2=\frac{x_1+x_0}{2}$ и так далее.

Таким образом, с каждым следующим приближением сужаем в два раза отрезок, которому принадлежит корень уравнения. Поэтому следующее приближение будет точнее. Сужение проводим до тех пор, пока длина отрезка не станет меньше заданной точности.

Схема алгоритма решения уравнения методом деления отрезка пополам приведена на рисунке 56. Вычисление значений $f(x_i)$ производится в подпрограмме.

Задача 1. Найти корни уравнения $2\lg x-\frac{x}{2}+1=0$ с точностью $h=0{,}01$.

Решение. Графически отделим корень уравнения. Для этого преобразуем уравнение к виду

$$\lg x-\frac{x}{4}+\frac{1}{2}=0;\quad \lg x=\frac{x-2}{4}.$$

Построим графики $y_1=\lg x$ и $y_2=\frac{x-2}{4}$ (рис. 57).

Из графика определяем, что уравнение имеет два действительных корня $c_1\in[0{,}1;\ 1]$, $c_2\in[4;\ 5]$. Для уточнения корней по схеме алгоритма составим программу (табл. 37). Распределим регистры памяти: $a\to$РПа, $b\to$РПb, $x\to$РП0, $\delta\to$РП1, $h\to$РПc, $k\to$РП2, $f(x)\to$РП3, $f(a)\to$РП4 — для подпрограммы $x\to$РП5, $f(x)\to$РП6, обращение к подпрограмме РПd.

Перед пуском программы необходимо разнести исходные данные сначала для первого корня: $a=0{,}1\to$РПа, $b=1\to$РПb, $h=0{,}01\to$РПc, $48\to$РПd. Пуск программы на выполнение производится клавишами [В/О] [С/П]. После выполнения программы на индикаторе высвечивается значение a, после нажатия клавиши [x↔y] на индикаторе высветится значение b. Ответ: $0{,}39\leqslant c_1\leqslant 0{,}40$. Аналогично поступаем для второго корня. Получаем ответ: $4{,}68\leqslant c_2\leqslant 4{,}69$.

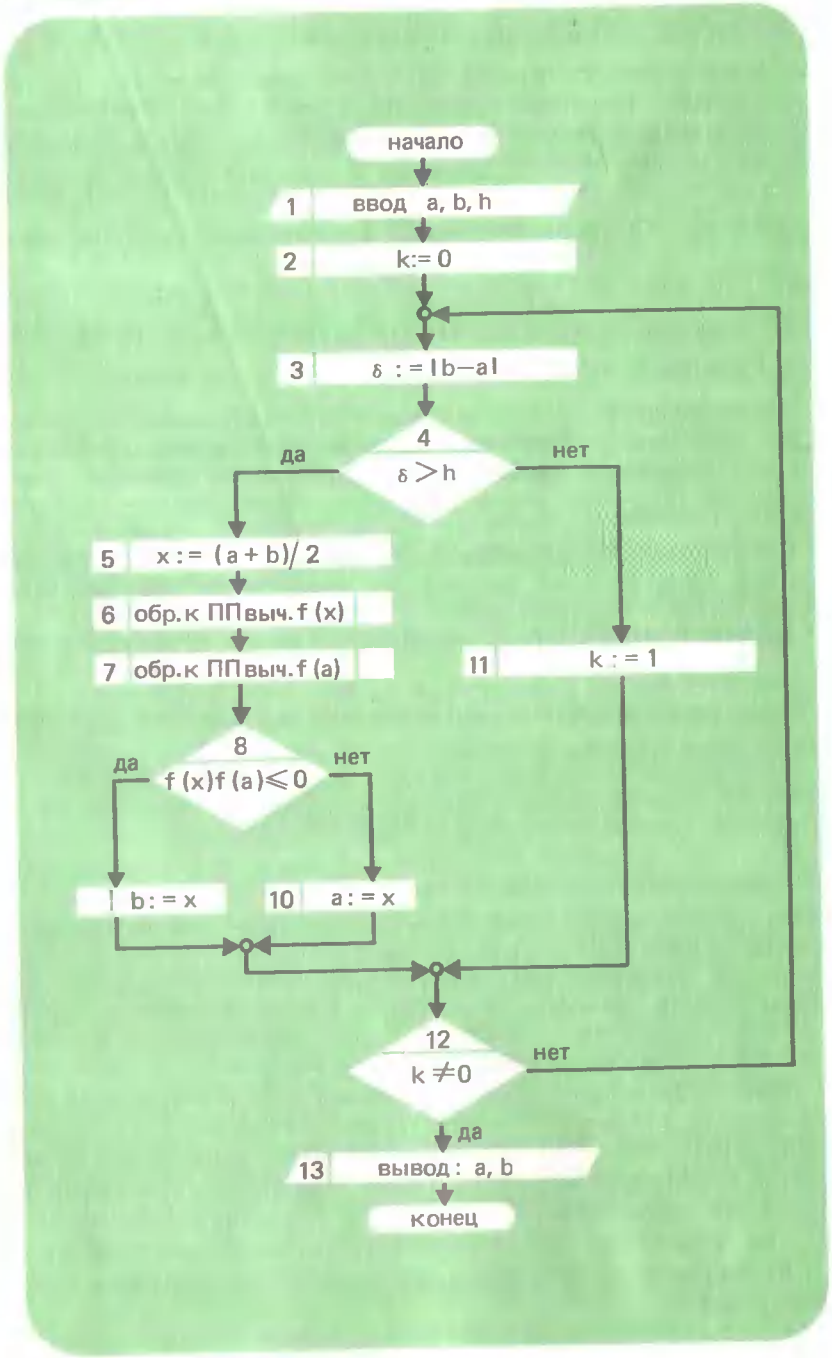


Рис. 56

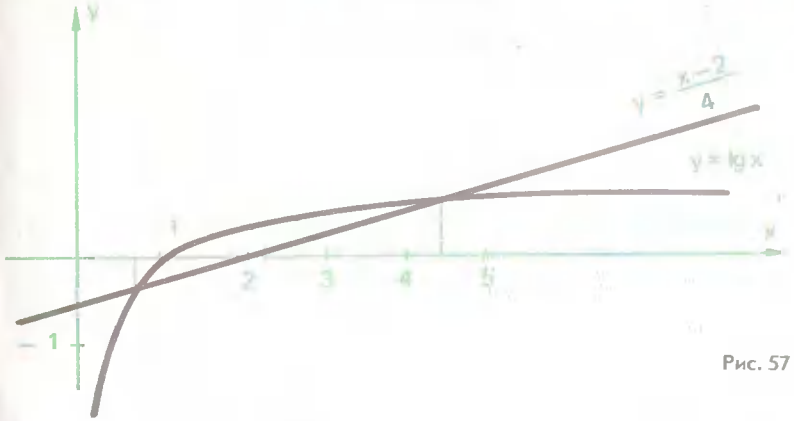


Рис. 57

Таблица 37

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
00	0	00	24	x→П 5	45
01	x→П 2	42	25	К ПП d	—Г
02	П→x b	6L	26	П→x 6	66
03	П→x a	6—	27	П→x 3	63
04	—	11	28	×	12
05	К |x|	31	29	/—/	0L
06	x→П 1	41	30	F x≥0	59
07	П→x c	6[	31	3 6	36
08	П→x 1	61	32	П→x 0	60
09	—	11	33	x→П b	4L
10	F x<0	5[	34	БП	51
11	4 0	40	35	4 2	42
12	П→x a	6—	36	П→x 0	60
13	П→x b	6L	37	x→П a	4—
14	+	10	38	БП	51
15	2	02	39	4 2	42
16	÷	13	40	1	01
17	x→П 0	40	41	x→П 2	42
18	П→x 0	60	42	П→x 2	62
19	x→П 5	45	43	F x≠0	57
20	К ПП d	—Г	44	0 2	02
21	П→x 6	66	45	П→x b	6L
22	x→П 3	43	46	П→x a	6—
23	П→x a	6—	47	С/П	50

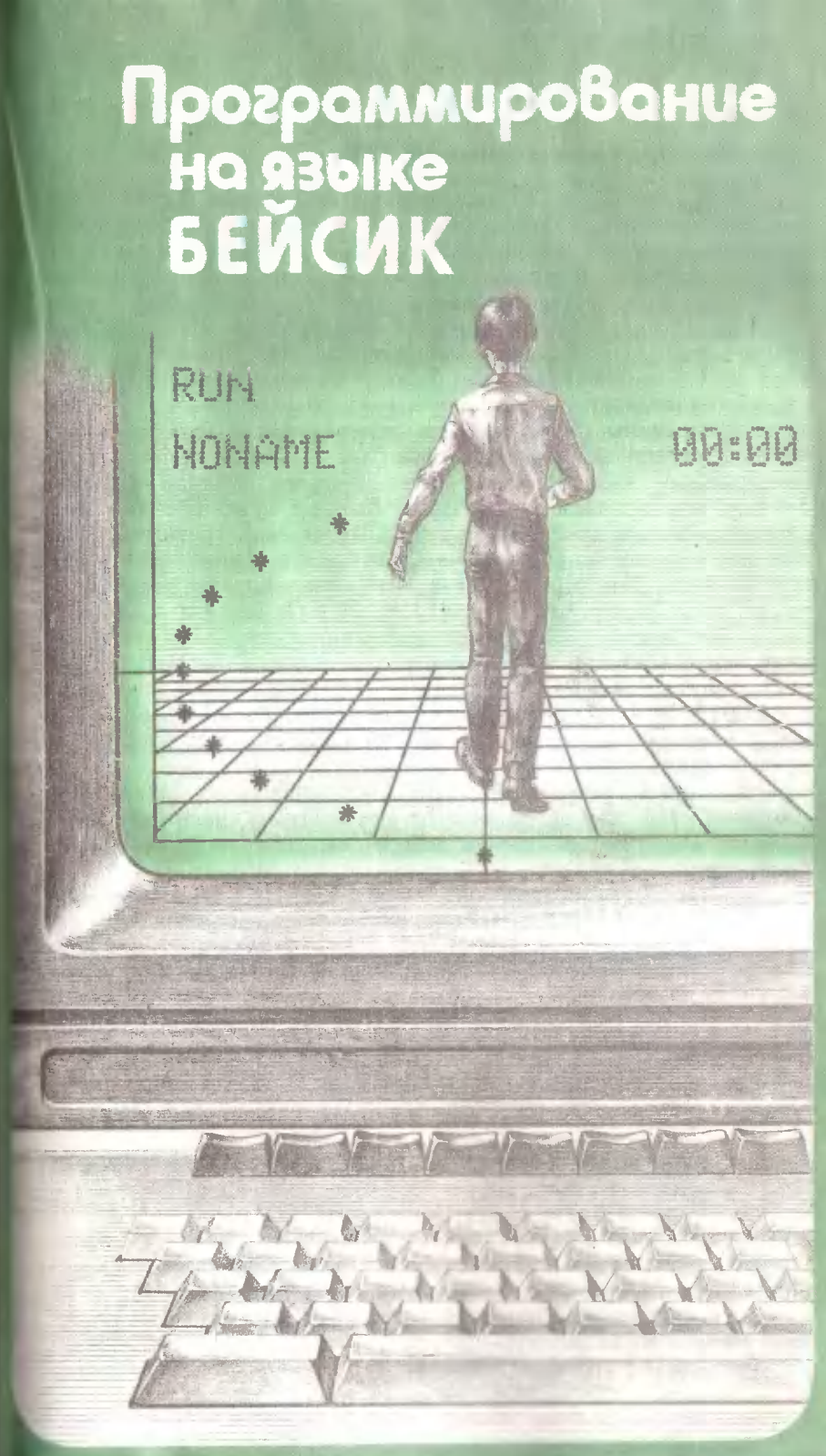
Продолжение

Адрес команды	Команда	Код команды	Адрес команды	Команда	Код команды
48	П→x 5	65	56	/—/	0L
49	F lg	17	57	П→x 7	67
50	2	02	58	+	10
51	×	12	59	1	01
52	x→П 7	47	60	+	10
53	П→x 5	65	61	x→П 6	46
54	2	02	62	В/О	52
55	÷	13			

Упражнения

Найти корни уравнений с точностью 0,01:

1. $\sin x - 0{,}2x = 0$.
2. $\cos x = x^2$.
3. $x - 10 \sin x = 0$.
4. $\sqrt{x+1} = \frac{1}{x}$.
5. $x^2 - 10 \lg x - 3 = 0$.



1. Язык программирования БЕЙСИК

Человек и ЭВМ — партнеры в процессе решения задачи. А как известно, партнеры должны «говорить» на языке, понятном обоим сторонам. Такими языками являются алгоритмические языки программирования — искусственные языки, специально созданные человеком. Естественные языки — русский, белорусский или, к примеру, английский, для непосредственного диалога с машиной непригодны, так как имеют слишком большой словарный запас и обладают неоднозначностью фраз, а машина может исполнять только четкие, однозначные инструкции. Именно поэтому ученые были вынуждены разрабатывать специальные языки, понятные ЭВМ. Эти языки содержат несколько десятков слов какого-нибудь естественного языка, цифры, математические символы и четкие правила образования новых фраз. Каждое предложение, написанное на языке программирования, имеет единственный смысл, поэтому машину можно «научить» понимать и исполнять такую инструкцию. Для персональных ЭВМ разработаны диалоговые языки программирования БЕЙСИК, ЛОГО, Рапира и др. Они позволяют вести непосредственный диалог с ЭВМ, используя клавиатуру машины.

Название языка БЕЙСИК возникло от сокращения английских слов Beginner's All- purpose Symbolic Instruction Code (BASIC), что означает «многоцелевой язык символических инструкций для начинающих».

Этот язык используется преимущественно в диалоговых системах. На нем составляются программы решения задач вычислительного характера с небольшим объемом исходной информации.

Программа, записанная на языке БЕЙСИК, состоит из операторов, комментариев и текстов. Операторы предназначены для описания данных и операций над ними и указывают на однозначно определенные действия для машины. В одной строке можно писать несколько операторов, разделяя их двоеточием. Комментарии поясняют программу и облегчают ее понимание. Они не влияют на исполнение программы.

2. Алфавит

Для написания программ используют следующие символы:

1. Двадцать шесть прописных букв латинского алфавита:

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z.

2. Десять цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

3. Специальные символы: + (плюс), — (минус), * (звездочка), / (наклонная черта), ¬ (возведение в степень), ‘ (апостроф), : (двоеточие), ; (точка с запятой).

