Министерство путей сообщения
Российской Федерации

Московский государственный университет
путей сообщения (МИИТ)

Кафедра математического обеспечения
автоматизированных систем управления

Д.И.ЛАВРУХИН, В.П.СОЛОВЬЁВ

Утверждено
редакционно-издательским
советом университета

ПРОЕКТИРОВАНИЕ
ТИПОВЫХ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ ОБРАБОТКИ
ЧИСЛОВОЙ ИНФОРМАЦИИ НА ЭВМ

Издание второе

Методические указания
к лабораторным и практическим работам
по дисциплине "Программирование"
для студентов факультета "Техническая кибернетика"

Москва - 1995

В методические указания включены наборы типовых задач обработки числовой информации на ЭВМ. Большое число задач обеспечивает возможность выбора индивидуальных заданий с учетом уровня подготовки студентов.

При выполнении индивидуального задания студенту необходимо:

выбрать и описать математический метод решения задачи;

спроектировать схему алгоритма решения задачи;

определить способ организации данных;

записать текст программы на алгоритмическом языке который указан преподавателем;

провести тестирование и отладку программы на ЭВМ;

проанализировать результаты и оформить отчет в соответствии с требованиями стандартов.

ВВЕДЕНИЕ

Понятие алгоритма является одним из исходных понятий в теории алгоритмов и, как любое другое исходное понятие математической теории, строго не определяется. Интуитивно под алгоритмом мы понимаем некоторое формальное предписание, действуя согласно которому можно получить нужное решение задачи.

Общие свойства алгоритма:

1) *детерминированность* - для всех процедур, которые можно назвать алгоритмами, последующая система величин должна однозначно определяться исходной системой величин;

2) *массовость* - процедура, которую естественно назвать алгоритмом, должна применяться для бесконечного множества различных возможных для данной задачи систем величин;

3) *результативность (направленность)* - на каждом этапе применения алгоритма должно быть ясно показано, что понимается под результатом решения задачи;

4) *дискретность* - алгоритм - это процедура последовательного получения систем величин;

5) *элементарность действий* - вычисление значений последую-

щей системы величин должно быть простым и носить локальный характер.

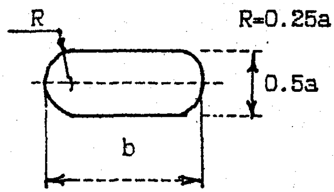
Все перечисленные свойства - эмпирические. Это те свойства, которые подмечены во всех построенных алгоритмах.

Для записи алгоритмов используются различные средства: словесное описание; формульно-словесное; язык блок-схем; язык операторных схем; язык проектирования программ PDL (Process Design Language); языки программирования.

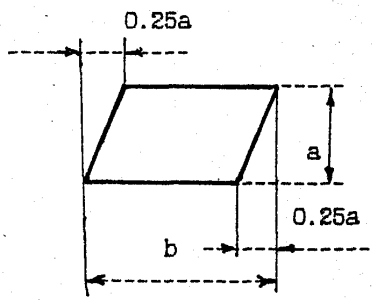
При проектировании программ довольно широко используется язык блок-схем.

Блок-схемой называют графическое изображение логической структуры алгоритма, в котором каждый этап переработки данных представлен в виде геометрических фигур (блоков), смысл которых однозначно определен предварительными соглашениями. Действующим ГОСТом (ГОСТ 19.003-80) предусмотрены следующие основные элементы блок-схем:

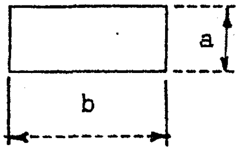
1) **пуск, останcвка**



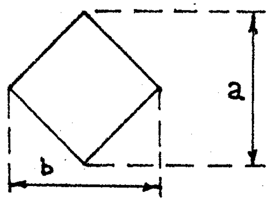
2) **ввод/вывод**



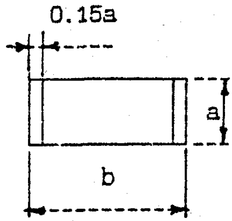
3) **процесс** (указывает на обработку данных)



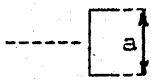
4) **альтернатива** (блок проверки условия, указывающий направление вычислительного процесса в зависимости от выполнения условия, указанного в блоке)



5) **вызов подпрограмм**



6) **комментарий**



Здесь размер a=10,15,20, ... мм; b=1.5a.

Блоки на блок-схеме нумеруются.

Блок-схемы используются на этапе проектирования и записываются с такой степенью детализации, чтобы на следующем этапе (этапе кодирования алгоритма на выбранном языке программирования) можно было бы каждому блоку однозначно поставить в соответствие нужную последовательность инструкций выбранного языка программирования.

Для решения практически любого класса задач можно разработать много вариантов алгоритмов, различающихся своим качеством. В настоящее время общепринятым является соблюдение определенных правил при выборе алгоритма: отыскивается не произвольный алгоритм решения задачи, а алгоритм, построенный с использованием фиксированного множества элементарных конструкций - *базисных управляющих структур*. Таковыми являются: *последовательность* (см. рис.1), *структуры ветвления* (см. рис.2), *циклические структуры* (см. рис.3).

