

681.3

MSY

МИНИСТЕРСТВО ПУТЕЙ СООБЩЕНИЯ СССР

л 208

МОСКОВСКИЙ ОРДЕНА ЛЕНИНА
И ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ
ИНСТИТУТ ИНЖЕНЕРОВ ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

Кафедра прикладной математики

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К ЛАБОРАТОРНОМУ ПРАКТИКУМУ
ПО ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКЕ**

Москва — 1986

681.3
154
МИНИСТЕРСТВО ПУТЕЙ СООБЩЕНИЯ СССР

Московский ордена Ленина
и ордена Трудового Красного Знамени
институт инженеров железнодорожного транспорта

Кафедра прикладной математики

У т в е ж д е н о
редакционно-издательским
советом института

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К ЛАБОРАТОРНОМУ ПРАКТИКУМУ
ПО ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКЕ

для студентов специальности
"СТРОИТЕЛЬСТВО ЖЕЛЕЗНЫХ ДОРОГ,
ПУТЬ И ПУТЕВОЕ ХОЗЯЙСТВО"

Москва - 1986

Методические указания к лабораторному практикуму по вычислительной технике предназначены для студентов специальности "Строительство железных дорог". Проводится в 1-м семестре и рассчитан на 34 ч. Студенты выполняют лабораторные работы на ЭВМ "Искра-226". Целью практикума является ознакомление студентов с принципами работы ЭВМ, изучение алгоритмического языка БЕЙСИК и основ программирования, изучение основных принципов работы за пультом ЭВМ, отладки и проведения расчетных работ.

При описании основных операторов языка БЕЙСИК, авторы, учитывая специфику проведения занятий в ВЦ института, отошли от традиционного изложения. В указаниях после алфавита языка, записи чисел, переменных и основных принципов программирования излагаются операторы ввода и вывода информации, а затем операторы присваивания, перехода, цикла и т.д.

В В Е Д Е Н И Е

1. Функциональная схема ЭВМ

Для выполнения заданий ЭВМ "Искра-226" имеет следующие устройства:

1) устройство управления, которое служит для управления работой ЭВМ в автоматическом режиме, организации выполнения управляющих команд пользователя и выполнения требуемых арифметических и логических операций;

2) оперативное запоминающее устройство (ОЗУ), для хранения программ (память), промежуточных и окончательных результатов;

3) долговременное запоминающее устройство (ДЗУ) для длительного хранения информации на магнитной ленте (МЛ) МК-60 и магнитном диске (МД). Соответственно в ЭВМ включены устройства для записи и вывода информации с МЛ и МД;

4) блок отображения символьной и графической информации (БОСГИ) для отображения на экране вводимой с клавишного устройства информации и, по желанию пользователя, содержимого памяти, информации о выполнении программы, результатов и т.п.;

5) клавишное устройство для ввода программы, начальных данных, управления работой ЭВМ и редактирования;

6) блок питания;

7) цифрпечать, для вывода результатов и программы на бумагу.

Основные связи между устройствами указаны на рис. 1.