4. Знаки операций отношения: = (равно), > (больше), < (меньше). Для написания операций: «не равно» используются два символа <> или ><, «больше или равно»—>=, «меньше или равно» — <=.

5. Круглые скобки: ((левая скобка),) (правая скобка).

В языке БЕЙСИК используются английские слова, каждое из которых рассматривается как отдельный элемент (ключевое или служебное слово) языка. Они имеют единственное начертание и значение. Перечень служебных слов БЕЙСИКа приведен на втором форзаце.

Для записи комментариев и текстов можно использовать также прописные буквы русского алфавита.

3. Величины БЕЙСИКа

При составлении программ используются величины трех видов: константы, переменные и номера строк.

Константой называется величина, которая определена при написании программы и не изменяется в процессе ее исполнения. В БЕЙСИКе используются две формы записи чисел. Одна из них близка к естественной форме. Для разделения целой и дробной частей используется точка. Например: .5; 100.; —5.7. Такая форма записи называется формой с фиксированной точкой.

Вторая форма близка к стандартной форме записи числа. Например, если задано число $1{,}2 \cdot 10^4$, то на БЕЙСИКе используется форма с десятичным множителем 1.2E4 или 1.2E04.

Порядок числа может быть записан только целым числом. Такая форма называется формой с плавающей точкой. При записи программ необходимо учитывать диапазон допустимых чисел для ЭВМ, на которой исполняется программа. Для «Электроники ДЗ-28» числа должны удовлетворять ограничениям: модуль числа не более $(1—10^{-12}) \cdot 10^{99}$, количество цифр в числе не превышает 12.

Переменной называется величина, которой в процессе исполнения программы присваиваются некоторые значения.

Переменная обозначается идентификатором (именем). Идентификаторы переменных состоят из одной буквы или из буквы и следующей за ней цифры. Например, A, B4, W7. В пределах одной программы можно использовать $26+26 \cdot 10=286$ различных идентификаторов переменных.

Номера строк представляют собой натуральные числа от 1 до 7999 и пишутся в начале каждой строки программы. Нумерация строк идет в порядке возрастания чисел. Никакие две разные строки не могут быть занумерованы одним и тем же числом. Номера располагаются в начале каждой строки. При записи программ на языке БЕЙСИК каждой строке рекомендуем ставить в соответствие номер реализуемого блока схемы, приписывая справа нуль.

4. Встроенные функции

Для вычисления значений многих математических функций используются так называемые встроенные функции. При программировании достаточно написать наименование соответствующей функции и аргумент, заключенный в скобки. Для тригонометрических функций аргумент задается только в радианах.

Список встроенных функций приведен на втором форзаце.

5. Выражения

Простым выражением называется константа, переменная, встроенная функция или композиция функций. Арифметические выражения образуются из простых выражений с помощью операций сложения, вычитания, умножения, деления, возведения в степень, изображаемых соответственно следующими символами: +, —, *, /, ¬.

Символы операций писать обязательно. В БЕЙСИКе запись производится в строку без подстрочных и надстрочных знаков.

Например, выражения $\frac{2\sin\frac{x}{2}}{\sqrt{b}}$, $\frac{-1:2x+a^2}{4x^2}$, $(a+x^3)^{\frac{2}{3}}$ на языке БЕЙСИК запишется следующим образом:

```
2*SIN (X/2)/SQR (B)
-1/2(2*X)+A¬2)/(4*X*X)
(A+X¬3)¬(2/3)
```

При вычислении выражений операции исполняются в такой последовательности: 1) вычисление значений функций; 2) возведение в степень; 3) умножение и деление; 4) сложение и вычитание. Порядок можно изменить с помощью скобок.

6. Арифметический оператор присваивания

В результате выполнения программы значения отдельных переменных изменяются в соответствии с алгоритмом решения задачи.

Наиболее употребительным приемом изменения значения переменной является использование оператора присваивания.

Общая форма записи арифметического оператора присваивания:

m LET α=β

Здесь m — номер строки, LET (пусть) — служебное слово, α — идентификатор переменной, = — символ присваивания, β — арифметическое выражение.

Исполнение оператора сводится к следующему: вычисляется значение выражения, записанного справа от символа «=», затем

полученное значение присваивается переменной, записанной слева от символа «=», т. е. найденное значение засылается в ячейку памяти машины. Эта ячейка автоматически выделяется для хранения значения переменной, указанной в операторе. Следовательно, к моменту исполнения оператора значения всех переменных, входящих в арифметическое выражение, должны быть определены.

Примеры записи оператора присваивания:

```
30 LET A=3.4
40 LET B1=-0.7
50 LET X=A*COS(B1)+3
```

Упражнение 1

Записать на языке БЕЙСИК следующие выражения:

1) $-2{,}138xy$.
2) $7{,}3\,\frac{ax^2}{cy^3}$.
3) $-3{,}65\cdot10^{-5}\,\frac{\sqrt{x^2+5}}{y^{3/2}}$.
4) $(\sin^2 x^2+e^{3a})^{2/3}$.
5) $e^{\sqrt{2\sin 3x}}$.
6) $\sin xe^{x\sqrt{1+x^2}}\lg y$.
7) $\frac{ax^2-\cos\sqrt{x^4+1}+b}{bx+2\ln|x|+3}$.

7. Задание исходных значений

При исполнении программ все переменные к моменту их использования должны быть определены, т. е. им должны быть заданы числовые значения. Рассмотрим вначале несколько простейших приемов задания значений переменным.

I. Задание значений переменным с помощью операторов присваивания.

Задача. Присвоить переменной x значение 0,5, затем вычислить $\sin x$ и присвоить его значение переменной y.

Решение.
```
10 LET X=0.5
20 LET Y=SIN(X)
```

II. Задание значений с помощью оператора INPUT.

Оператор INPUT используется для задания значений в режиме диалога. Общая форма записи оператора:

m INPUT α, β, ..., γ

Здесь m — номер строки, INPUT (ввести) — служебное слово, α, β, ..., γ — список идентификаторов. При исполнении оператора INPUT машина приостанавливает работу, а на экране высвечи-

вается символ «?». Пользователь должен набрать на клавиатуре нужное число (если чисел несколько, то они разделяются запятыми) и нажать клавишу [ПС]. Последовательность набора чисел определяется списком переменных, указанных в операторе INPUT.

Примеры записи оператора INPUT:

```
30 INPUT K
60 INPUT B, C, L
90 INPUT 'ВВЕДИТЕ K=' K
```

8. Вывод информации

После получения результата его нужно вывести из памяти машины. Вывод информации осуществляется на экран дисплея или на магнитную ленту.

Результаты вычислений и пояснительные тексты выводятся на экран дисплея с помощью оператора PRINT.

Его форма записи:

m PRINT α, β, ..., γ

Здесь m — номер строки, PRINT (напечатать) — служебное слово, α, β, ..., γ — список идентификаторов переменных, значения которых нужно вывести. Список может содержать пояснительный текст, заключенный между символами ‘ ‘.

Примеры записи оператора PRINT:

```
40 PRINT 'КОРНИ ДЕЙСТВИТЕЛЬНЫЕ'
50 PRINT A
60 PRINT C; D; F
70 PRINT 'X='X; ': СУММА='S
```

Поясним результаты исполнения операторов. После исполнения оператора, отмеченного номером 40, на экране дисплея появится текст: КОРНИ ДЕЙСТВИТЕЛЬНЫЕ. После исполнения следующего оператора будет напечатано число — значение переменной A. В операторе, отмеченном номером 60, содержится три идентификатора. На экран выводятся три числа, расположенные в одну строку через пробелы. И наконец, при исполнении последнего оператора на экране будет изображено: X=, затем число, равное значению переменной x, после него будут располагаться точка с запятой, слово СУММА= и число, равное значению переменной S. Если предположить, что $x=5$, а $S=148$, то на экране дисплея появится запись:

X=5; СУММА=148

Рассмотрим некоторые дополнительные возможности оператора PRINT.

I. m PRINT выражение

При исполнении оператора на экране дисплея появится значение выражения.

Пример 1. 10 PRINT 2*3

На экране дисплея появится число 6.

Пример 2. 10 PRINT 3+LGT(100)

На экране дисплея появится число 5.

II. m PRINT

При исполнении оператора будет пропущена одна строка без печати. Оператор PRINT в такой форме удобно использовать для разделения записей.

III. Управление записью.

Для размещения записей экран дисплея условно делится на четыре зоны по 20 позиций в каждой. Это значит, что в одну строку могут быть записаны только четыре числа.

Переход на следующую зону определяется разделителем «,» (запятая).

Пример 3. 10 PRINT A, B, C, D, E

В результате исполнения этого оператора на одной строке экрана дисплея появятся значения переменных A, B, C и D, причем каждое в своей зоне по 20 позиций, а в первой зоне следующей строки появится значение переменной E.

Для более плотной записи используется разделитель «;» (точка с запятой), вызывающий переход на следующую позицию.

Пример 4. 10 PRINT A; B; C; D; E

В этом случае значения переменных A, B, C, D и E будут записаны на экране дисплея одно за другим в одну строку.

Для пропуска нескольких зон или позиций необходимо задать подряд соответствующие разделители «,» или «;».

Пример 5. 10 PRINT,, A, B

Перед записью значения переменной A будут пропущены две зоны.

Разделители используются и для управления записью. Отсутствие запятой или точки с запятой в конце списка вывода вызывает после записи переход к началу новой строки. Для продолжения записи в строке, сделанной предыдущим оператором PRINT, необходимо в конце списка вывода предыдущего оператора задать разделитель «,» или «;».

Пример 6.

Запись 10 PRINT A, 20 PRINT B эквивалентна следующей: 30 PRINT A, B

IV. Форматный вид записи.

В операторе PRINT с помощью специальных элементов формата можно задать требуемый для пользователя вид записи, называемый форматным.

Спецификация формата! $n_1 \cdot n_2$!, где n_1 и n_2 цифры, означает, что на экран выводится знак числа (плюс не изображается), n_1 цифр целой части, точка и n_2 цифр после точки.

Пример 7. 20 PRINT! 3.4! A; B; C

На запись значений каждой переменной отводится три позиции для целой части и четыре позиции для дробной части.

Форматный вид записи может задаваться как для отдельного оператора PRINT, так и для программы в целом. Указание о формате действует до замены его новым указанием.

Пример 8.

а)
```
10 LET X=3.85
20 LET B=—7.938
30 PRINT! 1.3! 'X=', X;
40 PRINT 'B=', B
50 END
RUN
```

б)
```
10 LET X=3.85
20 LET B=—7.938
30 PRINT! 1.3! 'X='.X;
'B=', B
50 END
RUN
```

Результат выполнения программ одинаков:

X=3.85 B=—7.938

9. Примеры линейных программ

Рассмотренных операторов достаточно для записи программ, реализующих линейные алгоритмы.

Задача 1. Написать программу определения расстояния между двумя точками, заданными своими координатами $A(x_1, y_1)$, $B(x_2, y_2)$.

Решение. Схема алгоритма решения задачи изображена на рисунке 8.

Для исполнения алгоритма возьмем следующие значения координат:

$$x_1=3{,}2,\ y_1=-1{,}7,\ x_2=6{,}1,\ y_2=3{,}2.$$

Буквы, приведенные в схеме, используем в качестве идентификаторов. Зададим исходные значения переменным с помощью операторов присваивания. Программа будет иметь вид:

```
10 REM РАССТОЯНИЕ МЕЖДУ ТОЧКАМИ A(X1,Y1)
11 REM И B(X2,Y2)
12 LET X1=3.2
13 LET Y1=-1.7
14 LET X2=6.1
15 LET Y2=3.2
20 LET K=(X1-X2)^2
30 LET L=(Y1-Y2)^2
40 LET D=SQR(K+L)
50 PRINT D
60 END
RUN
```

На экране появится сообщение:

5.693856338
ОСТАНОВ В СТРОКЕ 60

Оператор REM — неисполняемый оператор. Он используется для записи комментариев к программе.

Оператор END является последним оператором программы и указывает на ее конец.

Эту же программу можно записать иначе, если ввести исходные данные посредством оператора INPUT. Программа будет иметь вид:

```
10 REM РАССТОЯНИЕ МЕЖДУ ТОЧКАМИ
11 REM A(X1,Y1) И B(X2,Y2)
12 INPUT X1,X2,Y1,Y2
20 LET K=(X1-X2)^2:LET L=(Y1-Y2)^2
40 LET D=SQR(K+L)
50 PRINT D
60 END
RUN
```

При исполнении программы машина приостановится после оператора INPUT. На экране высветится символ «?». На клавиатуре дисплея нужно набрать соответствующие числа (координаты двух точек), разделяя их запятой. Наберем, например,

4, 5, 3, 2

После нажатия клавиши [ПС] на экране высвечивается сообщение

3.162277660
ОСТАНОВ В СТРОКЕ 60

Упражнение 2

1. Написать программы вычисления функций. Значения аргументов определите с помощью операторов присваивания или INPUT:

а) $y=(2x^3+\sin^2(a+x))(x+e^{-ax})$;

б) $y=(b\cdot\sqrt{|x|}+\lg|a|)(\sin x^2+b^2)$;

в) $y=\dfrac{(9{,}87+a)(3{,}4\cdot 10^2+b\ \mathrm{tg}\ a)}{b^2+1}$.

2. Заданы координаты точек A и B. Написать программу вычисления координат точки C, симметричной точке A относительно B.

3. Написать программу вычисления площади полной поверхности и объема конуса, если известны: а) длина образующей и величина угла наклона ее к плоскости основания; б) длина высоты конуса и величина угла наклона образующей к плоскости основания.

4. Радиус шара задан координатами точек A и B. Написать программу вычисления площади поверхности и объема шара.

10. Условные и безусловные переходы

С помощью рассмотренных ранее операторов языка БЕЙСИК можно описать линейные алгоритмы, в которых содержатся информационные блоки и вычисления по формулам.

Для реализации алгоритмов, содержащих структуры РАЗВИЛКА или ПОВТОРЕНИЕ, используются операторы условного и безусловного переходов.

1. Оператор безусловного перехода. Общая форма записи оператора

m GOTO m_1

Здесь m — номер строки, содержащей оператор, GOTO (идти к) — служебное слово, m_1 — номер строки, на которую осуществляется переход.

Примеры записи оператора GOTO:

70 GOTO 30
160 GOTO 250

2. Логические выражения. Условие P, содержащееся в блоке проверки условия, образуется из двух арифметических выражений с помощью символов операций отношения: $>$, $\geqslant$, $<$, $\leqslant$, $=$, $\neq$ и называется *логическим выражением.*

Пример A1>7
X+2<>12

Здесь символы «<>» соответствуют символу «≠», т. е. $x+2\neq 12$. Логическое выражение может быть истинным или ложным. Если оно истинно, то управление передается по стрелке «да», если ложно,— по стрелке «нет».