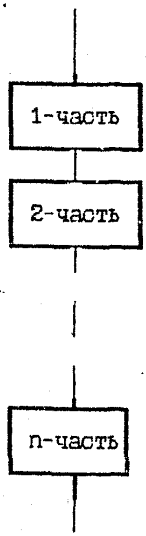


Рис. 1

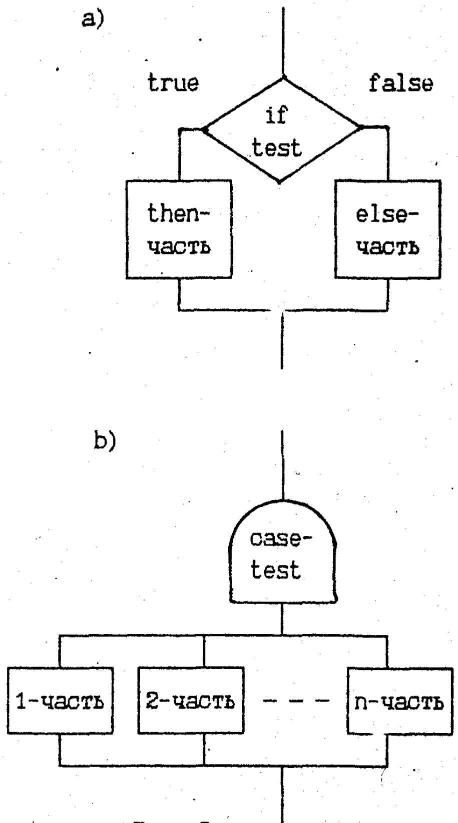


Рис. 2

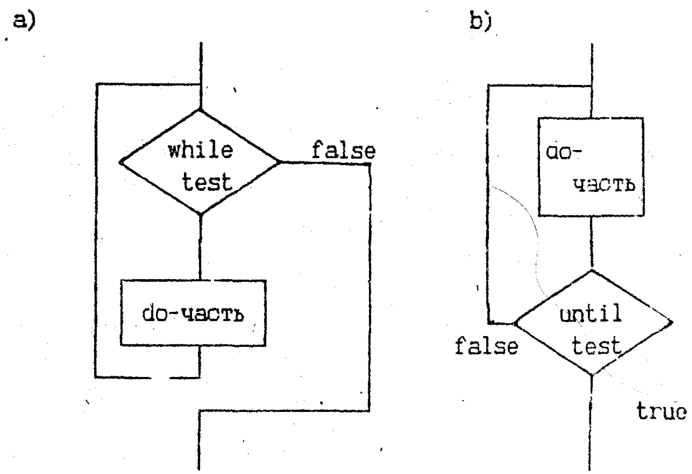


Рис. 3

1. ПРОЕКТИРОВАНИЕ И ОТЛАДКА ПРОГРАММ РЕАЛИЗАЦИИ ВЕТВЯЩИХСЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ

Вычислительный процесс называется ветвящимся, если в зависимости от значений исходных данных или промежуточных результатов вычислительный процесс реализуется по одному из нескольких возможных путей.

Структурированные программы такого рода строятся с использованием управляющих структур "*последовательность*" и структур *ветвления*.

Варианты заданий

1. Составить структурированную программу решения линейной системы алгебраических уравнений с двумя неизвестными:

$$a_{11}x_1 + a_{12}x_2 = a_{13};$$
$$a_{21}x_1 + a_{22}x_2 = a_{23}$$

2. Составить структурированную программу с входными данными X,C,P,T (все величины положительные), которая проверяет

можно ли построить параллелограмм из отрезков с длинами X,C,P,T.

3. Составить структурированную программу с входными данными A,B,C (все величины положительные), печатающую одно из четырех слов: "нельзя", "остроугольный", "прямоугольный", "тупоугольный" в зависимости от того, какой треугольник можно построить из отрезков с длинами A,B,C.

4. Составить структурированную программу с входными данными a,b,c,p,t, проверяющую, пройдет ли кирпич с ребрами a,b,c в прямоугольное отверстие со сторонами p,t. Просовывать кирпич в отверстие разрешается только так, чтобы каждое из его ребер было параллельно или перпендикулярно каждой из сторон отверстия.

5. Составить структурированную программу, в результате выполнения которой p:=1, если точка m(x,y) принадлежит заданной (замкнутой) области B, и p:=0 - в противном случае (см. рис. 4)

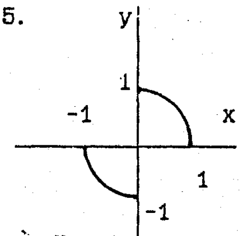


Рис.4

6. Составить структурированную программу, в результате выполнения которой переменной p:=1, если точка m(x,y) принадлежит заданной (замкнутой) области B, и p:=0 - в противном случае (см. рис. 5).

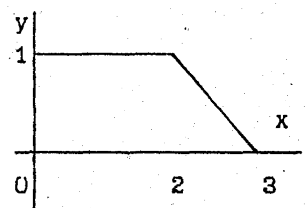


Рис. 5

7. Составить структурированную программу, в результате выполнения которой p:=1, если точка m(x,y) принадлежит заданной (замкнутой) области B, и p:=0 - в противном случае (см. рис. 6).

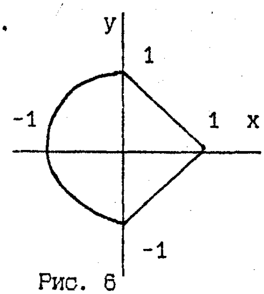
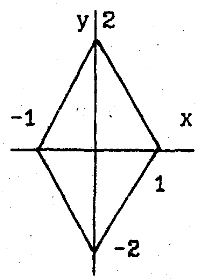


Рис. 6

8. 

Составить структурированную программу, в результате выполнения которой p:=1, если точка m(x,y) принадлежит заданной (замкнутой) области B, и p:=0 - в противном случае (см. рис. 7).