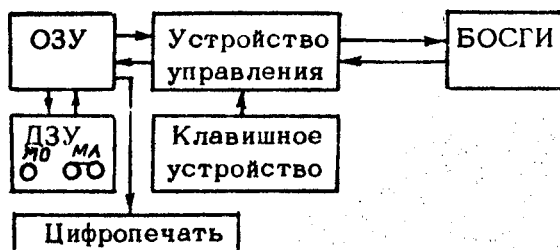


Рис. 1

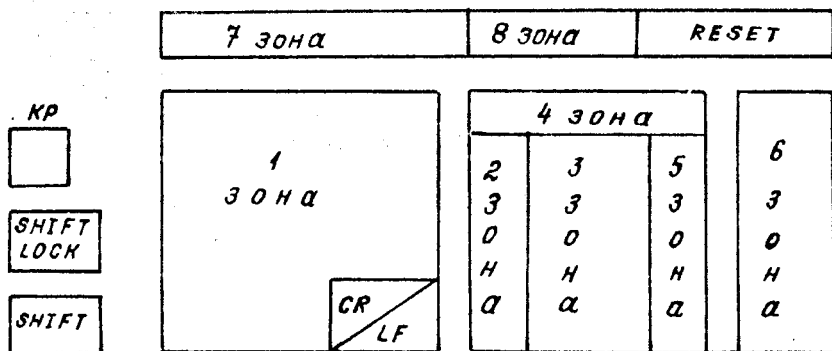
2. Клавишное устройство

На клавишном устройстве можно выделить восемь функциональных зон. Клавиши некоторых зон имеют два или три регистра.

Первая зона имеет по нижнему регистру часто встречающиеся буквосочетания и слова, по среднему – буквы русского алфавита, по верхнему – буквы латинского алфавита и некоторые знаки; вторая зона содержит клавиши управления счетом и редактирования при вводе; третья – цифры 0 ÷ 9 и десятичную точку; четвертая – часто используемые операторы языка; пятая – имеет клавиши арифметических операций, скобок и некоторых математических функций; шестая зона – зона редактирования текста программы; восьмая зона содержит клавиши управления курсором.

Изучение седьмой зоны программой практикума не предусмотрено.

Для переключения регистров служат клавиши *SHIFT*, *SHIFT LOCK*, красная клавиша, которую в дальнейшем будем определять как лат/рус.



Для включения нижнего регистра на длительное время необходимо нажать клавишу *SHIFT LOCK* до фиксации. Выключение клавиши *SHIFT LOCK* производится нажатием клавиши *SHIFT*. Если необходимо включить нижний регистр на короткое время, то достаточно нажать клавишу *SHIFT* и удерживать ее в утопленном состоянии. При отпускании клавиши *SHIFT* нижний регистр отключается.

При выключенном нижнем регистре состояние клавиатуры зависит от положения клавиши лат/рус. При утопленной клавише лат/рус включен верхний регистр, а при отпущенной - средний.

В зонах, имеющих два регистра, объединены средний и верхний, а в зонах, имеющих один регистр, - все три.

В н и м а н и е! Несмотря на тождественность написания, набор буквы русского алфавита вместо латинской ЭВМ квалифицирует как ошибку. Текст задания можно вводить как с использованием слов и словосочетаний, имеющихсся на клавиатуре, так и посимвольно.

На правой стороне клавиатуры, вверху расположена кнопка " *RESET* ", которая используется для подачи команды аварийного останова. Нажатие кнопки " *RESET* " прекращает выполнение оператора, действие устройств ввода-вывода и передает управление пользователю.

Текст программы со значениями всех ранее перечисленных параметров остается в памяти. В нижнем правом углу первой зоны находится кнопка "CR/LF", при нажатии которой очередная набранная строка символов передается либо к исполнению, либо в память, в зависимости от содержания.

3. Решение задач с помощью ЭВМ

Решение задачи на ЭВМ можно разделить на следующие этапы:

- 1) постановка задачи;
- 2) выбор метода решения и составление блок-схемы (алгоритма) программы;
- 3) составление программы - задания для ЭВМ;
- 4) ввод программы в память ЭВМ;
- 5) отладка программы и вычисления на ЭВМ;
- 6) анализ результатов.

ОПИСАНИЕ ЯЗЫКА БЕЙСИК

Программирование задач на ЭВМ "Искра-226" осуществляется на алгоритмическом языке БЕЙСИК. В настоящих указаниях рассматривается подмножество языка для ЭВМ "Искра-226", максимально приближенное к возможностям ЭВМ серии СМ, чтобы облегчить, в случае необходимости, переход к вычислениям на этих машинах.