3. Условный оператор. Общая форма записи оператора:

m IF P GOTO m_1
m IF P THEN m_1

Здесь IF (если) — служебное слово; P — логическое выражение, GOTO или THEN (иначе) — служебные слова, m_1 — номер строки, на которую осуществляется передача управления, если значение P истинно. Если значение P ложно, то оператор GOTO m_1 (или THEN m_1) пропускается и выполняется строка программы, следующая непосредственно за строкой, помеченной номером m.

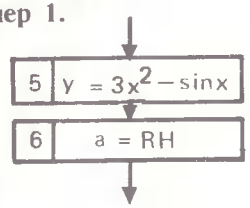
Напомним, что в программах несколько операторов можно записывать в одной строке, отделяя их один от другого символом «:» (двоеточие).

11. Реализация базовых структур на языке БЕЙСИК

Во всех примерах этого параграфа приводится фрагмент схемы алгоритма и соответствующая этому фрагменту последовательность операторов.

1. СЛЕДОВАНИЕ. Эта структура реализуется арифметическими операторами присваивания, содержащими символы БЕЙСИКа.

Пример 1.



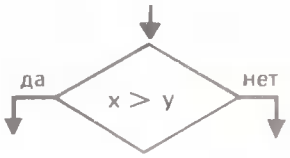
50 LET Y=3 ∗ X→2 − SIN (X)

60 LET A=R ∗ H

Структура СЛЕДОВАНИЕ может содержать информационный блок. Реализация таких блоков была показана в пп. 6—8 и может осуществляться с помощью операторов присваивания INPUT и PRINT. Более сложные конструкции блоков рассмотрены ниже.

2. РАЗВИЛКА. В структурах, содержащих блок проверки условия, после условного оператора будем записывать операторы, реализующие блок, следующий по стрелке «да».

В структуре РАЗВИЛКА блок проверки условия реализуется оператором IF. Для выполнения оговоренного выше условия необходимо символ операции отношения изменить на противоположный. Тогда блоку проверки условия



можно поставить в соответствие оператор

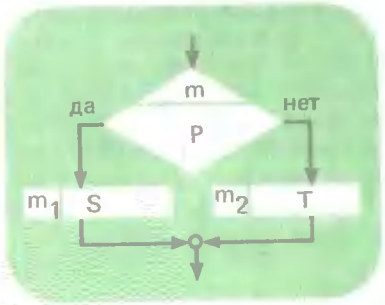
m IF X⩽Y GOTO m_1

Если значение отношения $x>y$ ложно, то $x\leqslant y$ истинно. Носителем номера m_1 является строка, в которой записан оператор, первым следующий по стрелке «нет».

Программные фрагменты, соответствующие структурам ПОЛНАЯ РАЗВИЛКА и НЕПОЛНАЯ РАЗВИЛКА приведены на рисунках 58 и 59. Здесь $\overline{P}$ — высказывание, противоположное P; S и T — операторы, содержащиеся в функциональных блоках и записанные на языке БЕЙСИК; номер строки m соответствует номеру блока проверки условия, номер m_1 — номеру первого блока, следующего по стрелке «да», номер m_2 — номеру блока, следующего по стрелке «нет», номер m_3 соответствует номеру блока, следующего непосредственно за объединяющим блоком.

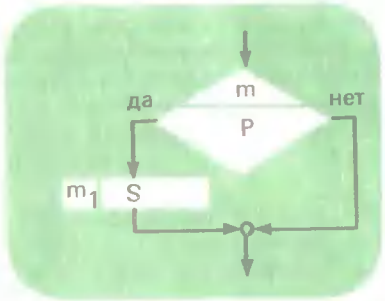
На рисунках 60 и 61 приведены фрагменты схем алгоритмов и соответствующие им программы.

3. ПОВТОРЕНИЕ. Блок проверки условия реализуется как и в структуре РАЗВИЛКА.



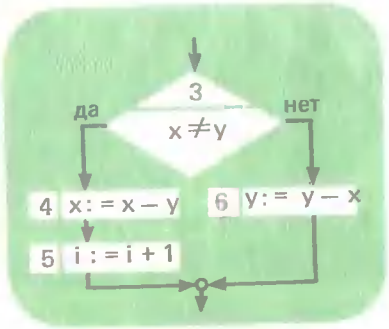

```
m    IF P̄ GOTO m2
m1   S : GOTO m3
m2   T
m3 ...
```

Рис. 58



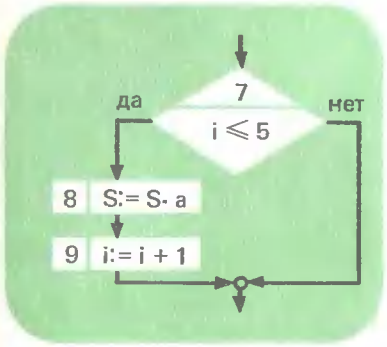

```
m    IF P̄ GOTO m3
m1   S
m3 ...
```

Рис. 59



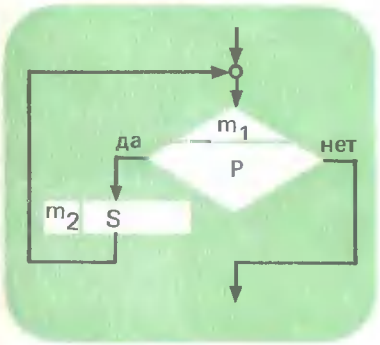

```
30   IF X = Y GOTO 60
40   LET X = X – Y
50   LET I = I + 1 : GOTO 70
60   LET Y = Y – X
70 ...
```

Рис. 60



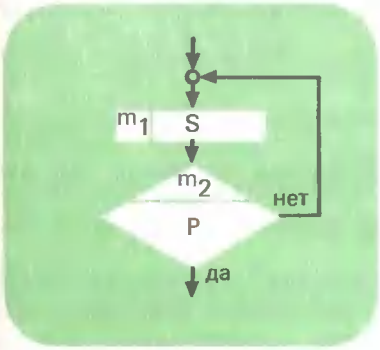

```
70 IF I > 5 GOTO 100
80   LET   S = S*A
90   LET   I = I + 1
100. . .
```

Рис. 61




```
m1   IF P̄ GOTO m3
m2   S : GOTO m1
m3 . . .
```

Рис. 62




```
m1   S
m2   IF P̄ GOTO m1
```

Рис. 63

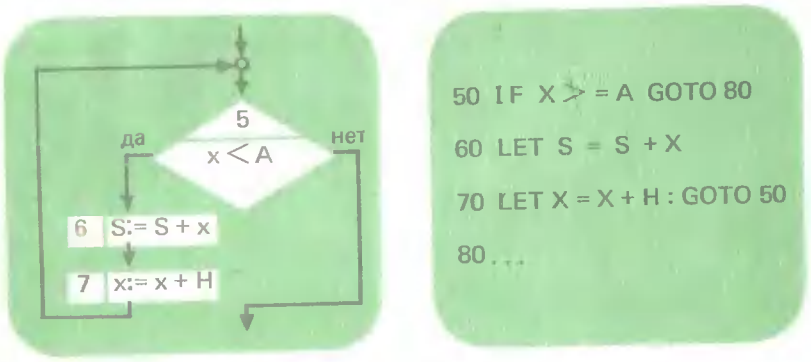


Рис. 64

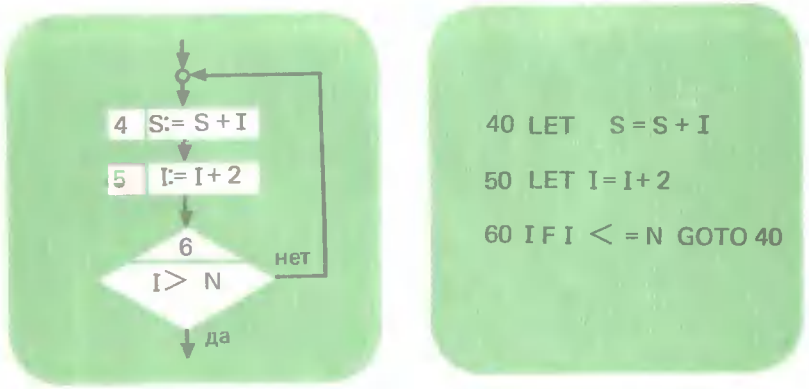


Рис. 65

Программные фрагменты, реализующие структуры цикл-ПОКА и цикл-ДО, приведены на рисунках 62 и 63.

На рисунках 64 и 65 приведены фрагменты схем алгоритмов и соответствующие им программы.

12. Запись структурированных программ

Реализуем несколько схем алгоритмов программ на языке БЕЙСИК.

Задача 1. Написать программу решения задачи 3 из введения.

Решение. Схема алгоритма решения этой задачи изображена на рисунке 10. Выделив в схеме базовые структуры и заменив их программными фрагментами, приведенными в п. 10, запишем программу:


```
5 REM ВЫЧИСЛЕНИЕ ФУНКЦИИ
6 REM Y=(X+7)/(X^2-5*X+6)
10 INPUT 'ВВЕДИТЕ X='X
20 LET A=X*X-5*X+6
30 IF A=0 GOTO 70
40 LET B=X+7
50 LET Y=B/A
60 PRINT 'X='X;'Y='Y:GOTO 80
70 PRINT 'ПРИ X=',X,'ЗНАЧЕНИЕ Y НЕ ОПРЕДЕЛЕНО'
80 END
RUN
```

После исполнения оператора INPUT машина приостановится. На экране появится текст:

ВВЕДИТЕ X=

На клавиатуре дисплея нужно набрать число — значение переменной X. Наберем, например, 7. Нажмем клавишу [ПС]. На экране высветится:

X=7.000000000 Y =7.000000000E—0,1
ОСТАНОВ В СТРОКЕ 80

Если после сообщения ВВЕДИТЕ X= ввести число 2, то на экране появится сообщение:

ПРИ X=2 ЗНАЧЕНИЕ Y НЕ ОПРЕДЕЛЕНО
ОСТАНОВ В СТРОКЕ 80

Задача 2. Написать программу решения задачи 5 из введения.

Решение. На рисунке 12 приведена схема алгоритма решения задачи. Программа ее решения будет иметь вид:

```
5 REM РАСПОЛОЖЕНИЕ ТРЕХ ЧИСЕЛ
6 REM В ПОРЯДКЕ ВОЗРАСТАНИЯ
10 INPUT 'ВВЕДИТЕ A,B,C'A,B,C
20 IF A>=B GOTO 80
30 IF B>=C GOTO 50
40 PRINT !5.2!A,B,C:GOTO 130
50 IF A>=C GOTO 70
60 PRINT A,C,B:GOTO 130
70 PRINT C,A,B:GOTO 130
80 IF B>=C GOTO 120
90 IF C>=A GOTO 110
100 PRINT B,C,A:GOTO 130
110 PRINT B,A,C:GOTO 130
120 PRINT C,B,A
130 END
RUN
```

На экране высвечивается сообщение:

ВВЕДИТЕ A, B, C

Введем через запятую, например, 4, 5, 3. После нажатия клавиши [ПС] на экране появится результат:

3.00 4.00 5.00
ОСТАНОВ В СТРОКЕ 130

13. Операторы цикла

Если известно начальное и конечное значения параметра цикла и шаг приращения, то структуру ПОВТОРЕНИЕ можно реализовать с помощью операторов FOR и NEXT.

Общая форма записи оператора FOR:

m FORa=α TO β STEP γ

Здесь m — номер строки, FOR (для), TO (до), STEP (шаг) — служебные слова, a — идентификатор управляющей переменной, α — начальное значение, β — конечное значение, γ — шаг приращения управляющей переменной. α, β, γ могут быть выражениями, значения которых определены к началу исполнения цикла.

Последним оператором циклического участка программы является оператор NEXT, который имеет вид

m NEXT a

Здесь m — номер строки, NEXT (следующий) — служебное слово, a — идентификатор управляющей переменной.

При исполнении оператора FOR определяются параметры цикла: начальное значение, конечное значение и шаг приращения, управляющей переменной присваивается начальное значение.

Если параметры цикла заданы так, что достичь конечного значения невозможно (конечное значение больше начального при отрицательном шаге приращений или конечное значение меньше начального при положительном шаге приращения), то цикл не исполняется и управление передается строке, следующей за оператором NEXT этого цикла.

При исполнении оператора NEXT производится изменение значения управляющей переменной на величину шага приращения и проверяется выполнение условия выхода из цикла. Цикл прекращается, когда значение управляющей переменной оказывается строго больше (при положительном шаге приращения) или строго меньше (при отрицательном шаге приращения) конечного значения. Управление при этом передается строке, следующей за оператором NEXT.

Если условие выхода из цикла не выполнено, управление передается строке, следующей за оператором FOR этого цикла.

Задача. Написать программу формирования по заданному аргументу I, который изменяется от 1 до 15 с шагом 1, таблицы квадратов, кубов, обратных величин, корня квадратного из числа I.

Решение. Текст программы может быть таким:

```
10 FOR I=1 TO 15 STEP 1
20 PRINT !7.4!I;I^2;I^3;1/I;SQR(I)
30 NEXT I
40 END
```

В результате исполнения программы построчно записывается таблица из пяти столбцов: в первом столбце записываются значения аргумента I от 1 до 15 с шагом 1, во втором — квадраты чисел I, в третьем — кубы чисел I, в четвертом — обратные величины, в пятом — корни квадратные из чисел I. В соответствии с форматом !7.4! на запись каждого числа отводится 7 позиций на целую часть и 4 позиции на дробную часть. Символ «;» в операторе PRINT означает, что запись по зонам отменяется и все пять чисел записываются в одну строку в формате !7.4!.

Итоговая таблица будет иметь вид

1.0000	1.0000	1.0000	1.0000	1.0000
2.0000	4.0000	8.0000	.5000	1.4142
...	...	...	...	...

После выхода из цикла значение управляющей переменной сохраняется и может использоваться в программе. Не допускается переход к операторам цикла извне.

14. Ввод и вывод информации

I. Ввод информации с помощью операторов DATA и READ.

Оператор DATA является неисполняемым. С его помощью формируется блок данных — множество чисел (таблица, массив). Массив формируется перед началом исполнения программы.

Общая форма записи оператора:

m DATA числа, разделенные запятой.