Рис. 7

9. Составить структурированную программу, в результате выполнения которой переменной p будет присвоено значение 1, если точка m(x,y) принадлежит заданной (замкнутой) области B, и значение 0 - в противном случае (см. рис. 8).

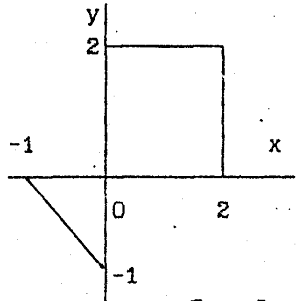


Рис. 8

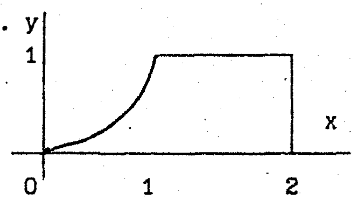
10. 

Рис. 9

Составить структурированную программу, в результате выполнения которой переменной P будет присвоено значение 1, если точка m(x,y) принадлежит заданной (замкнутой) области B, и значение 0 - в противном случае (см. рис. 9).

11. Составить структурированную программу для определения значения переменной z:

$$z=\begin{cases} x^2 + y^2 & \text{при } x^2 + y^2 <= 1; \\ x + y & \text{при } x^2 + y^2 > 1 \text{ и } y >= x; \\ 0.5 & \text{при } x^2 + y^2 > 1 \text{ и } y < x \end{cases}$$

12. Составить структурированную программу для определения значения переменной z:

$$z = \frac{\min(x,y) + 0.5}{1 + (\max(x,y))^2}$$

13. Составить структурированную программу для определения значения переменной z:

$$z = \begin{cases} 3/4 & \text{при } x > 0; \\ (x^2 + y^2)/8 & \text{при } x <= 0 \text{ и } x^2 + y^2 < a; \\ 4(x-y) & \text{при } x <= 0 \text{ и } x^2 + y^2 >= a \end{cases}$$

14. Составить структурированную программу для определения значения переменной t:

$$t = \frac{|x| + \min(x,y,z)}{|y| + \max(y,z)} .$$

15. Составить структурированную программу для определения значения переменной s:

$$s = \frac{\min\left(\dfrac{x + y + z}{3},\ x \cdot y \cdot z\right)}{1 + \min^2\left(\dfrac{x + y + z}{3},\ x \cdot y \cdot z\right)}$$

16. Составить структурированную программу для определения значения переменной t по заданному значению x:

$$t = (u + v)/(u^2 + v^2),$$

где $u = x^2$, $v = x/2$, если x принадлежит отрезку [-2,-0.5] или отрезку [0.5, 1] и $u = 0$, $v = 1 + x + x^2$, если x не принадлежит ни одному из этих отрезков.

17. Составить структурированную программу преобразования заданных вещественных значений X и T по правилу:

если X и T отрицательны, то каждое значение заменить его модулем;

если отрицательно только одно из них, то оба значения увеличить на 0.5;

если оба значения неотрицательны и ни одно из них не принаддежит отрезку [0.5, 2.0], то оба значения увеличить в 10 раз;

в остальных случаях значения X и T оставить без изменения.

18. Составить структурированную программу для решения биквадратного уравнения

$$x^4 + 2px^2 + q = 0.$$

19. Составить структурированную программу для определения полярных координат ρ и φ точки на плоскости по ее прямоугольным координатам x и y:

$$\rho = \sqrt{x^2 + y^2};$$

$$\varphi = \begin{cases} \operatorname{arctg}(y/x), & \text{если } x>0,\ y>0; \\ \pi/2, & \text{если } x=0, y>0; \\ \pi + \operatorname{arctg}(y/x), & \text{если } x<0; \\ 3\pi/2, & \text{если } x=0,\ y<0; \\ 2\pi + \operatorname{arctg}(y/x), & \text{если } x>0, y<0. \end{cases}$$

20. Составить структурированную программу для нахождения решения квадратного уравнения $ax^2 + bx + c = 0$ с полным исследованием возможных соотношений между коэффициентами этого уравнения и предполагая, что коэффициенты a, b, c не могут одновременно обратиться в нуль.

21. Составить структурированную программу для вычисления значения переменной z:

$$z = \begin{cases} \operatorname{arctg}\dfrac{x+y}{1-xy}, & \text{если } xy < 1; \\ \pi + \operatorname{arctg}\dfrac{x+y}{1-xy}, & \text{если } x>0 \text{ и } xy>1; \\ -\pi + \operatorname{arctg}\dfrac{x+y}{1-xy}, & \text{если } x<=0 \text{ и } xy>1; \\ \pi/2, & \text{если } xy = 1. \end{cases}$$

22. Составить структурированную программу для вычисления значения переменной t по заданным значениям вещественных переменных x и y:

$$t = \frac{\min(x,y) - \min(xy,x-y)}{\min(x+y,0.5)}.$$

23. Составить структурированную программу, которая из совокупности вещественных данных X1, X2, X3 образует последовательность данных P1, P2, P3, расположенных в порядке убывания.

