В. Е. АЛЕКСЕЕВ А. С. ВАУЛИН Г. Б. ПЕТРОВА

Вычислительная техника в инженерных и экономических расчетах

сборник задач и упражнений

для вузов

издательство Высшая Школа

В. Е. АЛЕКСЕЕВ, А. С. ВАУЛИН, Г. Б. ПЕТРОВА

Вычислительная техника в инженерных и экономических расчетах

Сборник задач и упражнений

Под редакцией д-ра техн. наук, проф. А. В. Петрова

Допущено Министерством высшего и среднего специального образования СССР в качестве учебного пособия для студентов высших учебных заведений

МОСКВА «ВЫСШАЯ ШКОЛА» 1984

ББК 65.9(2)21
А47
УДК 681.332.6

Рецензенты:
кафедра «Вычислительная техника» Московского института электронного машиностроения (зав. кафедрой — *П. П. Сыпчук*), д-р техн. наук, проф. *В. И. Дракин.*

Алексеев В. Е., Ваулин А. С., Петрова Г. Б.
А47 Вычислительная техника в инженерных и экономических расчетах. Сборник задач и упражнений: Учеб. пособие для вузов./Под ред. *А. В. Петрова.* — М.: Высш. шк., 1984.— 136 с., ил.
35 к.

В пособии рассматриваются типовые задачи по программированию на алгоритмических языках ФОРТРАН-IV и ПЛ/1, приводятся задания для самостоятельной разработки схем алгоритмов и программ, выполняемых на ЭВМ третьего поколения. Для большинства задач даются ответы.

Приведенные методические указания по разработке программ тесно связаны с материалом учебника «Вычислительная техника в инженерных и экономических расчетах» (см. № 132).

А $\frac{2405000000-329}{001(01)-84}$ 129—84

ББК65.9(2)21
6Ф7

ПРЕДИСЛОВИЕ

Рост научно-технического прогресса, усиление роли науки — объективная тенденция современного этапа коммунистического строительства. Экстенсификация экономического развития сменяется интенсификацией, осуществляемой при широком использовании достижений науки. Характерная черта научно-технической революции — стремительно возрастающая роль ЭВМ во всех областях человеческой деятельности. Без ЭВМ в настоящее время немыслимо решение широкого круга инженерных, научных, экономических и управленческих задач. Использование ЭВМ в многовариантных и оптимизационных расчетах, управление технологическими процессами и организационно-экономическом управлении народным хозяйством, автоматизации проектирования сложных объектов позволяет получить значительный экономический эффект за счет сокращения сроков проектирования, повышения качества проектных работ, что, в свою очередь, приводит к снижению себестоимости и повышению производительности труда.

Следовательно, темпы научно-технического прогресса в ведущих областях народного хозяйства в огромной степени будут определяться качеством и номенклатурой вычислительных средств и их программным обеспечением. Такое широкое внедрение высокопроизводительных средств вычислительной техники в различные сферы народного хозяйства диктует необходимость подготовки специалистов, сочетающих знания своей специальности с математическими методами решения задач и навыками использования для этих целей ЭВМ.

Сборник задач имеет практическую направленность и дополняет учебник «Вычислительная техника в инженерных и экономических расчетах» под ред. А. В. Петрова. Материал сборника и учебника соответствует программе одноименного курса и методически взаимоувязан.

В сборнике задач с единых позиций рассмотрены способы алгоритмизации задач, правила применения языковых средств ФОРТРАНа и ПЛ/1 и приемы программирования, что является его отличительной особенностью. Он содержит необходимые сведения теоретического характера, иллюстрирован конкретными примерами, что вполне достаточно для разработки алгоритмов и программ реальных задач. Единая методическая основа данного пособия и доступность изложения материала позволяет студентам самостоятельно изучать вопросы алгоритмизации и программирования инженерных задач.

Пособие состоит из трех разделов. Первый раздел включает задачи, связанные с разработкой структур алгоритмов и наиболее часто используемые приемы программирования. В следующих двух разделах излагаются синтаксические конструкции языков ФОРТРАН и ПЛ/1, правила организации структур программ, различных конфигураций. Нумерация задач дана по разделам.

Задачи, отмеченные символом ▲, имеют ответ, а символом ● — даны с решением.

Авторы выражают благодарность коллективу кафедры «Вычислительная техника» Московского института электронного машиностроения и д-ру техн. наук, проф. В. И. Дракину, взявшим на себя труд по рецензированию настоящего пособия и сделавшим ряд ценных замечаний.