1. Алфавит языка БЕЙСИК

Алфавит языка состоит из следующих символов:

- а) 26 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;
- б) 31 буква русского алфавита (без Ё и ъ);
- в) десятичные цифры от 0 до 9;
- г) знаки: !, ", #, \$, %, &, ', (,), /, J, ^, ©, [,], >, =, <, ;, :, \, ;, -, ., +, *.

При записи программы на бумаге ноль обычно подчеркивают, чтобы отличить его от буквы О.

2. Запись чисел в программе

В программе различаются целые и действительные числа (константы).

Целое число имеет вид $\pm N$, причем знак $+$ может быть опущен, а N — любое целое число, удовлетворяющее неравенству $0 \leq |N| \leq 7999$. Например, 15, -7895, +1, -3.

Действительные числа могут быть записаны в двух формах, но должны удовлетворять неравенству

$$1 \cdot 10^{-99} \leq |Z| \leq (1 - 10^{-12}) \cdot 10^{99}$$

Если число имеет форму $\pm N.M$; где N, M — целые числа, а точка обозначает запятую, отделяющую целую часть от дробной, то говорят, что число записано в естественном виде. Общее количество цифр в N и M не должно превышать 13. Знак $+$ может быть опущен. Если целая или дробная части равны нулю, то они соответственно могут быть опущены. Например, 0571, .571, -7.981, -.675, +0572, .572, 573., 573.0.

Если число имеет форму XEK , где X — действительное число (мантисса), записанное в естественном виде, E — разделительный символ, а K — целое число, причем $0 \leq |K| \leq 99$, то говорят, что число записано в экспоненциальной форме, оно равно $Z = XEK = X \cdot 10^K$, при этом подразумевается, что данное число принадлежит указанному выше допустимому диапазону действительных чисел, например

$$6,4E+10 = 6,4 \cdot 10^{10}; \quad -10 \cdot 4 = 10,58 \cdot 10^{-4} = 000,1058$$

П р и м е р ы

1. Запись чисел в виде целых констант: 23, +518, -6347. Запись 9745 неправильна, так как это число больше 7999 и не может рассматриваться как целая константа.

2. Запись действительных чисел в естественной форме 0.0781, .5768, 171.3015, +63.158, -0.00811, -075.

-46,-71.195. Запись 1783.43496170025 неправильна, так как использует более 13 цифр.

3. Запись чисел в экспоненциальной форме $25.8E 14$, $45.1015E + 7,79.1E -05$, $-.4590E + 18$, $-85 E 9,184 E 3$,

Упражнение 1

1. Представить приведенные ниже числа в виде целых констант:

$0,0237 \cdot 10^5$; $-32,2 \cdot 10^5$; $0,64 \cdot 10^3$;
 $(0,3)^3 \cdot 10^4$; $+128 \cdot 2^3$; -0 ; $+26,3 \cdot 10^7$; $0,6701 \cdot 10^4$.

2. Указать, какие из приведенных ниже чисел являются целыми константами:

23, -23, +23, $0,23 \cdot 10^2$; 0, -0, +0, 0.00, 444, $3 \cdot 10^{-7}$;
 $-0,1634 \cdot 10$, +2156, -2^{-12} ; 4, 2^8 ; $-64 \cdot 2^{-3}$; 0,0006,
27632, 0027106.

3. Указать, какие из приведенных ниже чисел можно рассматривать как действительные константы в естественной форме:

-0.2423 ; 10^3 ; 24,67 ; + 7.83462 ;
 -0.000000 ; $26.3 \cdot 10^{-0,3}$;
6427646.1 ; 0,2847 ; -0.666666 ;
 $+2237$; $0.2 \cdot 10^{-07}$; -899623. 148091519 ;
 $+0.0$, -0, 0, -16^2 ; 5556555 ;
 $+34.000000$, -2,36 ; -0.00.