24. Составить структурированную программу вычисления значения переменной m:

$$m = p(z) + 0.31,$$

где $z = x^2 + 3x$, $p(z) = \begin{cases} \sqrt{z}, & \text{если } z >= 0; \\ z^3 + z + 1, & \text{если } z < 0. \end{cases}$

25. Составить структурированную программу вычисления значения y:

$$y = \begin{cases} \dfrac{x^2+1}{\ln x}, & \text{если } 0 < x < 5; \\ \sqrt{x-5}, & \text{если } x >= 5; \\ \dfrac{x^3+x}{2}, & \text{если } x <= 0. \end{cases}$$

26. Составить структурированную программу вычисления значения переменной y:

$$y = \begin{cases} \sqrt{a^2 - b^2} - a, & \text{если } a^2 + b^2 >= 1; \\ \dfrac{a^2 + b^2}{a - b}, & \text{если } a^2 + b^2 < 1 \text{ и } a >= 0; \\ a^2 + b^2 - ab, & \text{если } a^2 + b^2 < 1 \text{ и } a < 0. \end{cases}$$

27. Составить структурированную программу вычисления значения переменной y:

$$y = \begin{cases} a + bx + cx^2, & \text{если } x>3; \\ (a* \sin x)^2, & \text{если } 2<=x<=3; \\ \sqrt{a + bx}, & \text{если } 0<=x<=2; \\ a*\ln|x|, & \text{если } -2<=x<=0; \\ (ax^4 + bx^4)/4, & \text{если } x<=-2. \end{cases}$$

28. Составить структурированную программу вычисления значения переменной z:

$$z = \ln|y|,$$

где $y = \begin{cases} x^3 - 1 & \text{при } x>0; \\ \cos x + 1 & \text{при } x<=0. \end{cases}$

29. Составить структурированную программу вычисления значения переменной z:

$$z = \begin{cases} \pi\ , \text{ при } x>0; \\ \dfrac{(x^2 + y^2)}{8}\ , \text{ при } x<=0 \text{ и } x^2 + y^2<a; \\ 4(x - y)\ , \text{ при } x<=0 \text{ и } x^2 + y^2>= \ . \end{cases}$$

30. Составить структурированную программу вычисления значения переменной f:

$$f = \begin{cases} \pi - \mathrm{arctg}(x/y)\ , & \text{если } x>0 \text{ и } y\neq 0; \\ \dfrac{x^3 - \ln x}{y^2 - 4}\ , & \text{если } x>0 \text{ и } y = ?; \\ \sqrt{|x|}\ y^5 & \text{в остальных случаях.} \end{cases}$$

2. ПРОЕКТИРОВАНИЕ И ОТЛАДКА ПРОГРАММ РЕШЕНИЯ АЛГЕБРАИЧЕСКИХ И ТРАНСЦЕНДЕНТНЫХ УРАВНЕНИЙ

В практических вычислениях довольно часто приходится решать уравнения вида

$$f(x) = 0, \qquad (1)$$

где функция f(x) определена и непрерывна на некотором интервале $a < x < b$.

Всякое значение x, обращающее функцию f(x) в нуль, называется корнем уравнения (1). Найти точные значения корней уравнения удается лишь в частных случаях. Поэтому разработаны методы численного решения уравнений вида (1), которые позволяют отыскивать приближенные значения корней этого уравнения. При этом приходится решать две задачи:

1) отделение корней, то есть отыскание достаточно малых

областей, в каждой из которых заключен один и только один корень уравнения;

2) вычисление корней с заданной точностью.

Для отделения корней можно воспользоваться графическим способом. Для вычисления корней с заданной точностью существует множество методов, наиболее употребительными из которых являются метод итераций, метод Ньютона, метод хорд и метод половинного деления.

Метод итераций - это процесс последовательного вычисления чисел x_i $(i = 1,2,3,\dots)$ по итерационной формуле $x_i = \varphi(x_i)$.

Итерационная формула может быть получена из уравнения (1) представлением его в виде $x = \varphi(x)$, что можно сделать всегда и многими способами. Например, выделить из уравнения (1) x, а остальное перенести в правую часть. Или представить уравнение (1) в виде $x = x + \lambda f(x)$, где λ - некоторая константа. При этом $\varphi(x) = x + \lambda f(x)$.

Процесс итераций продолжается до тех пор, пока для двух последовательных приближений x_{i-1} и x_i не будет обеспечено выполнение условия $|x_i - x_{i-1}| < \varepsilon$. При практическом нахождении корней методом итераций нужно представить $\varphi(x)$ так, чтобы $|\varphi'(x)| < 1$.

Структурированная схема алгоритма, соответствующая методу итераций, представлена на рис. 10.

Метод Ньютона: процесс уточнения корня уравнения осуществляется по итерационной формуле

$$x_i = x_{i-1} - f(x_{i-1})/f'(x_{i-1})$$

до тех пор, пока соблюдается условие $|x_i - x_{i-1}| > \varepsilon$.

Особенности применения метода Ньютона:

- на отрезке $[a,b]$, на котором функция имеет один корень, $f'(x)$ и $f''(x)$ определены, непрерывны и сохраняют постоянные знаки;

- начальное приближение x_0 целесообразно выбирать так, чтобы было выполнено условие $f(x_0) \cdot f''(x_0) > 0$;

- метод эффективен для решения тех уравнений, для которых значение модуля производной $|f'(x)|$ вблизи корня достаточно велико.

Метод половинного деления. Если

дано уравнение f(x) = 0, где функция f(x) непрерывна на отрезке xa,b] и f(a)·f(b) < 0, то метод половинного деления всегда приводит к результату. Он требует большого объема вычислительной работы, но его целесообразно использовать, когда из-за сложности функции f(x) вычисление f'(x) в методе Ньютона и φ'(x) в методе итераций затруднительно.