Авторы ставили своей целью помочь читателям в практической работе при изучении вопросов алгоритмизации и программирования на алгоритмических языках ФОРТРАН-IV и ПЛ/1. Насколько это удалось, авторы надеются судить по отзывам и замечаниям, которые можно направлять по адресу: *101430, Москва, ГСП-4, Неглинная ул., д. 29/14, изд-во «Высшая школа», за что авторы заранее благодарны.*

Авторы

ВВЕДЕНИЕ

Процедура подготовки задач к решению на цифровых вычислительных машинах до настоящего времени остается достаточно сложной и трудоемкой и включает в себя следующие основные этапы:

1. Постановка задачи (задача, которую предстоит решать на ЭВМ, формулируется пользователем или получается им в виде задания).
2. Разработка или выбор существующего алгоритма решения задачи.
3. Выбор численных методов решения для отдельных участков алгоритма с учетом заданной точности, быстродействия и т. д.
4. Составление схемы программы, представляющей собой графическое изображение программы в виде отдельных блоков и их связей (схема в значительной степени облегчает знакомство с программой и упрощает ее запись на алгоритмическом языке).
5. Написание программы (по схеме программы, используя алгоритмический язык, составляют программу и записывают ее на бланках).
6. Подготовка исходных данных (в большинстве случаев для программы требуются исходные данные, которые также записываются на бланках).
7. Перенос программы и исходных данных на носитель информации (перфокарты, перфоленты).
8. Формирование задания, которое описывается на языке управления заданиями с помощью управляющих операторов.
9. Отладка программы (отладка программы может подразделяться на ряд подэтапов, включая визуальный контроль, синтаксический контроль и т. д.).

Особое место при подготовке задач к решению на ЭВМ занимают разработка или выбор алгоритма, написание программы на алгоритмическом языке и ее отладка. Описание алгоритма представляет собой общую схему решения задач.

Различают способы описания алгоритмов словесный, операторный и схемный. Наибольшее распространение в настоящее время получил схемный способ, при котором вычислительный процесс расчленяется на отдельные операции, отображающиеся в виде условных графических блочных символов. С 1.07.81 введены ГОСТ 19002—80 и 19003—80, определяющие правила выполнения схем и перечень блоков, их наименования, форму и размеры.

Эффективное использование ЭВМ при решении научно-технических задач требует умения программировать на алгоритмических языках. Наиболее распространены в настоящее время проблемно-ориентированные алгоритмические языки ФОРТРАН, ПЛ/1 и КОБОЛ. Алгоритмический язык ФОРТРАН характеризуется простотой использования, близостью записи конструкций к обычной математической записи, большими возможностями ввода — вывода информации, получением программ после трансляции, близких к оптимальным, простотой и наглядностью определения ошибок по сообщениям транслятора, наличием большой библиотеки научных подпрограмм, включающей практически все численные методы решения. Алгоритмический язык ПЛ/1, обладая всеми преимуществами языков ФОРТРАН, АЛГОЛ-60, КОБОЛ, имеет много других, присущих именно ему достоинств. Это позволяет расширить область его применения (например, использовать для решения задач экономических, планово-статистических, моделирования, логических, в реальном масштабе времени и для разработки программного обеспечения ЭВМ). Язык ПЛ/1 допускает блочную структуру программ, хорошо приспособлен для обработки символьных текстов и организации информации в списки.

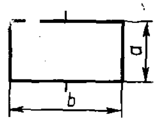
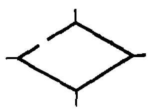
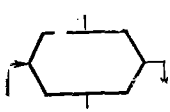
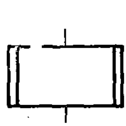
1. РАЗРАБОТКА АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ

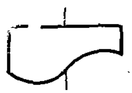
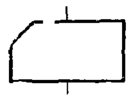
Подготовка задачи для решения на ЭВМ состоит из следующих этапов: математической формулировки условия задачи, выбора численного метода ее решения, разработки схемы алгоритма, составления программы на алгоритмическом языке и ее отладки. В настоящем сборнике условия большинства задач представлены в математической формулировке с указанием численного метода решения. Поэтому отпадает необходимость в выполнении первых двух этапов и можно непосредственно приступать к разработке схем алгоритмов решения задачи на ЭВМ.

Алгоритм — некоторая конечная последовательность предписаний (правил), однозначно определяющих процесс преобразования исходных и промежуточных данных в результат решения задачи. Таким образом, при разработке алгоритма решения задачи математическая формулировка является основой для определения последовательности действий, приводящих к получению искомого результата. Алгоритм должен отвечать требованиям детерминированности результатов, массовости применения для исходных данных, результативности, обеспечивающей получение результата через конечное число шагов.