Например,

30 DATA — 1, 0.75, 0.2E — 0.1

В исходной программе может быть несколько операторов DATA, необязательно следующих подряд. В этом случае блок данных формируется в порядке их следования.

Например,

10 DATA — 1, 0.75, 0.2E — 1
. . . . . .
100 DATA 0.2, 3.7, —10.75
. . . . . .
130 DATA 0.75E03, 13.13

Блок данных, сформированный этими тремя операторами, будет содержать 8 чисел, записанных в таком порядке: —1, 0.75, 0.2E—1, 0.2, 3.7, —10.75, 0.75E03, 13.13.

При исполнении программы числовая информация из блока данных выбирается с помощью оператора READ.

Например,

30 DATA —1, 3.7, 13.13, 0.75E3
40 READ X, Y, Z
50 READ A

В результате исполнения оператора, помеченного номером 40, переменной X будет присвоено значение — 1, переменным Y и Z —

соответственно 3.7 и 13.13. Число 0,75E3; оставшееся в блоке данных, может быть выбрано другим оператором READ, например, помеченным меткой 50, т. е. каждый последующий оператор READ продолжает выборку данных из следующей позиции, на которой закончилась выборка предыдущим оператором.

Блок данных допускает только последовательную выборку чисел. Нельзя, например, сразу выбрать пятое число. Вторичная выборка чисел из блока данных возможна только после восстановления значений в блоке данных. Для этого используется оператор RESTORE. Общая форма его записи:

m RESTORE

После его исполнения выборка чисел из блока данных начнется с самого первого числа.

Оператором READ пользуются для формирования массивов значений переменных.

II. Организация вывода информации с использованием стандартной функции TAB.

Для расширения возможностей языка БЕЙСИК введена функция TAB, которая используется с оператором PRINT. Ее можно применять для вычерчивания графика функции по точкам, для размещения текста в любом месте экрана. Функция TAB записывается в виде:

TAB (выражение) 'цепочка символов'

При исполнении функции TAB вычисляется значение выражения, расположенного после слова TAB. Затем находится целая часть этого выражения; это число из промежутка [1,80] рассматривается как координата (номер) позиции строки, начиная с которой будут выведены символы, расположенные за словом TAB в апострофах.

Пример 1.

10 PRINT TAB(15) '*'

При выполнении оператора в 15-й позиции строки будет помещен символ *.

Пример 2.

40 PRINT TAB(18) '10-й КЛАСС'

Начиная с 18-й позиции строки пишется текст: 10-й КЛАСС

Пример 3.

10 FOR X=—6 TO 6:PRINT TAB(X¬2)'*':NEXT X

В строке с номером 10 помещены три оператора: FOR, PRINT, NEXT, разделенные двоеточием. Исполняя их, ЭВМ изобразит на экране дисплея параболу с осью, параллельной оси X. Парабола будет представлена 13 точками, причем в качестве точки использован символ «*». При желании строку 10 можно записать в

виде трех строк — каждый оператор FOR, PRINT, NEXT в отдельной строке.

Пример 4.

```
10 PRINT TAB(21'> ТАБЛИЦА УМНОЖЕНИЯ'
20 PRINT
30 FOR K=0 TO 9
40 FOR I=0 TO 4
50 PRINT !2.0!K'*'I' ='K*I;
60 NEXT I
70 PRINT
80 NEXT K
90 PRINT
100 FOR K=0 TO 9
110 FOR I=0 TO 9
120 PRINT !2.0!K'*'I' ='K*I;
130 NEXT I
140 PRINT
150 NEXT K
160 END
RUN
```

Поясним программу. Результатом ее исполнения является запись на экране таблицы умножения. Оператор с номером 10 обеспечивает запись с 21-й позиции строки текста ТАБЛИЦА УМНОЖЕНИЯ. Группа операторов с номерами 30—80 вычисляет и записывает первые пять столбцов таблицы умножения:

0*0=0	0*1=0	0*2=0	0*3=0	0*4=0
1*0=0	1*1=1	1*2=2	1*3=3	1*4=4
...	...	...	...	...
4*0=0	4*1=4	4*2=8	4*3=12	4*4=16

Операторы 20 и 90 организуют пропуск пустых строк. С помощью операторов 100—130,150 выводится вторая часть таблицы умножения.

III. Программы с использованием массивов.

С каждым массивом связан идентификатор — имя массива (буква или буква и цифра). Для ссылки на нужный элемент массива достаточно указать идентификатор массива и индекс элемента в скобках. Если массив двумерный, то в скобках индексы разделяются запятой.

Например, A (7), C (K), M (7,3), B1 (K, L)

Значение индекса начинается с 0.

В БЕЙСИКе нет никаких ограничений на сложность индексных выражений. Однако практически сложнее, чем $\alpha_1 * \beta \pm \alpha_2$, индексное выражение не записывают. Здесь $\alpha_i (i=1.2)$ — числа, β — переменная.

Значения индексов после их вычисления округляются до ближайшего целого.

Для описания массивов используется оператор DIM. Общая форма записи оператора DIM

m DIM $V_1(I_1)$, $V_2(I_2)$,, ..., $V_n(I_n)$

Здесь m — номер строки, DIM (размерность) — служебное слово, V_i — идентификатор, l_i — границы массива, т. е. максимальные значения индексов.

Задача 1. Задан массив *A*, содержащий *n* элементов. Составить программу вычисления суммы квадратов этих элементов.

Решение. Схема алгоритма решения задачи изображена на рисунке 17. Соответствующая ей программа будет иметь вид:

```
1 REM СУММА КВАДРАТОВ N ЭЛЕМЕНТОВ
2 REM МАССИВА A
10 DIM A(100)
11 INPUT 'ВВЕДИТЕ N НЕ БОЛЬШЕ 8'N
12 FOR I=0 TO N
13 READ A(I)
14 NEXT I
15 DATA -1.5,1.1,0.8,3.4,7.1,8.5,-1.2,3.1,6.2
20 LET S=0
30 FOR I=1 TO N
40 LET S=S+A(I)^2
60 NEXT I
70 PRINT !3.2!'S=';S
80 END
RUN
```

Здесь рассматривается массив *A*, содержащий 9 элементов. БЕЙСИК позволяет вести счет элементов от нуля. Элементы массива описаны оператором DATA. Ввод элементов и организация массива осуществляется операторами 12—14. После исполнения команды RUN (пуск) на экране высветится сообщение

ВВЕДИТЕ N НЕ БОЛЬШЕ 8

Введем, например, 3. После нажатия клавиши [ПС] на экране высветится сообщение

S=13.41
ОСТАНОВ В СТРОКЕ 80

Будет найдена сумма квадратов чисел 1,1; 0,8; 3,4.

IV. Примеры организации вывода информации.

При выводе в каждой позиции может быть записан один символ. Пробел — пропуск позиции, также рассматривается как символ.

Печать по зонам обеспечивается использованием запятой в качестве разделителя между элементами списка вывода в операторе PRINT. Каждая запятая вызывает перемещение курсора к началу следующей зоны. Для отмены печати по зонам нужно в списке вывода оператора PRINT использовать другой разделитель — точку с запятой. Символ «;» в операторе PRINT вызывает пропуск одной позиции и новую запись в той же строке (и, возможно, в той же зоне).

Задача 2. Написать программу вычисления и вывода на печать таблицы значений функции $y=\sin x$, при $0\leqslant x\leqslant \pi/2$ с шагом 0,1.

Решение. Оформим решение задачи одной строкой:

```
10 FOR X=0 TO #PI/2 STEP .1:PRINT !1.4!X,SIN(X):NEXT X
```

В строке 10 помещены три оператора: FOR, PRINT, NEXT, разделенные двоеточием. Используется число π, приближенное значение которого хранится в памяти ЭВМ: π=3,141592654.

Для вызова числа π необходимо использовать имя #PI.

Программа обеспечивает печать таблицы значения функции $y=\sin x$ для x от 0 до $\pi/2$ с шагом 0.1. Значения аргумента и функции даются в формате !1.4!, т. е. одна позиция на целую часть и четыре позиции на дробную часть. Будет напечатано 16 значений.

Задача 3. Написать программу решения задачи 7 из введения.

Решение. Схема алгоритма решения задачи приведена на рисунке 14. Запишем соответствующую ей программу:

```
4 REM ВЫЧИСЛЕНИЕ ФУНКЦИИ
10 INPUT 'ВВЕДИТЕ X0,H,N'X0,H,N
20 PRINT TAB(7)'I',TAB(28)'X',TAB(47)'Y'
25 FOR I=3 TO 50:PRINT TAB(I)'-';:NEXT I:PRINT
30 FOR I=1 TO N STEP 1
40 LET X=X0+I*H
50 IF X>=0 GOTO 70
60 LET Y=2-7*X:GOTO 100
70 IF X<=2 GOTO 90
80 LET Y=X:GOTO 100
90 LET Y=X-2
100 PRINT!5.4!I,X,Y
120 NEXT I
130 END
RUN
```

После выполнения команды RUN на экране высветится сообщение:

ВВЕДИТЕ X0, H, N

Наберем на клавиатуре дисплея значения, например 0, 0.1, 10. После нажатия клавиши [ПС] на экране высветится заголовок таблицы и значения переменных

I	X	Y
1.0000	.1000	—1.9000
2.000	.2000	—1.8000
...	...	...
10.0000	1.0000	—1.0000

Задача 4. Написать программу решения задачи 11 из введения.

Решение. На рисунке 20 изображена схема алгоритма решения задачи. Запишем программу решения задачи.

```
1 REM РАСПОЛОЖЕНИЕ ЭЛЕМЕНТОВ
2 REM МАССИВА A В ПОРЯДКЕ ВОЗРАСТАНИЯ
10 DATA 12,6,-8,16,0,1,5
11 DIM A(10)
```

```
13 PRINT 'ВВЕДИТЕ N НЕ БОЛЬШЕ  6'
14 INPUT 'N='N
15 FOR I=0 TO N
16 READ A(I)
17 NEXT I
20 FOR I=0 TO N-1
30 FOR J=I+1 TO N
40 IF A(I)<=A(J) GOTO 90
50 LET R=A(I): LET A(I)=A(J)
70 LET A(J)=R
90 NEXT J
110 NEXT I
120 FOR K=0 TO N
121 PRINT!4.0!A(K);
122 NEXT K
130 END
RUN
```

После исполнения команды RUN на экране высветится сообщение:

ВВЕДИТЕ N НЕ БОЛЬШЕ 6

Наберем на клавиатуре дисплея, например, 6. После нажатия клавиши [ПС] на экране через небольшой промежуток времени высветится последовательность чисел

—8 0 1 5 6 12 16

ОСТАНОВ В СТРОКЕ 130

15. Подпрограммы

Одна и та же последовательность операторов может много раз повторяться в разных местах программы, может встречаться в различных программах, никак между собой не связанных. Например, последовательность операторов, реализующих вычисление логарифмической, тригонометрических функций и др. При составлении программ можно каждый раз повторить эту последовательность операторов. Однако это увеличивает объем программы и является дополнительным источником ошибок. Поэтому поступают так: выделяют эту последовательность операторов из программы и записывают в памяти машины отдельно, а в программе на соответствующем месте указывают лишь оператор обращения к этой выделенной последовательности операторов. Такой независимый элемент программы называют *подпрограммой*. Программу, из которой обращаются к подпрограмме, называют *головной программой*.

Подпрограмма завершается оператором RETURN (возврат). Общая форма записи оператора RETURN:

m RETURN

В качестве первого оператора подпрограммы обычно используют неисполняемый оператор REM.

Обращение к подпрограмме осуществляется оператором GOSUB. Общая форма записи оператора GOSUB:

m GOSUB m_1

Здесь m — номер строки, GOSUB (идти к подпрограмме) — служебное слово, m_1 — номер первой строки подпрограммы.

Оператор GOSUB передает управление строке, помеченной номером m_1.

После исполнения подпрограммы по оператору RETURN происходит возврат в головную программу. Вычисления в головной программе продолжаются со строки, следующей за оператором GOSUB.

Задача 1. Составить программу для вычисления числа сочетаний по формуле:

$$C_n^m = \frac{n!}{m!(n-m)!}, \quad (n > m).$$

Решение. В формуле несколько раз встречается вычисление факториала. Целесообразно поэтому эти вычисления оформить в виде подпрограммы. Обозначим N1=n! M1=m! C=C_n^m

```
1000 REM ВЫЧИСЛЕНИЕ ФАКТОРИАЛА ЧИСЛА
1010 LET F=0
1020 FOR I=1 TO L
1030 LET F=F*I
1040 NEXT I
1050 RETURN
1060 END
5 PRINT 'N БОЛЬШЕ M'
7 PRINT 'ВВЕДИТЕ 0<N<=69 И M>0'
10 INPUT 'ВВЕДИТЕ N,M'N,M
11 PRINT !4.2!
20 LET L=N
30 GOSUB 1000
40 LET N1=F
50 LET L=M
60 GOSUB 1000
70 LET M1=F
80 LET L=N-M
90 GOSUB 1000
100 LET C=N1/M1/F
110 PRINT 'C(';N;',';M;')=';C
120 END
RUN
```

После исполнения команды RUN на экране появится сообщение:

N БОЛЬШЕ M

ВВЕДИТЕ 0<N<=69 И M>0

ВВЕДИТЕ N, M

Если на клавиатуре набрать числа 5 и 3, разделив их запятой, то после нажатия клавиши [ПС] на экране высветится сообщение:

C (5.00; 3.00) =10.00

ОСТАНОВ В СТРОКЕ 120

Задача 2. Составить программу упорядочения n элементов массива А в порядке возрастания.

Решение. Воспользуемся схемами алгоритмов, изображенными на рисунках 21 и 22 (см. задачу 11 из введения).

```
1  REM РАСПОЛОЖЕНИЕ ЭЛЕМЕНТОВ
2  REM МАССИВА В ПОРЯДКЕ
3  REM ВОЗРАСТАНИЯ
4  REM ГОЛОВНАЯ ПРОГРАММА
10 DIM A(6)
11 DATA 6,5,1,-8,16,12,0
12 PRINT 'ВВЕДИТЕ N НЕ БОЛЬШЕЕ 6'
13 INPUT 'N='N
14 FOR K=0 TO N
15 READ A(K)
16 NEXT K
20 FOR K=0 TO N-1
40 GOSUB 1000
50 LET R=A(K)
60 LET A(K)=A(L)
70 LET A(L)=R
80 NEXT K
90 PRINT 'УПОРЯДОЧЕННЫЙ МАССИВ'
91 FOR I=0 TO N:PRINT !4.0!A(I);:NEXT I
100 END
1000 REM ПОДПРОГРАММА
1010 LET M=K
1020 LET L=M:LET B=A(M)
1040 FOR I=M+1 TO N
1050 IF B<=A(I) GOTO 1070
1060 LET B=A(I):LET L=I
1070 NEXT I
1080 RETURN
1090 END
RUN
```

Подпрограммы могут содержать обращения к другим подпрограммам. При такой записи внутренняя подпрограмма называется *вложенной*.