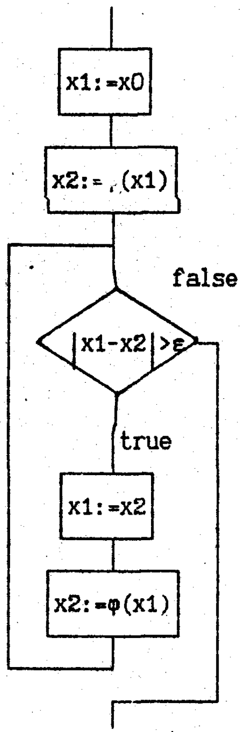


Рис. 10

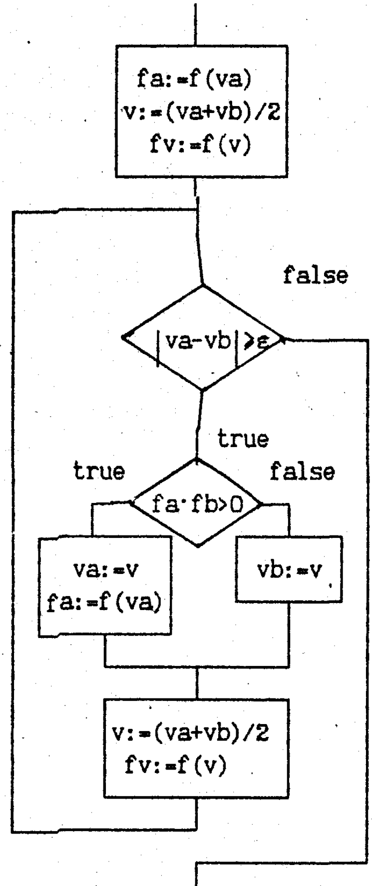


Рис. 11

Процесс деления отрезка пополам продолжается до тех пор, пока длина отрезка, на концах которого функция имеет противоположные знаки, не будет меньше наперед заданного числа ε.

Структурированная схема алгоритма, соответствующая методу половинного деления, представлена на рис. 11.

Варианты заданий

Проверить условие сходимости, записать итерационную формулу, выбрать начальное приближение корня из указанного отрезка, составить структурированную программу для нахождения корня уравнения с абсолютной погрешностью $\varepsilon=10^{-3}$. В программе предусмотреть подсчет и выдачу на печать количества итераций, необходимых для вычисления значения корня с заданной точностью.

Варианты заданий приведены в табл.1. Там же указано приближенное значение корня, с которым необходимо сравнить полученные в результате выполнения программы значения.

Таблица 1

#	Уравнение	Отрезок, содержащий корень	Метод численного решения	Приближенное значение корня
1	2	3	4	5
1	$3\sin\sqrt{x} + 0.35x - 3.8 = 0$	[2;3]	итерац.	2.2985
2	$0.25x^3 + x - 1.25 = 0$	[0;2]	Ньютона	1.0001
3	$x + \sqrt{x} + \sqrt[3]{x} - 2.5 = 0$	[0.4;1]	Пол.дел	0.7376
4	$x - 1/(3 + \sin(3.6x)) = 0$	[0;0.85]	итерац.	0.2624
5	$0.1x^2 - x\ln x = 0$	[1;2]	Ньютона	1.1183
6	$\mathrm{tg}x - (1/3)\mathrm{tg}^3x + (1/5)\mathrm{tg}^5x - 1/3 = 0$	[0;0.8]	Пол.дел	0.3333
7	$\arccos x - \sqrt{1 - 0.3x^3} = 0$	[0;1]	итерац.	0.5629
8	$3x - 4\ln x - 5 = 0$	[2;4]	Ньютона	3.2300
9	$\cos(2/x) - 2\sin(1/x) + 1/x = 0$	[1;2]	Пол.дел	1.8756
10	$\sqrt{1 - 0.4x^2} - \arcsin x = 0$	[0;1]	итерац.	0.7672
11	$e^x - e^{-x} - 2 = 0$	[0;1]	Ньютона	0.8814
12	$\sin(\ln x) - \cos(\ln x) + 2\ln x = 0$	[1;3]	Пол.дел	1.3749

Продолжение таблицы 1

1	2	3	4	5
13	$x - 2 + \sin(1/x) = 0$	[1.2;2]	итерац.	1.3077
11	$e^x + \ln x - 10x = 0$	[3;4]	Ньютона	3.5265
15	$\cos x - e^{-x^2/2} + x - 1 = 0$	[1;2]	Пол.дел	1.0804
16	$1 - x + \sin x - \ln(1+x) = 0$	[0;1.5]	итерац.	1.1474
17	$3x - 14 + e^x - e^{-x} = 0$	[1;3]	Ньютона	2.0692
18	$\sqrt{1 - x} - \mathrm{tg}x = 0$	[0;1]	Пол.дел	0.5768
19	$x + \cos(x^{0.52} + 2) = 0$	[0.5;1]	итерац.	0.9892
20	$3\ln^2 x + 6\ln x - 5 = 0$	[1;3]	Ньютона	1.8832
21	$\sin x^2 + \cos x^2 - 10x = 0$	[0;1]	Пол.дел	0.1010
22	$x^2 - \ln(1 + x) - 3 = 0$	[2;3]	итерац.	2.0267
23	$2x \cdot \sin x - \cos x = 0$	[0.4;1]	Ньютона	0.6533
24	$e^x + \sqrt{1 + e^{2x}} - 2 = 0$	[-1;0]	Пол.дел	-0.2877
25	$\ln x - x + 1.8 = 0$	[2;3]	итерац.	2.8459
26	$x \cdot \mathrm{tg}x - 1/3 = 0$	[0.2;1]	Ньютона	0.5472
27	$\mathrm{tg}(x/2) - \mathrm{ctg}(x/2) + x = 0$	[1;2]	Пол.дел	1.0769
28	$0.4 + \mathrm{arctg}\sqrt{x} - x = 0$	[1;2]	итерац.	1.2388
29	$\sqrt{1 - x} - \cos\sqrt{1 - x} = 0$	[0;1]	Ньютона	0.4538
30	$0.6 \cdot 3^x - 2.3x - 3 = 0$	[2;3]	Пол.дел	2.4200