Наиболее наглядный способ представления алгоритмов — их изображение в виде схем — последовательности блоков, предписывающих выполнение определенных функций, и связей между ними. Внутри блоков указывается поясняющая информация, характеризующая выполняемые ими действия. Блоки схемы обычно имеют сквозную нумерацию. Конфигурацию и размер блоков, а также порядок построения схем определяют ГОСТ 19002—80 и 19003—80. В табл. 1.1 приведены некоторые наиболее часто употребляемые блоки и даны пояснения к ним.

Таблица 1.1

Название символа	Символ	Примечание
Процесс		Вычислительное действие или последовательность вычислительных действий
Решение		Проверка условий
Модификация		Начало цикла
Предопределенный процесс		Вычисление по подпрограмме или стандартной подпрограмме

Название символа	Символ	Примечание
Документ		Вывод данных, печать результатов
Перфокарта		Ввод данных с перфокарт или вывод данных на перфокарты
Соединитель		Разрыв линий потока
Пуск, останов		Начало, конец, останов, вход и выход в подпрограммах
Комментарий		Пояснения, содержание подпрограмм, формулы

По характеру связей между блоками различают алгоритмы линейной, разветвляющейся и циклической структуры.

1.1. АЛГОРИТМЫ ЛИНЕЙНОЙ СТРУКТУРЫ

Алгоритм линейной структуры (линейный алгоритм) — алгоритм, в котором блоки выполняются последовательно друг за другом. Такой порядок выполнения блоков называется *естественным*.

● **1.1.** Вычислить высоты треугольника со сторонами a, b, c по формулам

$$h_a=(2/a)\sqrt{p(p-a)(p-b)(p-c)};$$

$$h_b=(2/b)\sqrt{p(p-a)(p-b)(p-c)};\quad h_c=(2/c)\sqrt{p(p-a)(p-b)(p-c)},$$

где $p=(a+b+c)/2$.

Для решения любой нетривиальной задачи может иметь место несколько алгоритмов, приводящих к получению результата ее решения. Из всех возможных алгоритмов следует выбирать наилучший в смысле некоторого критерия. Часто в качестве критерия используют либо оценку точности решения задачи, либо затраты времени на ее решение, либо некоторый интегральный критерий, включающий оценки и точности, и затрат времени.

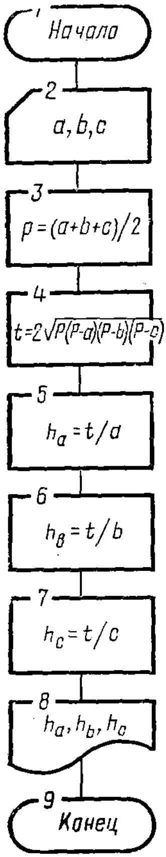


Рис. 1.1

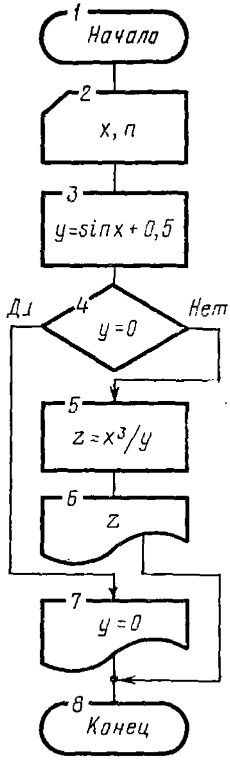


Рис. 1.2

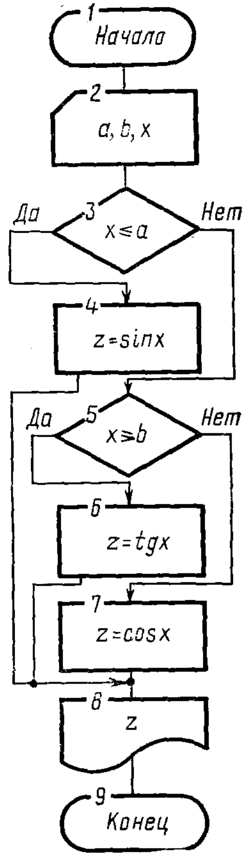


Рис. 1.3

При организации решения задачи для исключения повторений вычислять высоты следует не по приведенным выше формулам непосредственно, а используя промежуточную переменную $t=2\sqrt{p(p-a)(p-b)(p-c)}$, тогда $h_a=t/a$, $h_b=t/b$, $h_c=t/c$.