Упражнение 3

1. Написать программу вычисления значения функций:

а) $v = \begin{cases} x^2-3x+2, & \text{если } -5<x\leqslant 3, \\ \frac{1}{x}+x^2, & \text{если } x\leqslant -5 \text{ или } x>3; \end{cases}$

б) $c=\begin{cases} 2\cos x-\lg\frac{|x|}{4}+5, & \text{если } x<1, \\ e^{-x^2}+\sqrt{\frac{x+1}{x+2}}+x^3, & \text{если } 1\leqslant x<4, \\ \ln x, & \text{если } x\geqslant 4. \end{cases}$

2. Написать программу исследования числа решений системы уравнений:

а) $\begin{cases} y=a_1x+b_1, \\ y=a_2x+b_2; \end{cases}$ б) $\begin{cases} a_1x+b_1y=c_1, \\ a_2x+b_2y=c_2. \end{cases}$

3. Написать программу вычисления величины $S=\sum_{k=1}^{8} k!$.

4. Написать программу вычисления n первых членов: а) арифметической прогрессии, если заданы первый член и разность прогрессии; б) геометрической прогрессии, если заданы первый член и знаменатель прогрессии.

5. Задан одномерный массив A вещественных чисел. Написать программу:

а) подсчета числа положительных и суммы отрицательных элементов в массиве;

б) замены всех элементов массива, меньших 37,3, числом 37,3;

в) замены местами наибольшего и наименьшего элементов;

г) подсчета числа элементов, принадлежащих отрезку $[B, C]$;

д) построения массива C, элементы которого определяются следующим образом:

$$c_k=\sum_{i=1}^{k} a_i, \ (k=1, 2, \ldots, n);$$

е) дающую ответ «да» или «нет» в зависимости от того, встречается или нет число 7 в массиве.

16. Функции пользователя

Если в разных местах программы вычисляются значения одного и того же арифметического выражения, то для сокращения записи это выражение можно выделить и записать один раз в виде выражения — функции (иначе, функции пользователя). В программе на соответствующих местах записывается оператор обращения к этой функции.

Функции пользователя описывают с помощью оператора DEF (определять), сообщают ей имя, указывают параметр и выражение.

Общая форма записи оператора DEF:

m DEF FN α (A) = β

Здесь m — номер строки, DEF — служебное слово, FN — неизменяемая часть имени функции, α — изменяемая часть имени функции — одна прописная буква латинского алфавита, A — идентификатор простой переменной (формальный параметр), β — арифметическое выражение, которое может содержать обращения к другим функциям пользователя, исключая данную.

Формальные параметры должны быть определены до обращения к функции пользователя.

Пример 1.

```
10 DEF FNF (Y) = Y↑2 + Y + 2
20 LET A = FNF (3)
```

В строке, помеченной номером 10, записана функция $f(y)=$ $=y^2+y+2$. Обращение к функции пользователя осуществляется в следующей, помеченной номером 20, строке. В результате исполнения действий этой строки переменной А будет присвоено значение, равное 14.

Пример 2.

```
30 DEF FNB (X) = SIN (X) + COS (Y)
40 LET Y = 1.1 : LET A = FNB (0.5)
```

В результате исполнения этой последовательности операторов переменной А будет присвоено значение выражения $\sin(0.5)+\cos(1.1)$.

Пример 3.

```
10 DEF    FNC (X) = X ¬ 2
20 PRINT FNC (2 + FNC (2) )
30 END
```

Результатом исполнения записанной последовательности операторов будет запись на экране дисплея числа 36.

17. Тексты программ решения задач

Приведенные тексты программ решения задач повышенной трудности предлагаются для самостоятельного разбора.

Задача 1. Определение первых N простых чисел.

Решение. Просматривается множество натуральных чисел в порядке их возрастания и выбираются те из них, которые являются простыми. Просмотр заканчивается, когда выбирается N-е простое число. Проверка числа R на «простоту» осуществляется проверкой: делится оно или нет на простые числа, меньшие $\sqrt{R}$. Если число R не делится ни на одно простое число, меньшее $\sqrt{R}$, то оно простое. Первые три простых числа 1, 2 и 3 задаются. Остальные простые числа могут быть среди нечетных натуральных чисел.

Текст программы решения задачи:

```
1 REM ПЕРВЫЕ N ПРОСТЫХ ЧИСЛА
2 REM АЛГОРИТМ: ПРОСМАТРИВАЮТСЯ НАТУРАЛЬНЫЕ ЧИСЛА В
3 REM ПОРЯДКЕ ИХ ВОЗРАСТАНИЯ И ВЫБИРАЮТСЯ ТЕ ИЗ НИХ,
4 REM КОТОРЫЕ ЯВЛЯЮТСЯ ПРОСТЫМИ.
5 REM K- НОМЕР ПОСЛЕДНЕГО НАЙДЕННОГО ПРОСТОГО ЧИСЛА
6 REM R- ТЕКУЩЕЕ, ПРОВЕРЯЕМОЕ НА "ПРОСТОТУ" ЧИСЛО.
7 REM P- ИМЯ ТАБЛИЦЫ ПРОСТЫХ ЧИСЕЛ
8 REM H- НОМЕР ОЧЕРЕДНОГО ПРОСТОГО ЧИСЛА ИЗ P.
9 REM P1- ЗНАЧЕНИЕ ОЧЕРЕДНОГО ПРОСТОГО ЧИСЛА ИЗ P.
10 REM P2- ОЧЕРЕДНОЕ НАЙДЕННОЕ ПРОСТОЕ ЧИСЛО.
11 REM L- ЛОГИЧЕСКАЯ ПЕРЕМЕННАЯ, УПРАВЛЯЮЩАЯ РАБОТОЙ
12 REM ЦИКЛА ПО ЗАПИСИ ПРОСТОГО ЧИСЛА В P.
13 REM D- ЛОГИЧЕСКАЯ ПЕРЕМЕННАЯ,УЧАСТВУЮЩАЯ В
14 REM УПРАВЛЕНИИ ЦИКЛОМ ПО ПРАВЕРКЕ ЧИСЛА НА "ПРСТОТУ"
15 REM Q- ДРОБНАЯ ЧАСТЬ ПРИ ДЕЛЕНИИЧИСЕЛ P И P1.
18 DIM P(200)
19 PRINT 'ГОТОВА НАЙТИ НЕ БОЛЕЕ 200 ПРОСТЫХ ЧИСЕЛ'
```



```
20 INPUT 'УКАЖИТЕ ИНТЕРЕСУЮЩЕЕ ВАС КОЛИЧЕСТВО ЧИСЕЛ' N
21 PRINT 'СПАСИБО! МИНУТКУ!'
30 LET P(1)=1
40 LET P(2)=2
50 LET P(3)=3
60 LET R=3
70 LET K=3
80 IF K>=N GOTO 240
90 LET L=1
100 IF L><1 GOTO 220
110 LET R=R+2
120 LET H=2
130 D=1
140 IF P(A)>SQR(R) GOTO 190
141 IF D><1 GOTO 190
150 LET P1=P(H)
160 LET Q=R/P1-INT(R/P1)
161 IF Q><0 GOTO 180
170 LET D=0: GOTO 140
180 LET H=H+1: GOTO 140
190 IF D><1 GOTO 100
200 LET P2=R
210 LET L=0: GOTO 100
220 LET K=K+1
230 LET P(K)=P2: GOTO 80
240 PRINT 'ГОТОВО!'
241 PRINT 'ЕСЛИ ХОТИТЕ ВЫВЕСТИ ПОЛУЧЕННУЮ ТАБЛИЦУ,
242 PRINT 'ТО ВВЕДИТЕ 1, ИНАЧЕ - 0'
243 INPUT W
244 IF W=0 GOTO 300
245 LET I=1: LET K=20
247 IF K>N GOTO 260
248 FOR L=I TO K
249 PRINT !7.0!P(L)
250 NEXT L
251 PRINT 'ПРОДОЛЖАТЬ ВЫВОДИТЬ ЧИСЛА?'
252 PRINT 'ЕСЛИ ДА, ТО ВВЕДИТЕ 1, ИНАЧЕ - 0'
253 INPUT W
254 IF W=0 GOTO 300
255 LET I=I+20: LET K=K+20: GOTO 247
260 FOR M=I TO N
261 PRINT !7.0!P(M)
262 NEXT M
263 PRINT 'ТАБЛИЦА P ВЫВЕДЕНА ПОЛНОСТЬЮ'
264 PRINT 'И СОДЕРЖИТ '; N; 'ЧИСЕЛ'
300 END
RUN
```

Задача 2. Нахождение и запись простых чисел по правилу «решето Эратосфена».

Решение. Выписываются все натуральные числа, начиная с 1 до заданного числа N. Из этой последовательности затем исключаются все числа, кратные 2. Далее исключаются все числа, кратные 3. Этот процесс продолжают до тех пор, пока не выполнится неравенство $A_i^2>N$, где A_i — простое число.

Текст программы решения задачи:

```
10 PRINT 'ПРОСТЫЕ ЧИСЛА - РЕШЕТО ЭРАТОСФЕНА'
15 PRINT 'НАХОДИМ ПРОСТЫЕ ЧИСЛА,'
16 PRINT 'НЕ ПРЕВОСХОДЯЩИЕ ЧИСЛА N'
20 INPUT 'ВВЕДИТЕ N='N
```

```
30 DIM A(N-2)
40 REM ФОРМИРОВАНИЕ ИСХОДНОГО МАССИВА A(I)
50 LET A(0)=2
60 FOR I=1 TO N-2
70 LET A(I)=A(I-1)+1
80 NEXT I
85 REM ПРЕОБРАЗОВАНИЕ ИСХОДНОГО МАССИВА
90 FOR J=0 TO N-2
100 IF A(J)=0 GOTO 160
110 IF A(J)^2>N GOTO 170
120 LET L=A(J)+J
130 FOR P=L TO N-2 STEP A(J)
140 LET A(P)=0
150 NEXT P
160 NEXT J
170 REM ПЕЧАТЬ МАССИВА ПРОСТЫХ ЧИСЕЛ
180 PRINT!4.0! 'ПРОСТЫЕ ЧИСЛА НЕ ПРЕВОСХОДЯТ N='N
190 FOR M=0 TO N-2
200 IF A(M)=0 GOTO 220
210 PRINT!4.0!; A(M)
220 NEXT M
230 END
```

Задача 3. Отгадывание задуманного числа.

Решение. Текст программы решения задачи:

```
10 PRINT 'ИГРА В ОТГАДЫВАНИЕ ЗАДУМАННОГО ЧИСЛА'
20 PRINT 'ЗАДУМАЙТЕ НАТУРАЛЬНОЕ ЧИСЛО НЕ БОЛЕЕ 1000'
30 PRINT 'И ОТВЕЧАЙТЕ НА ВОПРОСЫ'
40 LET Y=0
50 LET I=0
60 PRINT 'ЕСЛИ ЗАДУМАННОЕ ЧИСЛО >=2, ТО ВВЕДИТЕ 0'
65 PRINT 'ИНАЧЕ ВВЕДИТЕ 1'
70 INPUT K
80 IF K=1 THEN Y=Y+1 : GOTO190
90 PRINT 'РАЗДЕЛИТЬ ЗАДУМАННОЕ ЧИСЛО НА 2'
95 PRINT 'ВВЕДИТЕ ОСТАТОК R'
100 INPUT 'R='R
110 LET Y=Y+R*2^I
120 LET I=I+1
130 PRINT 'ЕСЛИ ПОЛУЧЕННОЕ ЧАСТНОЕ >=2, ТО ВВЕДИТЕ 0'
135 PRINT 'ИНАЧЕ ВВЕДИТЕ 1'
140 INPUT K
150 IF K=1 THEN Y=Y+2^I : GOTO 190
160 PRINT 'РАЗДЕЛИТЕ ЧАСТНОЕ НА 2'
165 PRINT 'ВВЕДИТЕ ОСТАТОК R'
170 INPUT 'R='R
180 GOTO 110
190 PRINT 'ВЫ ЗАДУМАЛИ ЧИСЛО'
200 PRINT !4.0! 'Y='Y
201 PRINT 'ЕСЛИ Я ПРАВА, ТО ВВЕДИТЕ 1'
202 PRINT 'ИНАЧЕ ВВЕДИТЕ 0'
203 INPUT K
204 IF K=0 GOTO 210
205 PRINT 'БЛАГОДАРЮ !'
206 PRINT 'ВЫ НЕ УМЕЕТЕ ДЕЛИТЬ НА 2 И ОПРЕДЕЛЯТЬ ОСТАТОК.'
207 PRINT 'ХОТИТЕ СЫГРАТЬ ЕЩЕ?'
208 PRINT 'ЕСЛИ ДА, ТО НАЖМИТЕ НА КЛАВИШУ "ПС"'
209 PRINT 'И НАБЕРИТЕ СЛОВО "RUN"' : GOTO 213
210 PRINT 'ВЫ НЕ УМЕЕТЕ ДЕЛИТЬ НА 2'
211 PRINT 'ПОВТОРИТЕ, ПОЖАЛУЙСТА, РЕШЕНИЕ'
212 PRINT 'ДЛЯ ЭТОГО НАБЕРИТЕ СЛОВО "RUN"'
213 END
```


Задача 4. Определение четырехразрядного числа вида $aabb$, удовлетворяющего условию $aabb=(cd)^2$.

Решение. Текст программы решения задачи:

```
10 FOR I=1 TO 9
20 FOR J=0 TO 9
30 LET M=(I*100+J)*11
40 LET P=INT (SQR(M))
45 LET K=P^2
50 IF M-K<> GOTO 90
60 PRINT 'ЧИСЛО M РАВНО КВАДРАТУ ЧИСЛА P'
70 PRINT !4.0!'M='M
80 PRINT 'P='P
90 NEXT J
100 NEXT I
110 END
```

Задача 5. Вы забрели в сад, в котором имеются три калитки, и решили пройти через них, не пропустив ни одной. Набрав некоторое количество яблок, Вы отдали половину всех яблок и еще пол-яблока человеку у первой калитки, половину того, что осталось, и еще пол-яблока — человеку у второй калитки, а половину всех оставшихся яблок и еще пол-яблока — человеку у третьей калитки, и при этом не разрезали ни одного яблока. Сколько яблок Вы должны набрать в саду, чтобы вынести К яблок? Каково наименьшее число яблок надо собрать, чтобы выйти из сада?