3. ПРОЕКТИРОВАНИЕ И ОТЛАДКА ПРОГРАММ ВЫЧИСЛЕНИЯ ЧАСТИЧНЫХ СУММ ФУНКЦИОНАЛЬНЫХ РЯДОВ

К вычислению частичных сумм ряда сводится вычисление значений многих функций. Задача заключается в нахождении суммы некоторого количества слагаемых

$$S = \sum_n a_n(x).$$

Каждое слагаемое суммы зависит от параметра x и номера n, определяющего место этого слагаемого в сумме.

Структурированная схема алгоритма, соответствующая вычислению суммы заранее известного количества слагаемых, строится с использованием элементарных структур типа "последовательность" и "do-until".

Структурированная схема алгоритма, соответствующая вычислению суммы с точностью до очередного члена ряда, строится с использованием элементарных структур типа "последовательность" и "while-do".

Обычно формула общего члена суммы принадлежит к одному из следующих двух типов:

$$1)\quad \frac{x^n}{n!}\,; \quad (-1)^n \frac{x^{2n+1}}{2n+1}\,; \quad \frac{x^{2n}}{(2n)!}\,;$$

$$2)\quad \frac{\cos(nx)}{n}\,; \quad \frac{\sin((2n-1)x)}{2n-1}\,; \quad \frac{\cos(2nx)}{4n^2-1}\,.$$

В первом случае для вычисления очередного члена суммы целесообразно использовать рекуррентные соотношения, то есть выражать последующий член суммы через предыдущий. Это позволяет существенно сократить объем вычислительной работы. Кроме того, вычисление члена суммы по общей формуле в ряде случаев нецелесообразно.

Во втором случае член суммы целесообразно представить в виде двух сомножителей, один из которых вычисляется по рекуррентной формуле, а другой - непосредственно.

Сумма вычисляется циклически. При каждом прохождении цикла номер члена суммы изменяется на единицу, а сумма изменяется на значение n-го члена: $S_n = S_{n-1} + a_n$.

При составлении схем алгоритмов нет необходимости использовать переменные с индексами S_n, a_n, поскольку в вычислениях одновременно участвуют лишь два значения S_n и S_{n-1}, а также a_n и a_{n-1}. По этому вместо S_n и S_{n-1} можно использовать скалярную переменную S, а вместо a_n и a_{n-1} - скалярную переменную a.

На рис. 12 и рис. 13 представлены структурированные схемы алгоритмов вычисления частичных сумм функциональных рядов

$$S = \sum_{n=2}^{25} (-1)^n \frac{n}{n^2 - 1} \sin(nx)$$

и

$$S = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots, \text{ пока } \left| \frac{x^{2n+1}}{(2n+1)!} \right| > \varepsilon.$$

Варианты заданий

Составить структурированную программу вычисления суммы с точностью до очередного члена ряда $|a| < \varepsilon$ ($\varepsilon = 10^{-3}$) и значения соответствующей ей функции y для фиксированного значения x из указанного диапазона. Предусмотреть подсчет и выдачу на печать количества слагаемых, при котором достигается заданная точность вычисления.

Формулы для вычисления суммы и функции, диапазон допустимых значений x для каждого из 30 вариантов указаны в табл.2.

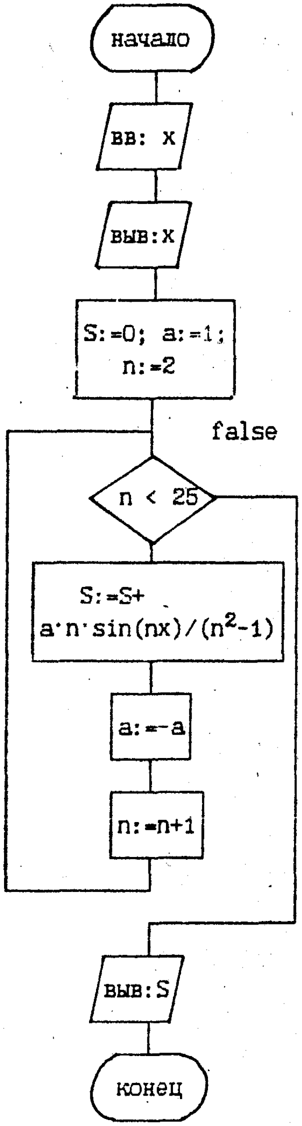

начало
вв: x
выв:x
S:=0; a:=1;
n:=2
false
n < 25
S:=S+
a·n·sin(nx)/(n2-1)
a:=-a
n:=n+1
выв:S
конец