С учетом сказанного схема алгоритма решения задачи будет иметь вид, представленный на рис. 1.1.

1.2. Вычислить площадь поверхности $s=\pi(R+r)l+\pi R^2+\pi r^2$ и объем $v=(1/3)\pi(R^2+r^2+Rr)h$ усеченного конуса.

1.3. Вычислить координаты центра тяжести трех материальных точек с массами m_1, m_2, m_3 и координатами (x_1, y_1), (x_2, y_2), (x_3, y_3)

по формулам $x_c=(m_1x_1+m_2x_2+m_3x_3)/(m_1+m_2+m_3)$, $y_c=(m_1y_1+$ $+m_2y_2+m_3y_3)/(m_1+m_2+m_3)$.

1.4. Вычислить координаты точки, делящей отрезок a_1a_2 в отношении $n_1:n_2$, по формулам $x=(x_1+\gamma x_2)/(1+\gamma)$, $y=(y_1+\gamma y_2)/(1+$ $+\gamma)$, где $\gamma=n_1/n_2$.

1.5. Вычислить медианы треугольника со сторонами a, b, c по формулам

$m_a=0{,}5\sqrt{2b^2+2c^2-a^2}$, $m_b=0{,}5\sqrt{2a^2+2c^2-b^2}$,

$m_c=0{,}5\sqrt{2a^2+2b^2-c^2}$.

1.6. Вычислить значение функции $y=ae^{-ax}\sin\omega x$ при $x=$ $=(\pi/2-\varphi)/\omega$.

1.7. Вычислить значения функций

$y=(e^{-x_1}+e^{-x_2})/2$ и $z=(a\sqrt{x_1}-b\sqrt{x_2})/c$,

где $x_1=(b+\sqrt{|b^2-4ac|})/(2a)$, $x_2=(b-\sqrt{|b_2-4ac|})/(2a)$.

1.8. Определить высоту треугольника, если его площадь равна s, а основание больше высоты на величину a.

1.2. АЛГОРИТМЫ РАЗВЕТВЛЯЮЩЕЙСЯ СТРУКТУРЫ

На практике редко удается представить схему алгоритма решения задачи в виде линейной структуры. Часто в зависимости от каких-либо значений промежуточных результатов необходимо организовать вычисление либо по одним, либо по другим формулам, т. е. в зависимости от выполнения некоторого логического условия вычислительный процесс должен идти по одной или другой ветви. Алгоритм такого вычислительного процесса называется *алгоритмом разветвляющейся структуры*. В общем случае количество ветвей в таком алгоритме разветвляющейся структуры не обязательно равно двум.

● **1.9.** Вычислить значение функции $z=x^3/y$, где $y=\sin nx+0{,}5$.

Казалось бы, решение этой задачи можно описать алгоритмом линейной структуры. Однако для удовлетворения свойств массовости и результативности алгоритма необходимо, чтобы при любых исходных данных был получен результат или сообщение о том, что задача не может быть решена при заданных исходных данных. Действительно, если $y=0$, задача не может быть решена, так как деление на 0 невозможно. Поэтому в алгоритме необходимо предусмотреть этот случай и выдать в качестве результата информацию о том, что $y=0$. Таким образом, рассматриваемый вычислительный процесс должен иметь две ветви: в одной, если $y\neq0$, необходимо вычислить и отпечатать значение переменной z, а в другой — вывести на печать информацию, что $y=0$.

Этот вычислительный процесс можно описать условным выражением:

$$\begin{cases} \text{вычислить } z=x^3/y, \text{ если } y\neq0; \\ \text{вывести Y} = 0, \text{ если } y=0. \end{cases}$$

Схема алгоритма решения этой задачи представлена на рис. 1.2, где после условного блока *4* располагаются блоки сначала одной ветви (блоки 5, 6), а затем второй ветви (блок 7). Поскольку после выполнения блоков первой ветви нет надобности выполнять блоки второй ветви, осуществляется переход сразу к концу алгоритма (к блоку 8).

В алгоритме дважды нарушается естественный порядок выполнения блоков: 1) при проверке условия $y=0$ (условный переход); 2) после выполнения блоков первой ветви (безусловный переход).