Решение. Текст программы решения задачи:

```
10  PRINT 'ЗАДАЧА КОЛЬРИДЖА'
20  PRINT 'K - КОЛИЧЕСТВО ЯБЛОК,КОТОРОЕ'
30  PRINT 'ВЫ ХОТИТЕ ВЫНЕСТИ ИЗ САДА'
40  INPUT 'ВВЕДИТЕ K='K
50  FOR I=0 TO 2
60  LET X=2*K+1
70  LET K=X
80  NEXT I
90  PRINT !4.0! 'КОЛИЧЕСТВО ЯБЛОК,КОТОРОЕ'
100 PRINT 'НУЖНО СОБРАТЬ, РАВНО'X
110 END
```

Задача 6. Генерал решил устроить смотр своему войску, организовав парад. Но в последний момент один солдат заболел. Когда по традиции войско построилось в шеренги по десять человек в каждой, то оказалось, что в последней шеренге не хватает одного солдата. Пришлось перестроиться в шеренги по 11 человек, но и такие шеренги не подошли.

При каждом перестроении по 9, 8, 7, 6, 5, 4, 3, 2 человека в шеренге всегда оказывалось, что в последней шеренге не хватает одного солдата. Известно, что число участников парада не превосходило 7000. Сколько человек участвовало в параде?

Решение. Текст программы решения задачи:

```
10 PRINT 'ПРАЗДНИЧНЫЙ ПАРАД'
20 LET M=10
30 IF (M-INT(M/9)*9)<>0 GOTO 100
40 IF (M-INT(M/8)*8)<>0 GOTO 100
50 IF (M-INT(M/7)*7)<>0 GOTO 100
```

```
60 IF (M-INT(M/11)*11)=1 GOTO 100
70 LET K=M-1
80 PRINT !6.0!'ЧИСЛО УЧАСТНИКОВ ПАРАДА='K
90 GOTO 110
100 LET M=M+10: GOTO 30
110 END
```

Задача 7. Прогноз погоды на завтра.

Решение. Текст программы решения задачи:

```
10 PRINT 'ПРОГНОЗ ПОГОДЫ НА ЗАВТРА'
20 PRINT 'ДЛЯ ПРЕДСКАЗАНИЯ ПОГОДЫ НА ЗАВТРА'
30 PRINT 'НЕОБХОДИМО ВВЕСТИ ПЯТЬ ПОКАЗАТЕЛЕЙ'
40 PRINT 'L1 - СОСТОЯНИЕ ПОГОДЫ СЕГОДНЯ'
50 PRINT 'L1=0 - ХОРОШАЯ ПОГОДА'
60 PRINT 'L1=1 - ПЛОХАЯ ПОГОДА'
70 INPUT 'ВВЕДИТЕ L1='L1
80 PRINT 'L2 - ДАВНО ЛИ СТОИТ ТАКАЯ ПОГОДА'
90 PRINT 'L2=0 - НЕДАВНО'
100 PRINT 'L2=1 - ДАВНО'
110 INPUT 'ВВЕДИТЕ L2='L2
120 PRINT 'L3 - ПОКАЗАНИЯ БАРОМЕТРА'
130 PRINT 'L3=0 - ПОКАЗАНИЯ ПАДАЮТ'
135 PRINT 'L3=1 -ПОКАЗАНИЯ РАСТУТ'
140 PRINT 'L3=2 - НЕ ИЗМЕНЯЮТСЯ'
160 INPUT 'ВВЕДИТЕ L3='L3
170 PRINT 'L4 - КАК МЕНЯЮТСЯ ЭТИ ПОКАЗАНИЯ?'
180 PRINT 'L4=0 - МЕНЯЮТСЯ БЫСТРО'
190 PRINT 'L4=1 - МЕДЛЕННО'
200 PRINT 'L4=2 - ВООБЩЕ НЕ МЕНЯЮТСЯ'
210 INPUT 'ВВЕДИТЕ L4='L4
220 PRINT 'L5 - ВРЕМЯ ГОДА'
230 PRINT 'L5=0 - ЛЕТО'
240 PRINT 'L5=1 - ЗИМА'
250 PRINT 'L5=2 - ВЕСНА'
260 PRINT 'L5=3 - ОСЕНЬ'
270 INPUT 'ВВЕДИТЕ L5='L5
280 IF L3-2=0 THEN LET F=1: J=L1+1: GOTO 430
290 IF L2><0 THEN LET F=L4+1: GOTO 310
300 LET F=L2+L4
310 IF L3=0 GOTO 360
320 IF L5-3><0 THEN LET J=2-L1: GOTO 430
330 IF L1=0 THEN LET J=2-L1: GOTO 430
340 LET J=6
350 GOTO 430
360 IF L4><0 THEN LET J=L1+1: GOTO 430
370 IF L5=0 GOTO 410
380 IF L5-2>=0 THEN LET J=4: GOTO 430
390 LET J=5
400 GOTO 430
410 IF L1><0 THEN LET J=L1+1: GOTO 430
420 LET J=3
430 PRINT 'ПРОГНОЗ ПОГОДЫ НА ЗАВТРА'
440 IF J=1 THEN PRINT 'ПЛОХАЯ ПОГОДА': GOTO 500
450 IF J=2 THEN PRINT 'ХОРОШАЯ ПОГОДА': GOTO 500
460 IF J=3 THEN PRINT 'ЛИВЕНЬ, ГРОЗА': GOTO 500
470 IF J=4 THEN PRINT 'СИЛЬНЫЙ ВЕТЕР': GOTO 500
480 IF J=5 THEN PRINT 'ОТТЕПЕЛЬ, СНЕГОПАД': GOTO 500
490 PRINT 'СИЛЬНЫЙ ВЕТЕР'
500 IF F=0 THEN PRINT 'ПРОДЕРЖИТСЯ СОВСЕМ НЕДОЛГО': GOTO 530
510 IF F=1 THEN PRINT 'ПРОДЕРЖИТСЯ НЕСКОЛЬКО ДНЕЙ': GOTO 530
520 PRINT 'БУДЕТ СТОЯТЬ ПОСТОЯННО'
530 END
```

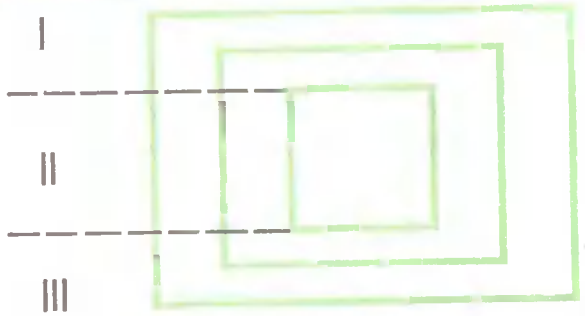



Рис. 66

Задача 8. Вычерчивание на экране дисплея «вложенных» прямоугольников.

Решение. Результатом исполнения этой программы в отличие от предложенных выше является не цифровое или текстовое сообщение на экране дисплея, а графический рисунок.

P — количество прямоугольников, которые можно изобразить на экране дисплея.

Учитывая размерность строки дисплея (80 символов) и расположение рисунка на экране дисплея, необходимо ограничить длину основания внешнего прямоугольника до 69, а количество вложенных прямоугольников — до 5. Высота последнего (самого внутреннего) прямоугольника не должна быть менее 3, иначе на экране он будет изображен в виде полосы.

Программа состоит из пяти частей. Первая часть (строки программы с номерами 9—22) организует ввод исходных данных: количество вложенных прямоугольников, основание внешнего прямоугольника, высоту последнего вложенного прямоугольника. Для простоты написания программы прямоугольники разбиты на три области. Рассмотрим конкретно случай трех прямоугольников (рис. 66).

Вторая часть программы вычерчивает область, помеченную цифрой 1 (строки программы 50—300); третья часть программы — область, помеченную цифрой II (строки 310—318); четвертая часть программы — область, помеченную цифрой III (строки 320—440); пятая часть программы (строки 550—680) содержит операторы подпрограмм для изображения соответствующих областей прямоугольников. Результат исполнения программы для $p=1$ и $p=5$ представлен соответственно на рисунках 67 и 68.

Рис. 67

Текст программы решения задачи:

```
9 PRINT'КОЛИЧЕСТВО ВЛОЖЕННЫХ ПРЯМОУГОЛЬНИКОВ?'
10 INPUT P
19 PRINT 'ОСНОВАНИЕ ВНЕШНЕГО ПРЯМОУГОЛЬНИКА (НЕЧЕТНОЕ ЧИСЛО)?'
20 INPUT A
21 PRINT 'ВЫСОТА ПОСЛЕДНЕГО ВЛОЖЕННОГО ПРМОУГОЛЬНИКА?'
22 INPUT C
50 LET P1=2*P-1
60 FOR J=1 TO P1
70 LET J1=J/2-INT(J/2)
80 IF J1=0 GOTO 250
90 IF P=1 GOTO 100
91 IF J=1 GOTO 100
92 LET L=2*J-1
94 GOSUB 600
100 LET L=2*J-1
110 LET K=A-2*(J-1)
120 GOSUB 550
130 IF J>1 GOTO 150
140 PRINT
150 IF P=1 GOTO 300
160 IF J=1 GOTO 300
170 LET K=A-2*(J-1)
180 GOSUB 650
190  PRINT
200 GOTO 300
250 LET L=2*J
260 GOSUB 650
270 LET K=A-2*(J-2)
280 GOSUB 650
290 PRINT
300 NEXT J
310 FOR J=1 TO C
312 GOSUB 600
314 GOSUB 650
316 PRINT
318 NEXT J
320 FOR J=1 TO P1
322 LET J1=J/2-INT(J/2)
324 IF J1=0 GOTO 400
326 IF P=1 GOTO 344
328 IF J=P1 GOTO 344
330 LET L=L-4
332 GOSUB 600
334 IF J>1 GOTO 340
336 LET L=L+4
338 GOTO 350
340 LET K=K+4
342 GOTO 350
344 LET L=1
346 LET K=A
350 GOSUB 550
352 IF J=P1 GOTO 490
354 IF P=1 GOTO 490
356 GOSUB 650
360 PRINT
370 GOTO 440
400 LET L=L-4
402 GOSUB 600
406 LET K=K+4
410 GOSUB 650
430 PRINT
435 LET L=L+4
```



```
436 LET K=K-4
440 NEXT J
490 GOTO 700
550 FOR I=L TO K STEP 2
560 PRINT TAB (I)'*';
570 NEXT I
580 RETURN
600 FOR I=1 TO L STEP 4
610 PRINT TAB(I)'*';
620 NEXT I
630 RETURN
650 FOR I=K TO A STEP 4
660 PRINT TAB(I)'*';
670 NEXT I
680 RETURN
700 END
```

Упражнение 4

1. Используя программу вычисления наибольшего общего делителя двух натуральных чисел в качестве подпрограммы, написать программу вычисления наибольшего общего делителя трех натуральных чисел.

2. Написать программу, формирующую массив *B*, элементы которого — суммы цифр соответствующих элементов заданного массива *A*. Вычисление суммы цифр числа предусмотреть в подпрограмме.

3. Задан прямоугольный двумерный массив. Напишите программу:

а) изменения всех элементов массива на противоположные по знаку;

б) вычисления наибольшего из чисел, встречающихся в массиве.

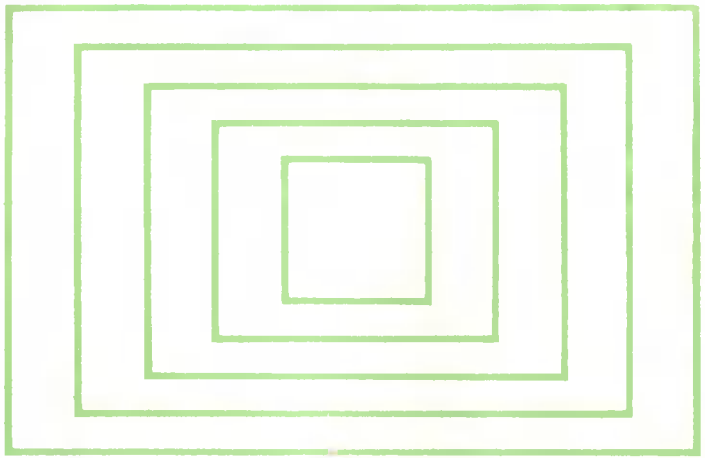


Рис. 68

4. Задан квадратный двумерный массив. Написать программу, определяющую массив C, i-м элементом которого является:

а) наименьший элемент i-й строки;
б) наибольший элемент i-го столбца;
в) сумма элементов i-й строки;
г) среднее арифметическое всех элементов i-го столбца.

18. Подготовка микро-ЭВМ «Электроника ДЗ-28» к работе

1. Подсоединить к сети вычислительное устройство и дисплей.

2. На задней панели вычислительного устройства сетевой тумблер поставить в положение «ВКЛ».

При этом:

на индикаторе (он состоит из двух регистров: X — нижний регистр, Y — верхний регистр) во всех разрядах появятся нули, загорится и погаснет нижняя лампочка индикатора (ПУ — периферийное устройство), так как периферийное устройство еще не включено;

загорятся индикаторные лампочки на панели пульта управления над клавишами

0	0	0	0
80	40	20	10

загорится лампочка слева от клавиши

0 [Р] — работа

3. На задней панели дисплея сетевой тумблер поставить в положение «ВКЛ». При этом раздается звуковой сигнал, и после прогрева на экране дисплея появляется таблица содержимого памяти. Загорается лампочка над клавишей [АВТ] — автономный режим. Звуковой сигнал и таблица исчезают после нажатия клавиш

[БЛОК] [¤]
[\] [БЛОК]

При включении дисплея на индикаторе пульта управления загорается лампочка ПУ и начинают мигать цифры. Для устранения мигания достаточно нажать на пульте управления клавишу [С] (сброс). В этот момент эти два устройства находятся в автономном режиме, позволяющем использовать вычислительное устройство как большой микрокалькулятор, а дисплей для отображения на экране символов, имеющихся на клавиатуре. В этом режиме удобно ознакомиться с расположением символов на клавиатуре, с клавишами управления курсуром, можно рисовать различные рисунки. Но с программами работать еще нельзя, так

как не организована загрузка в систему микро-ЭВМ интерпретатора языка БЕЙСИК.