Рис. 12

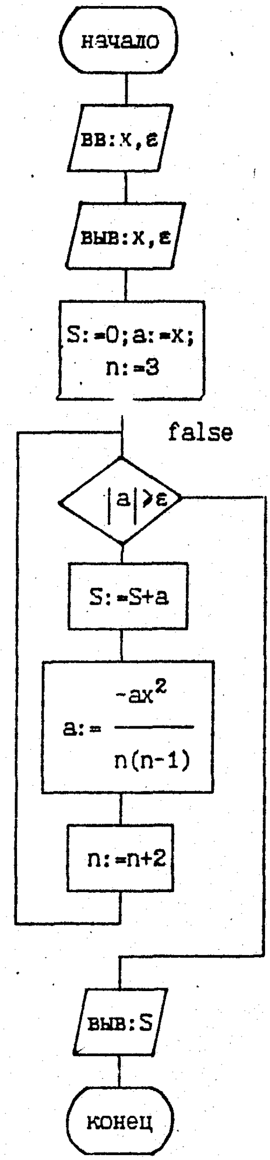

начало
вв:x,ε
выв:x,ε
S:=0;a:=x;
n:=3
false
|a|>ε
S:=S+a
-ax2
a:=
n(n-1)
n:=n+2
выв:S
конец

Рис. 13

4. ПРОЕКТИРОВАНИЕ И ОТЛАДКА ПРОГРАММ ТАБУЛИРОВАНИЯ ЗНАЧЕНИЙ ФУНКЦИИ

Варианты заданий

Составить структурированную программу табулирования значений функций $S(x)$ и $y(x)$ в указанном диапазоне изменения аргумента. Сумму вычислить из k членов ряда. Шаг изменения аргумента принять равным $\Delta x = |x_к - x_н|/9$.

Формулы для $S(x)$ и $y(x)$, диапазон допустимых значений, количество членов суммы указаны в табл.2.

Таблица 2

#	Функции S(x), y(x)	Диапазон	k
1	$S = 1 + \frac{\ln 3}{1!} x + \frac{\ln^2 3}{2!} x^2 + \ldots + \frac{\ln^n 3}{n!} x^n$ $y = 3^x$	$0.1 < x < 1$	10
2	$S = \cos x + \frac{\cos 2x}{2} + \ldots + \frac{\cos nx}{n}$ $y = -\ln\lvert 2\sin(x/2)\rvert$	$\pi/5 < x < 9\pi/5$	40
3	$S = x - \frac{x^3}{3!} + \ldots + (-1)^n \frac{x^{2n+1}}{(2n+1)!}$ $y = \sin x$	$0.1 < x < 1$	10
4	$S = \sin x - \frac{\sin 2x}{2} + \ldots + (-1)^n \frac{\sin nx}{n}$ $y = x/2$	$\pi/5 < x < 4\pi/5$	40

Продолжение таблицы 2

1	2	3	4
5	$S = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \ldots + \frac{x^n}{n!}$ $y = e^x$	$1<x<2$	15
6	$S = 1 + \frac{\cos(\pi/4)}{1!} x + \ldots + \frac{\cos(n\pi/4)}{n!} x^n$ $y = e^{x\cos(\pi/4)} \cos(x \cdot \sin(\pi/4)$	$0.1<x<1$	25
7	$S = 1 - \frac{x^2}{2!} + \ldots + (-1)^n \frac{x^{2n}}{(2n)!}$ $y = \cos x$	$0.1<x<1$	10
8	$S = x \cdot \sin\frac{\pi}{4} + x^2 \sin 2\frac{\pi}{4} + \ldots + x^n \sin n\frac{\pi}{4}$ $y = \frac{x \cdot \sin(\pi/4)}{1 - 2x \cdot \cos(\pi/4) + x^2}$	$0.1<x<0.8$	40
9	$S = x + \frac{x^5}{5} + \ldots + \frac{x^{4n+1}}{4n+1}$ $y = 0.25\ln[(1+x)/(1-x)] + 0.5\mathrm{arctg}x$	$0.1<x<0.8$	30
10	$S = 1 + \frac{\cos x}{1!} + \ldots + \frac{\cos(nx)}{n!}$ $y = e^{\cos x} \cdot \cos(\sin x)$	$0.1<x<1$	20

Продолжение таблицы 2

1	2	3	4
11	$S = 1 + \dfrac{3x^2}{1!} + \ldots + \dfrac{2n+1}{n!} x^{2n}$ $y = (1 + 2x^2)e^{x^2}$	$0.1<x<1$	10
12	$S = \dfrac{x\cos(\pi/3)}{1} + \dfrac{x^2\cos(2\pi/3)}{2} + \ldots$ $\ldots + \dfrac{x^n\cos(n\pi/3)}{n}$ $y = -0.5\ln(1-2x\cos(\pi/3) + x^2)$	$0.1<x<0.8$	35
13	$S = \dfrac{x-1}{x+1} + \dfrac{(x-1)^3}{3(x+1)^3} + \ldots + \dfrac{(x-1)^{2n+1}}{(2n+1)(x+1)^{2n+1}}$ $y = 0.5\ln x$	$0.2<x<1$	10
14	$S = -\cos x + \dfrac{\cos 2x}{2^2} + \ldots + (-1)^n \dfrac{\cos nx}{n^2}$ $y = \dfrac{1}{4} (x^2 - \dfrac{\pi^2}{3})$	$\pi/5<x<\pi$	20
15	$S = \dfrac{x^3}{3} - \dfrac{x^5}{15} + \ldots + (-1)^{n+1} \dfrac{x^{2n+1}}{4n^2 - 1}$ $y = (1+x^2)\mathrm{arctg}(x)/2 - x/2$	$0.1<x<1$	50