● **1.10.** Вычислить значение функции

$$z=\begin{cases}\sin x, \text{ если } x\leqslant a;\\ \cos x, \text{ если } a<x<b;\\ \operatorname{tg} x, \text{ если } x>b.\end{cases}$$

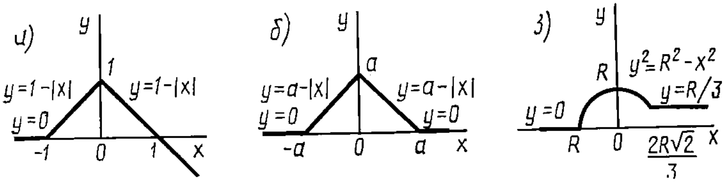


Рис. 1.4

Здесь вычислительный процесс имеет три ветви. С помощью условного блока можно проверить выполнение только условия, по которому будет определен выбор выражения для реализации одной ветви. Поэтому, чтобы установить, по какой из двух оставшихся ветвей должен идти вычислительный процесс в случае невыполнения первого условия, необходимо использовать еще один условный блок. Схема этого алгоритма представлена на рис. 1.3. Блок 3 проверяет условие $x\leqslant a$ и в случае его выполнения осуществляет переход к блоку 4, вычисляющему $z=\sin x$. Если $x>a$, то блок 5 проверяет условие $x\geqslant b$. Если это условие выполняется, то осуществляется переход к блоку 6, вычисляющему $z=\operatorname{tg} x$. В противном случае x лежит в интервале между a и b и происходит переход к блоку 7, вычисляющему $z=\cos x$. После вычислений по любой из формул осуществляется переход в общую ветвь к блоку печати.

1.11. Вычислить корни квадратного уравнения $ax^2+bx+c=0$. Если $d=b^2-4ac\geqslant0$, то корни действительные; следовательно, необходимо вычислить $x_{1,2}=e\pm f$. Если $d<0$, то корни мнимые; следовательно, необходимо вычислить e и f по формулам $e=-b/(2a)$,

$$f=\sqrt{|b^2-4ac|}/(2a).$$

1.12. Вычислить значение функции

$$q=\begin{cases}1{,}7\ e^{-x}, \text{ если } c^2-x>0;\\ 0{,}9\ e^{x}, \text{ если } c^2-x<0.\end{cases}$$

1.13. Вычислить значение функции, заданной графически (рис. 1.4, *а—в*) по заданному значению аргумента x.

1.14. Вычислить значение функции

$$c=\begin{cases} \cdot\mathrm{TRUE}\cdot, \text{ если } x>0; \\ \cdot\mathrm{FALSE}\cdot, \text{ если } x\leqslant 0. \end{cases}$$

1.15. Найти квадрат наибольшего из двух чисел a и b и вывести на печать признак $N=1$, если наибольшим является a, и $N=2$ — в противном случае.

1.16. Определить, попадает ли точка с координатами x_0, y_0 в круг радиусом r. Уравнение окружности $r^2=x^2+y^2$. Присвоить признаку $N=1$, если точка находится внутри круга, и $N=0$ — если вне круга.

1.17. Вычислить значение функции

$$z=\begin{cases} \ln x, \text{ если } x\geqslant 1; \\ 1, \text{ если } -1<x<1; \\ e^x, \text{ если } x\leqslant -1. \end{cases}$$

1.18. Определить, в каком квадранте находится точка с координатами x, y, и вывести на печать номер квадранта.

1.19. Округлить действительное положительное число x, меньшее 5, до ближайшего целого числа:

$$NX=\begin{cases} 0, \text{ если } x<0{,}5; \\ 1, \text{ если } 0{,}5\leqslant x<1{,}5; \\ 2, \text{ если } 1{,}5\leqslant x<2{,}5; \\ 3, \text{ если } 2{,}5\leqslant x<3{,}5; \\ 4, \text{ если } 3{,}5\leqslant x<4{,}5; \\ 5, \text{ если } x\geqslant 4{,}5. \end{cases}$$

1.20. Определить, является ли значение целочисленной переменной x кратным трем. Если это имеет место, то вывести значение x на печать, в противном случае вывести на печать НЕТ.

1.3. АЛГОРИТМЫ ЦИКЛИЧЕСКОЙ СТРУКТУРЫ

Часто при решении задач приходится многократно вычислять по одним и тем же математическим зависимостям при различных значениях входящих в них величин. Такие многократно повторяемые участки вычислительного процесса называются *циклами*. Использование циклов позволяет существенно сократить схему алгоритма и длину соответствующей ему программы. Различают циклы с заданным и неизвестным числом повторений. К последним относятся итерационные циклы, характеризующиеся последовательным приближением к искомому значению с заданной точностью.

● **1.21.** Вычислить и вывести на печать значения функции $y=a^3/(a^2+x^2)$ при x, изменяющемся от 0 до 3 с шагом 0, 1.