4. Загрузить интерпретатор языка БЕЙСИК в память системы.

Интерпретатор БЕЙСИКа находится на кассетной МЛ. Каждый вариант БЕЙСИКа имеет свою контрольную сумму. Пусть, например, интерпретатор БЕЙСИК имеет контрольную сумму 157 107.

Для загрузки необходимо:

вставить кассету с записью интерпретатора БЕЙСИКа в НМЛ (накопитель на магнитной ленте);

перемотать МЛ в начало, для этого погасить индикаторные лампочки над клавишами прямого кодирования [20] [10] и нажать клавишу [00]; считать программу с МЛ в ОЗУ (оперативное запоминающее устройство), нажав клавиши: [С] и [СЛ] (считывание с МЛ). На индикаторе должны погаснуть нули. При благополучном считывании одного блока интерпретатора с МЛ (на ленте, как правило, записано несколько дублирующих блоков) загорается лампочка ПУ и в регистрах X и Y появляются нули. Если при считывании информации с МЛ загорается средняя лампочка индикатора ОП (ошибка программы) или верхняя лампочка ОМ (ошибка считывания с МЛ) и по окончании считывания с МЛ на индикаторе появляются мигающие нули, то в этом случае считанный блок может оказаться плохим. Нужно считать следующий блок интерпретатора, нажав клавиши [С] [СЛ];

проверить контрольную сумму считанной с МЛ программы, нажав клавишу [КП] (контроль программы). Если контрольная сумма, появившаяся на регистре X, совпадает с эталонной, перейти к п. 5, иначе повторить считывание программы с МЛ в ОЗУ.

5. На клавиатуре дисплея нажать клавиши

[ДИАЛОГ] [ЛИН] [РОЛ]

6. На пульте устройства управления нажать клавиши [С] и [S] (связь).

На экране дисплея появится сообщение:

БЕЙСИК ДЗ-28, ВАРИАНТ 3А
Снимите КАССЕТУ!

Действия оператора микро-ЭВМ:

снять кассету с НМЛ;

нажать клавишу [ПС].

На экране дисплея появится сообщение:

В КОМПЛЕКС ВХОДИТ ПЕЧАТАЮЩЕЕ УСТРОЙСТВО [0 — НЕТ, 1 — ДА?]

Действия оператора микро-ЭВМ:

если печатающее устройство не входит в состав комплекса микро-ЭВМ или Вы им сейчас не пользуетесь, то надо просто нажать клавишу [ПС];

если печатающее устройство понадобится для работы, нажать клавиши [1] [ПС]. На экране дисплея появится текст:

ВВЕДИТЕ ТИП ПЕЧАТАЮЩЕГО УСТРОЙСТВА
(0 — ТПУ, 1 — УВВПЧ-30-0004, 2 — CONSUL,
3 — ROBOTRON)?

В ответ необходимо нажать клавишу соответствующей цифры на клавиатуре дисплея и [ПС]. После этого на экране дисплея появится сообщение:

НОМЕРА ВНЕШНИХ ПОДПРОГРАММ?

Действия оператора микро-ЭВМ:

если используются подпрограммы, записанные на МЛ, то указать номера этих подпрограмм и нажать клавишу [ПС];

если подпрограммы с МЛ не используются, то нажать клавишу [ПС]. На экране дисплея появится сообщение:

ГОТОВ
:

Система находится в диалоговом режиме и готова к приему команд с клавиатуры дисплея.

19. Работа в программном режиме

Начиная с этого момента, работа у экрана дисплея может проводится в различной последовательности, в зависимости от того, что нужно сделать.

Ситуация первая. Вы имеете текст программы, написанный на бумаге. Нужно произвести просчет по этой программе и получить какие-то результаты, не сохраняя текст программы на магнитной ленте, т. е. воспользоваться данной программой один раз.

Для этого необходимо выполнить следующие действия:

Набрать текст программы на экране дисплея, используя клавиатуру дисплея.

Для набора программы используются символьные клавиши. Буквы латинского алфавита набираются при включенной лампочке [0 ЛАТ], буквы русского алфавита — при отключенной лампочке [0 ЛАТ], используя клавишу переключения регистров [ЛАТ]. Цифры набираются независимо от состояния регистров. Специальные символы, указанные в нижнем обозначении клавиш клавиатуры, печатаются с помощью нажатой клавиши [СДВИГ]. Набор информации в каждой строке заканчивается нажатием клавиши [ПС], определяющей конец ввода информации в данной строке и переход к началу следующей строки. При этом на экране дисплея появляется символ «:». Курсор находится в первой позиции справа от этого символа. Таким образом набирается на экране весь текст программы. Если в процессе набора текста строки обнаружена ошибка, то, если еще не нажата клавиша [ПС], ошибочный символ можно исправить, используя клавишу забоя [←] в русском регистре. Нажимая эту клавишу, подводим курсор к ошибочному символу. При этом символы, под которыми передвигается курсор, исчезают. После этого, набрав правильную информацию до конца строки, нажимают клавишу [ПС]. Если ошибочная информация в строке была обнаружена после нажатия клавиши [ПС], то исправить ее можно только повторным вводом оператора с данным номером строки. При этом предыдущее содержание строки программы с этим номером заменяется на новое. Для вставки новой (или пропущенной) строки набрать на экране информацию строки с номером, промежуточным между номерами раздвигаемых строк, и нажать клавишу [ПС].

Для удаления нескольких строк или всей программы используется оператор CLEARP NC1, NC2, где NC1 и NC2 — номера удаляемых строк.

После исправления ошибки следует убедиться в правильности внесенных изменений. Для этого и для просмотра готового текста программы используется команда LIST без номера строки перед ней. LIST NC1 — просмотр текста строки программы с номером NC1. LIST NC1, NC2 — просмотр текста программы, начиная с номера NC1 по номер NC2.

Текст программы выводится на экран порциями, равными странице экрана. Если программа содержит больше строк, то для дальнейшего просмотра необходимо просто нажать клавишу [ПС].

Когда текст программы полностью набран и отредактирован на экране дисплея, программу можно запустить на исполнение. Для этого на клавиатуре набирается команда интерпретатора RUN без номера строки перед ней. Исполнение программы производится интерпретатором построчно. Поэтому, если в какой-то строке будет обнаружена ошибка, интерпретатор выдаст на экран сообщение

ОШИБКА ннн В СТРОКЕ НС

Здесь ннн — номер ошибки, НС — номер строки программы, в которой содержится ошибка. Исполнение программы прекращается. По номеру ошибки в таблице 3 «Сообщение интерпре-

татора БЕЙСИК об ошибках» (см. приложения) находите причину ошибки, исправляете ее и запускаете программу на исполнение.

При благополучном завершении программы на экране дисплея отображаются результаты работы программы и высвечивается сообщение

ОСТАНОВ В СТРОКЕ НС

Здесь НС — номер строки, в которой программа прекратила свое исполнение.

Отлаженный текст программы можно вывести на печатающее устройство. Для этого используется команда LIST#1. Для вывода всех результатов работы программы на печатающее устройство в начале программы используется оператор PRINT#1.

Например:

```
5 PRINT#1
:
30 PRINT! 5.0! K, L, M
:
80 PRINT A, B, C
```

В этом примере все переменные, указанные в строках 30 и 80, будут выведены на печать. Если необходимо вывести на печать часть результатов, а часть их сохранить на экране дисплея, то используется либо оператор PRINT#1, либо PRINT#0.

Например:

```
:
30 PRINT#1! 5.0! K, L, M
:
80 PRINT#0 A, B, C
:
120 PRINT#1! 3.2! F, D
```

В этом примере оператор в строке 30 выводит на печать переменные K, L, M. Оператор в строке 80 отменяет предыдущий оператор PRINT#1 и выводит переменные A, B, C на экран. Оператор в строке 120 отменяет оператор PRINT#0 строки 80 и выводит переменные F, D на печать.

Ситуация вторая. Нужно сохранить отлаженный текст программы на магнитной ленте для многократного пользования программой.

Для этого необходимо:

Поставить в НМЛ кассету, на которую будет записываться программа.

Если на этой кассете уже есть записанные программы, то надо сначала перемотать магнитную ленту в начало командой REWIND. Затем пропустить все имеющиеся на МЛ программы, набрав команду SKIPNF, где N — количество записанных программ. При этом МЛ начинает прокручиваться и останавливается

после последней записанной программы. Затем набрать команду

SAVE 'имя программы' NC1, NC2

Здесь 'имя программы' — произвольный текст, NC1, NC2 — номера строк программы.

При записи программы на МЛ интерпретатор формирует программный файл, состоящий из следующих записей:

а) заголовок файла;
б) запись текста строк программы;
в) конец файла.

В заголовок программного файла записывается имя программы.

Ситуация третья. Нужно воспользоваться программой, записанной на магнитной ленте.

Для этого необходимо загрузить в память машины программу, находящуюся на магнитной ленте, т. е.:

Поставить кассету с записанной программой в НМЛ;

Набрать на клавиатуре дисплея команду

REWIND

Загрузить программу с МЛ в память дисплея командой

SKIPNF [ПС]

LOAD «имя программы» NC1, NC2 [ПС]

На индикаторе вычислительного устройства гаснет лампочка ПУ и дисплей находится в состоянии ожидания завершения команды считывания (на экране в следующей строке мигает курсор, но нет символа «:», т. е. команды с клавиатуры дисплея сейчас не воспринимаются).

После завершения считывания магнитная лента останавливается и загорается лампочка ПУ. На экране дисплея появляется символ «:». Дисплей готов к принятию команд с клавиатуры.

Набираем оператор

PRINT OPEN

для того, чтобы прочесть имя загруженной в память программы. По этому оператору на экране дисплея появится сообщение:

P имя программы

Если имя загруженной в память программы совпадает с именем Вашей программы, то можно приступать к работе с программой, т. е. прочесть текст программы на экране, вывести его на печать, отредактировать, если необходимо, или сразу запустить программу на исполнение.

Если имя загруженной программы не совпадает с именем Вашей программы, то повторить загрузку программы с МЛ заново.

При считывании программы с магнитной ленты по команде SKIP может возникнуть сбойная ситуация: лента прокручивается до конца, не найдя нужного блока. В данном случае машина

окажется в состоянии зависания. Команды с клавиатуры дисплея не воспринимаются. Чтобы вернуться к диалоговому режиму, надо на пульте управления вычислительного устройства последовательно нажать клавиши

[С] [ШН] [▷] [М].

На экране дисплея появится символ «:». Дисплей готов к принятию команд.

20. Этапы решения задачи с использованием микро-ЭВМ

1. Ознакомиться с условием задачи. Четко уяснить смысл задачи и конкретную цель, к которой нужно прийти.
2. Определить исходные данные задачи и результаты, которые должны быть получены.
3. Составить схему алгоритма.
4. Реализовать полученную схему программой на языке БЕЙСИК.
5. Набрать текст программы на экране дисплея.

Команды программы набираются построчно на клавиатуре дисплея. В конце каждой набранной строки нажимается клавиша [ПС].

По мере заполнения последней строки, если была нажата клавиша [РОЛ], текст сдвигается на одну строку вверх, оставляя свободной последнюю строку экрана для набора команд. Это дает возможность постоянно видеть на экране предыдущую информацию.

6. Запустить программу на исполнение командой RUN, набранной без номера строки.

21. Сообщения об ошибках

При возникновении ошибки при вводе или исполнении программы пользователя интерпретатор выдает на экран сообщение об ошибке.

Для ошибок в строке, вводимой по оператору INPUT, будет указан нулевой номер строки.

При вводе программы пользователя возможно возникновение ошибок 0,2,4,5,6.

Остальные ошибочные ситуации анализируются в процессе исполнения операторов и программы.

Ошибки с номерами 0—99 вызывают останов программы и переход в непосредственный режим, т. е. в режим ожидания команд пользователя.

При ошибках с номерами 121—128 после печати сообщения об ошибке исполнение программы продолжается.

Таблица сообщений об ошибках помещена в приложениях (таблица 3).

Клавиатура алфавитно-цифрового дисплея микро-ЭВМ «Электроника Д3-28»

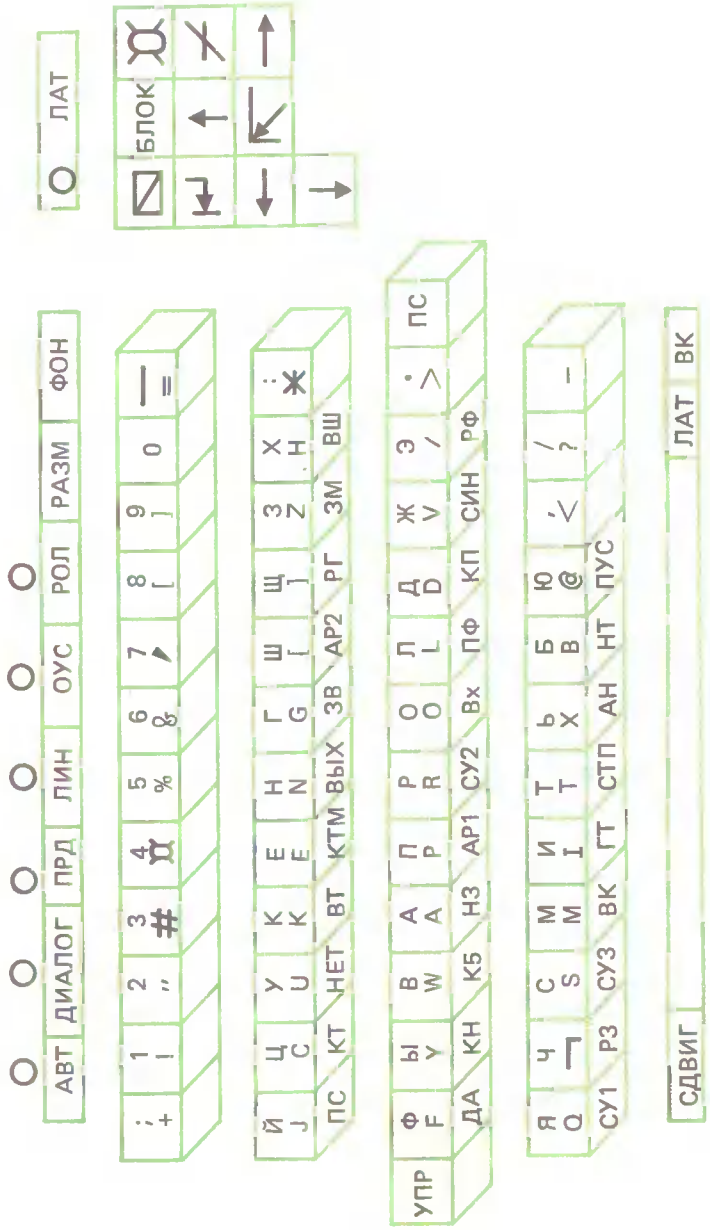


Таблица 2

Клавиатура устройства управления микро-ЭВМ «Электроника Д3-28»