4. Указать, какие из приведенных ниже записей можно рассматривать как действительные константы в экспоненциальной форме:

-24.06531 ; -1.E ; 2.3 E (-2) ; +E 04 ;
-E-02 ; 1.0E 08 ; - 1.0E, + 0, 001 E -6 ;
1.0E 8.2 ; - 0.0210⁰² ; 27621, 1. E 112 ;
 $-14.00/1 E - 0.3$; 2.702 E -02, E 01 ;
 $-13E-012 E 05$; 0101E 15, 100 E -06.

5. Представить приведенные ниже числа в виде действительных констант без экспонент:

-10^3 ; 10^5 ; $211,02 \cdot 10^4$; $-(0,5)^2 \cdot 10^{-2}$;
 $314 \cdot 2^6$; $0,26 \cdot 10^{-8}$; 64673, 767801 ;
 $-0,02345 \cdot 10^{14}$; 0,0 ; 10^{-9} ; 99,9 ;
 -00556 (Это число можно записать в виде -5556.) ;

$-39,643362802$; $2,5 \cdot 10^{-6}$; $211,02 \cdot 10^{-2}$;
 $+0,0000$; $-0,000001$; 10^8 ; $-1 \cdot 10^6$; $1,0 \cdot 10^{-0}$; 10^8 .

6. Представить приведенные ниже числа в виде действительных констант в экспоненциальной форме:

$0,2360027$; -10^{-4} ; 16427 ; -0 ; $+0$;
 -1 ; $-24,137$; $1/3$; $867342,17 \cdot 10^9$;
 $(-0,5)^2 \cdot 10^{-7}$; $0,26 \cdot 10^8$; -00556 ; $2155 \cdot 10^{-6}$;
 $-1/3$; $(4)^{1/2}$; $-39,643$; 1 ; -0 ; $+0$.

7. Указать, почему приведенные ниже записи не являются действительными константами:

$-E -04$; $0,2775$; $47.8 (E \emptyset 1)$; $+E \emptyset 3$;
 $-1.E(-\emptyset 4)$; $26.3E 118$; $3,65533899,45904555$;
 $-\emptyset,00.E \emptyset 2$; $4.3E 2.6$; $(-\emptyset,02)E 34$;
 $76,3E 5$; $4E - \emptyset 000$; $-\emptyset,0E$; $4627,01$;
 $37,01E$; $\emptyset 5701$, $(\emptyset.5 E 1) E \emptyset 5$;
 $-\emptyset,0457E 17$; \emptyset . $(-E)$.

8. Указать, какие из приведенных ниже действительных констант без экспонент определяют одно и то же число:

$893.$; $-0.$; 276.3 ; $-\emptyset,043$; $-\emptyset,0$; 3.60 ;
 $+893.$; 0276.3 ; 00.000 ; 0893.0 ; $0,0$;
 $+0000276,3000$; $0000893.$; $+3.6$; $-0,000000$;
 $+00,043$; $+276,300$; $893,000$; $-0,043000$; $+0$.

9. Указать, какие из приведенных ниже действительных констант в экспоненциальной форме определяют одно и то же число:

$-0.3E6$; $1,4768296 E-02$; $-5.E 12$; $-0,3 E 7$;
 $-0,50,0 E 11$; $-0,0030 E +8$; $-300000.E0$;
 $04768296 E 03$; $-0,005 E 15$; $-3.E 5$; $50000 E 8$
 $-300,0E 3$.

10. Указать, какие из приведенных ниже действительных констант определяют одно и то же число:

$-.46023 E-2$; $+074,6500$; $,0074605 E 4$;
 $-406023 E-6$; $+0,007465 E 4$; $0,156015 E 6$;
 $-0,0046023$; $23,1$; $. 7665 E +02$;
 $-4,06023 E -1$; $15,6015 E +04$; $-0,46,02300 E -4$;
 $156,015$; $7,465 E +1$; $74,605$; $,0046023$;
 $156015.$; $74650.E -0,3$; $7,4605 E1$.