Это цикл с заданным количеством повторений, которое определяется как $n=](x_m-x_0)/h[+1$, где x_0 и x_m — соответственно начальное и конечное значение аргумента; h — шаг изменения аргумента.

Перед первым выполнением цикла необходимо задать начальное значение аргумента x, равное 0, а затем организовать 31 раз вычисление и печать значений функции y. При каждом новом выполнении цикла необходимо изменять аргумент на шаг, равный 0,1. Чтобы процесс был конечным, необходимо задать условие окончания цикла. Таким образом, для организации цикла необходимо: задавать перед циклом начальное значение переменной, изменяющейся в цикле; изменять значение переменной перед каждым новым повторением цикла; проверять условие окончания цикла; управлять циклом, т. е. переходить к его началу, если он не закончен, или выходить из него по окончании. Последние три функции выполняются многократно.

Переменную, изменяющуюся в цикле, называют *параметром цикла*. В одном цикле может быть несколько параметров.

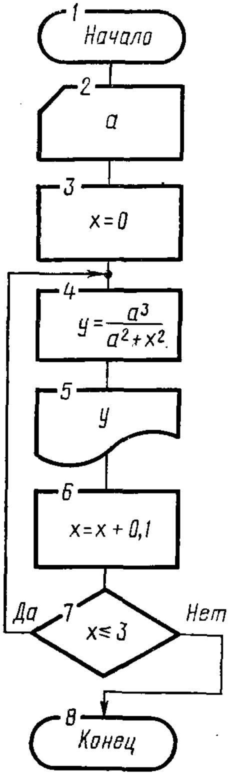


Рис. 1.5

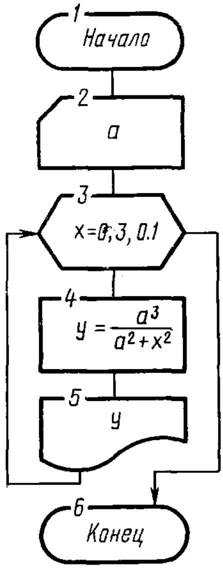


Рис. 1.6

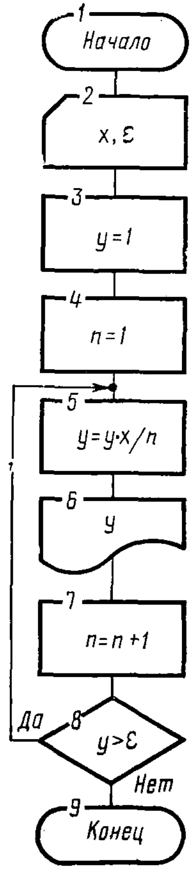


Рис. 1.7

Схема алгоритма приведена на рис. 1.5. На схеме блок *3* выполняет первую функцию, необходимую для организации цикла, блок *6* — вторую, блок *7* — третью и четвертую функции. Схема алгоритма получается во многих случаях более компактной и наглядной, если для ее построения использовать блок начала цикла, который выполняет все функции, необходимые для его организации (рис. 1.6).

1.22. Вычислить значения членов бесконечного ряда x, $\frac{x^2}{2!}$, $\frac{x^3}{3!}, \ldots, \frac{x^n}{n!}, \ldots$ до члена $x^n/n! \leqslant \varepsilon$.

Здесь имеет место итерационный цикл, так как заранее не известно, при каком n выполняется условие $x^n/n! \leqslant \varepsilon$. Для итерационных циклов число повторений зависит от некоторого промежуточного или окончательного результата, а не от параметра цикла.

Сравнивая два соседних члена ряда, видим, что $y_n/y_{n-1} = x/n$. Поэтому для уменьшения времени счета при вычислении текущего члена ряда целесообразно использовать в цикле рекуррентную формулу $y_n = y_{n-1}x/n$.

Чтобы использовать эту формулу для вычисления значения первого члена ряда $y_1 = y_0\ (x/1)$, необходимо, чтобы заданное начальное значение y_0 было равно 1. Параметром, изменяющимся в этом цикле, будет номер n члена ряда. Тогда формула для вычисления значения текущего члена ряда будет иметь вид $y = yx/n$.

Схема алгоритма такого вычислительного процесса приведена на рис. 1.7. Если использовать для организации цикла блок начала цикла, то в нем надо указать в качестве последнего значения параметра цикла некоторое большое число, заведомо большее того n, при котором выполняется условие $y \leqslant \varepsilon$.

1.23. Вычислить значения функции $z = \sqrt{(x_i + a_i)/2}$, если x_i и a_i — элементы массивов, состоящих из 40 элементов каждый.