В РАД	РАД В	sin x	cos x	tg x	arc sin x	arc cos x	arc tg x	дек. в пол.	пол. в дек.	sh x	ch x	th x	arc sh x	arc ch x	arc th x
80	40	20	10												
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15

Р	С	НМ	М	ОП	1/x	СК	7	8	9	◇	÷	÷П	ЗП		
В	КП	ПШ	СП	√X	X^2	ЗН	4	5	6	↓↑	x	xП	П		
ПВ	НШ	ИШ	△	10^X	lg X	E	1	2	3	→	—	П−	ВП		
П	Ш	ШН	S	e^X	ln X	π	0			←	+	П+	ЗП		

Таблица 3

Сообщения интерпретатора БЕЙСИК об ошибках

Номер ошибки	Содержание ошибки
0	Переполнение памяти, отведенной пользователю
1	Недопустимый оператор
2	Переполнение строки ввода
3	Недопустимый ограничитель в строке
4	Недопустимый номер строки
5	Несоответствие кавычек в предложении
6	Отсутствие открывающей скобки перед аргументом функции
7	Недопустимый оператор
10	Неправильная запись индексов
11	Неправильная размерность индекса. Массив не определен.
12	Несоответствие скобок в выражении
13	Недопустимый элемент выражения
14	Функция пользователя не определена
15	Неправильное имя переменной
20	Неправильная операция отношения
21	Недопустимый оператор IF
22	Неправильный DIM, COM
23	Недостаточно места для массива
24	Неправильный DEF
25	Нет данных для READ
26	Недопустимый оператор DATA
27	Неправильный формат команд в CMD
30	Неправильный формат FOR—NEXT
31	Нет NEXT
32	Не было FOR
33	Переполнение стека FOR—NEXT
34	Нулевой шаг FOR
35	Неправильный формат PRINT
36	Неправильно задан формат печати
37	Недопустимое выражение в TAB
38	Отсутствие открывающей записи в буфере МЛ
40	$HC_2 < HC_1$
41	Превышение уровня подпрограмм
42	RETURN без GOSUB
43	Нет строки для перехода по GOSUB или GOTO
44	Нет внешней подпрограммы с указанным номером
50	Неправильное предложение с операторами обслуживания МЛ и перфоленты
52	Сбой структуры файла
53	Отсутствие в ЗУ массива при приеме с МЛ и перфоленты
54	Не считан очередной блок данных с МЛ
55	Считанный с МЛ блок не помещается в ОЗУ
59	При загрузке или записи программы с МЛ указан только один номер строки
60	Нет ответа печатающего устройства
61	Нет ответа считывателя
62	Нет ответа перфоратора
63	Нет ответа накопителя при обмене данными
64	Нет ответа накопителя при приеме регистра состояния
65	Ошибка по контрольному коду
66	Ошибка нечетности
67	Ошибка при поиске
68	Нет ответа на выбор накопителя

Продолжение

Номер ошибки	Содержание ошибки
69	Нет ответа на выбор команды
70	Недопустимый номер накопителя
71	Неправильный формат SET
72	Недопустимый номер диска
73	Недопустимый номер сектора или дорожки
121	Недопустимые знаки при вводе по INPUT
122	Недостаточно данных для INPUT
123	Несуществующая переменная
124	Слишком много данных для INPUT
128	Некорректная операция (ОП) в процессе вычислений

Литература

Антипов И. А. Программирование на БЕЙСИКе // Математика в школе.— 1985.— № 4.— С. 48—52.

Блох А. Ш., Павловский А. И., Пенкрат В. В. Программирование на микрокалькуляторах.— Минск, 1981.— 192 с.

Блох А. Ш., Павловский А. И., Пенкрат В. В. Микрокалькуляторы в школе.— Минск, 1986.— 100 с.

Блох А. Ш., Кузнецов А. Т. Вычислительная математика и программирование.— Минск, 1983.— 208 с.

Громко Н. И. Введение в страну ЭВМ.— Минск, 1984.— 205 с.

Дьяконов В. П. Справочник по расчетам на микрокалькуляторах.— М., 1984.— 224 с.

Криницкий А. А. Алгоритмы вокруг нас.— М., 1984.— 224 с.

Программирование на языке БЕЙСИК-ПЛЮС для СМ-4.— М.: Финансы и статистика, 1982.— 32 с.

Программы факультативных занятий по информатике и вычислительной технике.— Минск, 1985.— 14 с.

Поснова М. Ф., Поснов Н. Н. Микрокалькулятор с программным управлением в учебной лаборатории.— Минск, 1985.— 207 с.

Растригин Л. А. Вычислительные машины, системы, сети...— М., 1982.— 224 с.

Трохименко Я. К., Любич Ф. Д. Микрокалькулятор, ваш ход! — М., 1985.— 223 с.

Уорт Т. Программирование на языке БЕЙСИК.— М., 1981.— 81 с.

СОДЕРЖАНИЕ

Введение
Общие сведения о вычислительных машинах . . . 3
Алгоритмы и их схемы . . . 6

Элементы программирования на микрокалькуляторах
1. Ввод и представление чисел. Выполнение арифметических операций 28
2. Левые и правые арифметические формулы . . . 32
3. Простейшие арифметические формулы . . . 34
4. Запись и вызов чисел из регистров памяти. Вычисления с использованием регистров памяти. Стековая память . . . 37
5. Общие свойства вычисления значений элементарных функций . . . 42
6. Особенности вычисления тригонометрических функций . . . 44
7. Решение задач с использованием элементарных функций . . . 46
8. Решение задач . . . 47
9. Программирование на микрокалькуляторе. Программная память. Понятие команды. Адрес и код команды. Составление линейных программ 50
10. Ввод, редактирование и отладка программы . . . 52
11. Команды перехода. Организация передачи управления . . . 53
12. Написание программ для разветвляющихся алгоритмов. Кодирование нечисловых ответов . . . 56
13. Составление программ для циклических алгоритмов . . . 64
14. Подпрограммы и обращение к ним . . . 70
15. Организация косвенных переходов. Косвенный вызов и засылка чисел в регистры памяти . . . 76
16. Команды косвенного безусловного и условного переходов . . . 78
17. Задачи на моделирование . . . 86
18. Вычисление определенных интегралов по формулам прямоугольников, трапеций . . . 90
19. Решение трансцендентных уравнений методом деления отрезка пополам . . . 94

Программирование на языке Бейсик
1. Язык программирования БЕЙСИК . . . 100
2. Алфавит . . . —
3. Величины БЕЙСИКа . . . 101
4. Встроенные функции . . . 102
5. Выражения . . . —
6. Арифметический оператор присваивания . . . —
7. Задание исходных значений . . . 103
8. Вывод информации . . . 104
9. Примеры линейных программ . . . 106
10. Условные и безусловные переходы . . . 108
11. Реализация базовых структур на языке БЕЙСИК . . . —
12. Запись структурированных программ . . . 112
13. Операторы цикла . . . 114
14. Ввод и вывод информации . . . 115
15. Подпрограммы . . . 120
16. Функции пользователя . . . 123
17. Тексты программ решения задач . . . 124
18. Подготовка микро-ЭВМ «Электроника ДЗ-28» к работе . . . 132
19. Работа в программном режиме . . . 134
20. Этапы решения задачи с использованием микро-ЭВМ . . . 138
21. Сообщения об ошибках . . . —

Приложения . . . 139
Литература . . . 142

И 74 **Информатика** и вычислительная техника. Факультатив. курсы: Элементы программирования на микрокалькуляторах; Программирование на языке БЕЙСИК: Учеб. пособие для 9—10 кл./А. И. Павловский, В. В. Пенкрат, А. Т. Кузнецов, Л. Н. Насенникова.— Мн.: Нар. асвета, 1988.— 143 с.: ил.
ISBN 5-341-00350-0.

Факультативный курс «Элементы программирования на микрокалькуляторах» (9 кл.) посвящен вопросам математического обслуживания электронной клавишной вычислительной машины с программной памятью «Электроника МК-61». «Программирование на языке БЕЙСИК» — факультативный курс (10 кл.), предназначенный для школ, имеющих возможность использовать микро-ЭВМ «Электроника ДЗ-28» с языком программирования БЕЙСИК. Помещены сведения об алгоритмах и их схемах, относящиеся к обоим факультативам.

И $\frac{4306022100-031}{М\ 303(03)-88}$ 120—87 ББК 73я72+32.97я72

Учебное издание

Павловский Альварес Иванович
Пенкрат Валентина Владимировна
Кузнецов Анатолий Тимофеевич
Насенникова Лариса Николаевна

ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Факультативные курсы:
Элементы программирования на микрокалькуляторах
Программирование на языке БЕЙСИК

Учебное пособие для 9—10 классов

Заведующий редакцией *Б. А. Кимбар*. Редактор *К. М. Лукашевич*. Художник *А. А. Богуш*. Исполнительская графика *В. П. Калинина*. Художественный редактор *Н. Л. Шавшукова*. Технические редакторы *М. Р. Калиберова, М. И. Чепловодская*. Корректор *Р. С. Ахремчик*.

ИБ № 2531

Сдано в набор 18.02.87. Подписано в печать 01.02.88. АТ 14 010. Формат 60×90 1/16. Бумага офсетная № 1. Гарнитура литературная. Офсетная печать. Усл. печ. л. 9+0,25 форз. Усл. кр.-отт. 19. Уч.-изд. л. 8,5+0,2 форз. Тираж 58 000 экз. Заказ 69. Цена 55 к.

Издательство «Народная асвета» Государственного комитета БССР по делам издательств, полиграфии и книжной торговли. 220600, Минск, проспект Машерова, 11.

Набор выполнен Минским ордена Трудового Красного Знамени полиграфкомбинатом МППО им. Я. Коласа. 220005, Минск, Красная, 23.

Отпечатано на Минской фабрике цветной печати 220115, Минск, Корженевского, 20.

Служебные слова языка БЕЙСИК

Служебное слово	Значение
COM	ОБЩИЙ
DATA	ДАННЫЕ
DEF	ОПРЕДЕЛИТЬ
DIM	РАЗМЕРНОСТЬ
END	КОНЕЦ
FOR	ДЛЯ
FN	ФУНКЦИЯ
GOSUB	ИДТИ К ПОДПРОГРАММЕ
GOTO	ИДТИ К
IF	ЕСЛИ
INPUT	ВВЕСТИ
LET	ПУСТЬ
NEXT	СЛЕДУЮЩИЙ
ON	ПО
OPEN	ОТКРЫТЬ
PRINT	НАПЕЧАТАТЬ
READ	ПРОЧИТАТЬ
REM	КОММЕНТАРИЙ
RE STORE	ВОССТАНОВИТЬ
RETURN	ВОЗВРАТ
STEP	ШАГ
STOP	ОСТАНОВ
TAB	ТАБУЛИРОВАТЬ
THEN	ТО
TO	К
CALL	ВЫЗВАТЬ
CLEAR	ОЧИСТИТЬ
LIST	ПРОСМОТРЕТЬ
LOAD	ЗАГРУЗИТЬ
REWIND	ПЕРЕМОТАТЬ МЛ В НАЧАЛО
SAVE	ХРАНИТЬ
SKIP	ПРОПУСК
RUN	ПУСК

Стандартные функции языка БЕЙСИК

Обозначение	Общепринятая запись	Примечание
DEG (x)	$\frac{180}{\pi} \cdot x$	ФУНКЦИЯ ПЕРЕВОДА РАДИАН В ГРАДУСЫ
RAD (x)	$\frac{\pi}{180} \cdot x$	ФУНКЦИЯ ПЕРЕВОДА ГРАДУСОВ В РАДИАНЫ
SIN (x)	sin x	АРГУМЕНТ В РАДИАНАХ
COS (x)	cos x	АРГУМЕНТ В РАДИАНАХ
TAN (x)	tg x	АРГУМЕНТ В РАДИАНАХ
ASN (x)	arcsin x	ЗНАЧЕНИЕ ФУНКЦИИ В РАДИАНАХ
ACS (x)	arccos x	ЗНАЧЕНИЕ ФУНКЦИИ В РАДИАНАХ
ATN (x)	arctg x	ЗНАЧЕНИЕ ФУНКЦИИ В РАДИАНАХ
HSN (x)	sh x	АРГУМЕНТ В РАДИАНАХ
HCS (x)	ch x	АРГУМЕНТ В РАДИАНАХ
HTN (x)	th x	АРГУМЕНТ В РАДИАНАХ
AHS (x)	arsh x	ФУНКЦИЯ В РАДИАНАХ
AHC (x)	arch x	ФУНКЦИЯ В РАДИАНАХ
AHT (x)	arth x	ФУНКЦИЯ В РАДИАНАХ
ABS (x)	$\lvert x \rvert$	МОДУЛЬ ЧИСЛА X
EXP (x)	e^x	ПОКАЗАТЕЛЬНАЯ ФУНКЦИЯ
EXT (x)	10^x	ПОКАЗАТЕЛЬНАЯ ФУНКЦИЯ
LOG (x)	ln x	НАТУРАЛЬНЫЙ ЛОГАРИФМ ЧИСЛА X
LGT (x)	lg x	ДЕСЯТИЧНЫЙ ЛОГАРИФМ ЧИСЛА X
SQR (x)	$\sqrt{x}$	КОРЕНЬ КВАДРАТНЫЙ ИЗ ЧИСЛА X
INT (x)	ent x	ЦЕЛАЯ ЧАСТЬ ЧИСЛА X НЕ ПРЕВОСХОДЯЩАЯ X
SGN (x)	$\mathrm{sgh}\ x = \begin{cases} 1, \text{ при } x > 0 \\ 0, \text{ при } x = 0 \\ -1, \text{ при } x < 0 \end{cases}$	СИГНАТУРА ЧИСЛА X
RND (x)	Псевдослучайное число с равномерным распределением на [0, 1]	При X = 0 устанавливается начальное значение, при X ≠ 0 следующее