11. Представить приведенные ниже числа в виде действительных констант без экспонент:
 10^{-4} ; $-26,3724645$; $1/3$; $-291,3 \cdot 10^{-5}$;
 $+9,99049$; $3,14 \cdot 10^3$; $-3/24$; $2^{+6}/3^2$;
 $1,64 \cdot 10^{-7}$; $-0,64 \cdot 10^5$; $84,67 \cdot 10^{-3}$; $+2$;
 $-0,0,6+0,1 \cdot 10^{-2}$; $0,2 \cdot 10^6$; $-1,1 \cdot 10^{-3}$.

12. Представить приведенные ниже числа в виде действительных констант с экспонентами:
 $0,0000000127$; $-176,2300100$;
 10^{+6} ; $+0,64 \cdot 10^3$; $2/3$; $-3/24$; $127/31$;
 10^{-11} ; $-0,1 \cdot 10^{25}$; $11/3$; $999,99 \cdot 10^6$;
 $-3,10^{-7}$; $0,6 \cdot 10^2 + 36,7 \cdot 10^{-3}$; $+116,0 \cdot 10^{-12}$;
 $-0,0000101 \cdot 10^{57}$; $0,000 \cdot 10^4$; -0 ;
 $24,6$; $17,00085$; $+0,0 \cdot 10^{-6}$.

13. Представить приведенные ниже числа в виде:

а) действительных констант без экспонент;

б) действительных констант в экспоненциальной форме;

$0,6 \cdot 10^{-3}$; $-147 \cdot 10^{-3}$; $+0,00004 \cdot 10^{10}$; $1/3$;
 $1/3 \cdot 10^{-4}$; $\sin \pi / 6$; $-0,4$; $9,9 \cdot 10^{-10}$;
 $0,34$; $0,000000007$; -36 ; $25/4 \cdot 10^{-3}$; -6 ;
 $+6$; 0 ; -0 ; $0,0$; $60600,36$; 1 .

3. Запись переменных в программе

Переменной называется величина, которая может принимать различные значения и к которой обращаются, используя ее имя (наименование, идентификатор переменной). Фактически в машине за каждой переменной закрепляется часть памяти, где хранится ее текущее значение.

В языке БЕЙСИК различаются простые переменные и массивы (индексированные переменные).

Простые переменные обозначаются буквой латинского алфавита или буквой и цифрой. Например, A, A1, A9, A0, ..., Z9. Таким образом, всего простые переменные имеют 286 различных имен. Массивы могут быть одномерными (векторами) и двумерными (матрицами).

Они также обозначаются буквой латинского алфавита или буквой и цифрой, но далее в круглых скобках обязательно определяются значения индексов либо целыми числами, либо простыми переменными. В случае двумерного массива значения индексов в круглых скобках разделяются запятой. Например $A(1(25), X(1), X(2), X(3))$, $A(5,5)$. Одновременно в задаче может использоваться не более 208 различных имен (идентификаторов).

Если переменная может принимать только целые значения, то она называется целой. Признаком целой переменной является знак % стоящий после имени. Например $X\%(1)$, $A\%$. Попытка присваивать целой переменной дробное значение приведет к присвоению целой части числа, если таковая по модулю меньше 7999, в противном случае ЭВМ выдает сообщение об ошибке.

4. Арифметические действия и арифметические выражения

К арифметическим действиям относятся следующие операции, которые обозначаются в языке БЕЙСИК соответствующим знаком:

- + — сложение;
- — вычитание;
- * — умножение;
- / — деление;
- ^ — возведение в степень.

Арифметическим выражением будем называть совокупность чисел, переменных, функций, соединенных по определенным правилам знаками арифметических операций и круглыми скобками, или одно число, переменную или функцию.

Арифметическое выражение всегда задает правило вычисления одного числа. Следует иметь в виду, что до вычисления значения арифметического выражения каждой переменной, входящей в него, должно быть присвоено некоторое значение.

Выражения на языке БЕЙСИК записываются в одну строчку. Ниже приведены примеры арифметических выражений, записанных на языке БЕЙСИК, и соответствующие им формулы алгебры.