1.24. Вычислить и вывести на печать положительные значения функции $y = \sin nx - \cos n/x$ при $n = 1, 2, \ldots, 50$.

1.25. Вычислить значения функции $a_i = \{ a_i$, если $a_i > 0$; если

1.34. Вывести на печать номера точек, лежащих в круге радиусом r. Координаты точек заданы массивами $(x_1, x_2, ..., x_{100})$, $(y_1, y_2, ..., y_{100})$.

1.35. Определить с точностью до 0,2 точку пересечения функции $y = x - e^{-(ax)^2/2}$ с осью x при изменении аргумента x от b_0 до b_m с шагом 0,2. Сначала следует определить знак функции y при $x = b_0$. Изменение знака функции является признаком пересечения оси x.

1.36. Вычислить члены ряда $\frac{x^2}{2!}, -\frac{x^3}{3!}, \ldots, (-1)^n \frac{x^n}{n!}, \ldots,$ модуль которых больше a.

1.37. В окружность радиусом r вписан многоугольник со стороной a_n. Сторона многоугольника с удвоенным числом сторон определяется по формуле $a_{2n} = \sqrt{2R^2 - 2R\sqrt{R^2 - a_n^2/4}}$. Определить a_{128}, если известны r и a_4.

1.38. Вывести на печать положительные элементы главной диагонали матрицы $\mathbf{X}$ $(n \times n)$.

1.39. Рассчитать траекторию движения снаряда по формулам $x = v_x t$; $y = v_y t - gt^2/2$ при постоянных скоростях v_x, v_y. Время t изменяется от нуля с шагом Δt.

1.40. Вычислить $c = \cos nx = \cos((n-1)x)\cos x - \sin((n-1)x) \times \sin x$, если $\cos x = 0{,}112$, $0 < x \leqslant \pi/2$, $i = 2, 3, ..., n$.

1.41. Определить количество цифр в целом числе n, если после деления числа n на $10k$ раз в целой части числа будет нуль, где k — количество цифр в числе n.

1.42. Вывести на печать элементы массива $(x_1, x_2, ..., x_n)$, кратные трем $(n \leqslant 50)$.

1.4. ХАРАКТЕРНЫЕ ПРИЕМЫ АЛГОРИТМИЗАЦИИ ЗАДАЧ

Рассмотрим приемы, наиболее часто используемые при решении практических задач.

Вычисление в цикле с несколькими одновременно изменяющимися параметрами

В рассмотренных ранее примерах алгоритмов циклической структуры в цикле изменялся только один параметр. На практике часто встречаются задачи, в которых необходимо использовать несколько параметров цикла, изменяющихся одновременно. Цикл с несколькими одновременно изменяющимися параметрами организуется по схеме, аналогичной схеме организации цикла с одним параметром. Для остальных параметров перед циклом необходимо задавать их начальные значения, а внутри его вычислять текущие.

● **1.43.** Вычислить значения функции $z = xy_i/(x + y_i)$, если x изменяется одновременно с y_i от начального значения a с шагом h, y_i являются элементами массива $(y_1, y_2, ..., y_{20})$.

Здесь в цикле, выполняемом 20 раз, изменяются два параметра: простая переменная x и i — индекс переменной y.

Схема алгоритма решения этой задачи представлена на рис. 1.8, где блок *4* задает закон изменения параметра i от 1 до 20 с шагом 1, блок *3* — перед циклом начальное значение параметра x, а блок *7* вычисляет новое значение параметра x.

1.44. Вычислить значения функции $z=\sqrt{(x_i+a)/2}$, если x_i являются элементами массива $(x_1, x_2, ..., x_{30})$, a изменяется от 2 с шагом 0,5. Считать $x_i+a>0$.