Продолжение таблицы 2

1	2	3	4
16	$S= \sin x + \frac{\sin 3x}{3} + \ldots + \frac{\sin(2n-1)x}{2n-1}$ $y= \pi/4$	$\frac{\pi}{10} < x < \frac{9\pi}{10}$	40
17	$S= 1 + \frac{x^2}{2!} + \ldots + \frac{x^{2n}}{(2n)!}$ $y= (e^x + e^{-x})/2$	$0.1 < x < 1$	10
18	$S= \frac{\cos 2x}{3} + \frac{\cos 4x}{15} + \ldots + \frac{\cos(2nx)}{4n^2-1}$ $y= 1/2 - \pi\lvert\sin x\rvert/4$	$0.1 < x < 0.8$	50
19	$S= 1 + \frac{2x}{1!} + \ldots + \frac{(2x)^n}{n!}$ $y= e^{2x}$	$0.1 < x < 1$	20
20	$S= 1 + 2\frac{x}{2} + \ldots + \frac{n^2+1}{n!} \cdot \frac{x^n}{2^n}$ $y= (x^2/4 + x/2 + 1)e^{x/2}$	$0.1 < x < 1$	30
21	$S= x - \frac{x^3}{3} + \ldots + (-1)^n \frac{x^{2n+1}}{2n+1}$ $y= \mathrm{arctg}\, x$	$0.1 < x < 0.5$	40

Продолжение таблицы 2

1	2	3	4
22	$S = 1 - \frac{3}{2}x^2 + \dots + (-1)^n \frac{2n^2+1}{(2n)!} x^{2n}$ $y = (1-0.5x^2)\cos x - 0.5x \cdot \sin x$	$0.1 < x < 1$	35
23	$S = -\frac{(2x)^2}{2} + \frac{(2x)^4}{24} - \dots + (-1)^n \frac{(2x)^{2n}}{(2n)!}$ $y = 2(\cos^2 x - 1)$	$0.1 < x < 1$	15
24	$S = -(1+x)^2 + \frac{(1+x)^4}{2} - \dots + (-1)^n \frac{(1+x)^{2n}}{n}$ $y = \ln \frac{1}{2 + 2x + x^2}$	$-2 < x < -0.1$	40
25	$S = x + \frac{x^3}{3!} + \dots + \frac{x^{2n+1}}{(2n+1)!}$ $y = (e^x - e^{-x})/2$	$0.1 < x < 1$	20
26	$S = \frac{x}{3!} + \frac{4x^2}{5!} + \dots + \frac{n^2}{(2n+1)!} x^n$ $y = \frac{1}{4} \left(\frac{x+1}{\sqrt{x}} \cdot \operatorname{sh}\sqrt{x} - \operatorname{ch}\sqrt{x} \right)$	$0.2 < x < 0.8$	20

Окончание таблицы 2

1	2	3	4
27	$S = x\cdot\cos(\pi/4) + x^2\cdot\cos(2\pi/4) + \ldots + x^n\cdot\cos(n\pi/4)$ $y = \dfrac{x\cdot\cos(\pi/4) - x^2}{1 - 2x\cdot\cos(\pi/4) + x^2}$	$0.1<x<0.8$	40
28	$S = 3x + 8x^2 + \ldots\ n(n+2)x^n$ $y = x(3 - x)/(1 - x)^3$	$0.1<x<0.8$	40
29	$S = \cos x + \dfrac{\cos(3x)}{3^2} + \ldots + \dfrac{\cos((2n-1)x)}{(2n-1)^2}$ $y = \pi^2/8 - \lvert x\rvert\pi/4$	$\pi/5<x<\pi$	40
30	$S = \dfrac{x^2}{2} - \dfrac{x^4}{12} + \ldots + (-1)^{n+1}\dfrac{x^{2n}}{2n(2n-1)}$ $y = x\cdot\mathrm{arctg}(x) - \ln\sqrt{1+x^2}$	$0.1<x<0.8$	10

5. СРЕДСТВА УПРАВЛЕНИЯ КОНСОЛЬЮ В ТУРБОПАСКАЛЕ

Консоль оператора физически состоит из двух устройств: клавиатуры и монитора. Программы управления консолью сосредоточены в Unit CRT. Они позволяют считывать символ с клавиатуры без отображения его на экране, устанавливать различные режимы отображения текста, определять "окна", определять и изменять положение курсора, управлять звуковым динамиком.

Рассмотрим процедуры и функции, входящие в состав этого пакета.

Функция *keypressed: boolean* возвращает TRUE, если на клавиатуре была нажата клавиша, порождающая символ.

Функция *readkey: char* возвращает символ нажатой клавиши или ждет нажатия клавиши.

Процедура *textmode(mode: word)* устанавливает режим отображения текста на экране. Все допустимые режимы определены константами в unit CRT (*mode* может иметь, например, значения BW40, CO40, BW80, CO80 и др.).

Процедура *window(xup, yup, xlow, ylow : byte)* определяет "окно". Курсор позиционируется в позицию (1,1) "окна".

Процедура *gotoxy(x,y : byte)* перемещает курсор в новую координату (координаты определяются в пределах "окна").

Функции *wherex : byte* и *wherey : byte* возвращают координаты текущего положения курсора.

Процедура *clrscr* производит очистку экрана.

Процедура *clreol* очищает конец текущей строки.

Процедуры *insline* и *delline* позволяют вставить или удалить строку в соответствии со своими названиями.

Процедуры *textcolor(color: byte)*, *textbackground(color: byte)* позволяют установить цвет и фон для выводимых на экран символов.

Процедуры *highvideo*, *lowvideo*, *normvideo* позволяют устанавливать пониженную, повышенную или обычную яркость для выводимых на экран символов.

Последовательность процедур *sound(hz: word)*, *delay(ms: word)*, *nosound* позволяет управлять динамиком, заставив его звучать на частоте *hz* герц в течение *ms* миллисекунд.

С О Д Е Р Ж А Н И Е