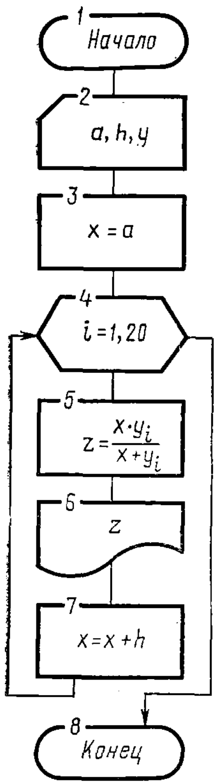


Рис. 1.8

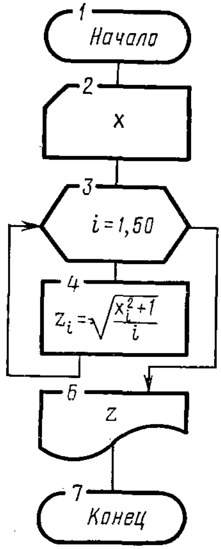


Рис. 1.9

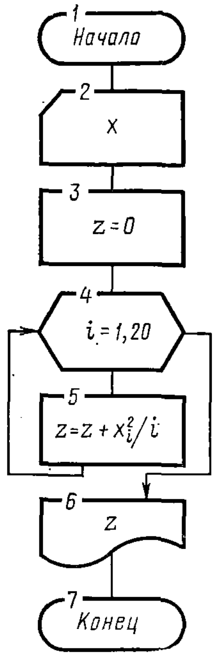


Рис. 1.10

1.45. Вычислить значения функции $z=(a+b+c_i)/i$, если a изменяется от 0 до 1 с шагом 0,1, b изменяется от 1 до 3 с шагом 0,2, c_i являются элементами массива $(c_1, c_2, ..., c_{11})$.

1.46. Вычислить значения функции $u=(x_i+y)/z_{2i-1}$, если x_i являются элементами массива $(x_1, x_2, ..., x_{50})$, z_i — массива $(z_1, z_2, ..., z_{100})$, а y изменяется от 1 с шагом 0,25.

1.47. Вычислить значения функции $z=x\sqrt{xy}$, где x изменяется от 1 с шагом 0,1 до 2, y изменяется от y_0 с шагом h. Считать $y>0$.

1.48. Вычислить и вывести на печать значения членов ряда

$$\frac{x+h}{3},\ \frac{x+2h}{5},\ \frac{x+3h}{7},\ldots,\frac{x+nh}{2n+1},\ldots,\frac{x+20h}{41}.$$

1.49. Вычислить и вывести на печать значения членов ряда

$$\frac{x_0+\Delta x}{n_0+3\Delta n},\ \frac{x_0+3\Delta x}{n_0+5\Delta n},\ldots,\frac{x_0+(2n-1)\Delta x}{n_0+(2n+1)\Delta n},\ldots.$$

Последнее значение знаменателя принять равным m.

Запоминание результатов

В рассмотренных выше задачах результатом вычислений было значение простой переменной, для записи которого в памяти ЭВМ выделяется одна ячейка. Если в процессе вычислений значение переменной изменяется, то в памяти ЭВМ после окончания вычислений остается лишь последнее значение результата. Чтобы записать все вычисленные значения, нужно выделить для их хранения необходимое количество ячеек памяти (массив), а текущий результат обозначить переменной с индексом.

● **1.50. Вычислить и запомнить значения функции** $z_i=\sqrt{(x_i^2+1)/i}$, где x_i — элементы массива $(x_1, x_2, ..., x_{50})$.

Схема алгоритма, реализующего этот вычислительный процесс, представлена на рис. 1.9. В схеме блок печати стоит за циклом, так как на печать выводится весь массив Z.

1.51. Вычислить и запомнить значения функции $z=a\,e^{bx-cx^2}$ при изменении аргумента x от 0 до 2 с шагом 0,1.

1.52. Вычислить и запомнить значения функции $z_i=(x_i+y_i)/2$, где X и Y — массивы из 45 элементов каждый.

1.53. Вычислить и запомнить значения функции

$$y_i=\begin{cases} x_i, \text{ если } x_i>0;\\ 0, \text{ если } x_i=0;\\ -1, \text{ если } x_i<0,\end{cases}$$

где x_i — элементы массива из 20 элементов.

1.54. Переписать в массив Y элементы массива $(x_1, x_2, ..., x_{40})$ в обратном порядке.

1.55. Переписать положительные элементы массива $(x_1, x_2, ..., x_{100})$ подряд в массив Y.

1.56. Записать в массив N подряд номера положительных элементов массива $(a_1, a_2, ..., a_{80})$.

1.57. Записать в массив Y подряд десять первых положительных элементов массива $(x_1, x_2, ..., x_{100})$.

1.58. Вычислить значения функции $y=n\sin x-\cos nx$, если x изменяется от x_0 до x_m с шагом h. Записать в массив Z подряд значения функции, удовлетворяющие условию $0\leqslant y\leqslant 1$.

1.59. Записать подряд в массив B элементы массива $(a_1, a_2, ..., a_{75})$, имеющие четные индексы.

1.60. Записать подряд в массив B элементы массива $(a_1, a_2, ..., a_{100})$, стоящие на четных местах, а элементы, стоящие на нечетных местах, — в массив C.