Выражение на языке БЕЙСИК	Выражение в обычных обозначениях
5	5
4/3 * 7 ^ 2	4/3 · 7 ²
A/B+C*D ^ 10	$\alpha/\beta + c\alpha^{10}$

5. Функции

В выражениях могут использоваться функции. Все функции могут быть разделены на те, правила вычисления которых имеются в памяти ЭВМ, — такие функции будем называть стандартными, и те, которые определяет программист. О функциях второго типа рассказывается в следующем разделе.

Стандартные функции обозначаются идентификаторами, состоящими из трех или шести букв. За идентификатором в скобках пишется аргумент. Аргументом может быть арифметическое выражение при условии, что значение функции при данном значении аргумента определено. В языке БЕЙСИК имеются следующие стандартные функции.

Запись на языке	Название и обозначение функции	Пример
<i>SIN(X)</i>	синус x , $\sin x$	<i>SIN(3.14)</i>
<i>COS(X)</i>	косинус x , $\cos x$	<i>COS(A/B)</i>
<i>TAN(X)</i>	тангенс x , $\operatorname{tg} x$	<i>TAN(A+B)</i>
<i>ARC SIN(X)</i>	арксинус x , $\operatorname{arcsin} x$	<i>ARC SIN(0)</i>
<i>ARCCOS(X)</i>	арккосинус x , $\operatorname{arccos} x$	<i>ARCCOS(0.5)</i>
<i>ARCTAN(X)</i>	арктангенс x , $\operatorname{arctg} x$	<i>ARCTAN(1.)</i>

Запись на языке	Название и обозначение функции	Пример
$SQR(x)$	квадратный корень, \sqrt{x}	$SQR(4)$
$SGN(x)$	сигнатура x , $Sgn\ x$ $Sgn\ x = \begin{cases} 0, & x = 0 \\ 1, & x > 0 \\ -1, & x < 0 \end{cases}$	$SGN(x)$
$LOG(x)$	натуральный логарифм, $\ln x$	$LOG(A\Lambda 2)$
$EXP(x)$	экспонента, e^x	$EXP(-x\Lambda 2)$
$ABS(x)$	модуль, $ x $	$ABS(A-B)$
$INT(x)$	целая часть, $[x]$ (ближайшее целое $Kx, \leq x$)	$INT(3.6)$

Примечание. Имеются еще функции, которые в данном пособии не рассматриваются.

Нажатие клавиши π (или # PI) вызывает значение числа $\approx 3,14159265359$.

Ввод функции может быть осуществлен в следующей последовательности: наименование функции, включая открывающую скобку, аргумент, закрывающая скобка. Ввод наименования обратной тригонометрической функции осуществляется нажатием клавиши "ARC", - клавиши наименования тригонометрической функции, включая, открывающую скобку ввода аргумента, скобки.

Примеры арифметических выражений со стандартными функциями:

$$\begin{array}{ll}
 SIN(SQR(A\Lambda 2) + B\Lambda 2) & \sin(\sqrt{a^2 + b^2}) \\
 ARCTAN(A\Lambda 4 + SQR(B)) & \arctg(a^4 + \sqrt{b}) \\
 A2 * INT(M+N) + 3\Lambda(-1) + & a_2 [M+N] + \frac{1}{3} + 2\arctg 1 \\
 + 2 * ARCTAN(1. \emptyset) &
 \end{array}$$

Арифметические операции выполняются только над цифровыми выражениями, т.е. над константами и переменными.

Машина выполняет операции слева направо в соответствии со старшинством операции:

- 1) вычисление значений функций;
- 2) возведение в степень (\wedge);
- 3) деление и умножение ($/$, \times);
- 4) сложение и вычитание ($+$, $-$).

Таким образом, сначала выполняются все вычисления значений функций, затем все возведения в степень, затем все деления и умножения, затем все сложения и вычитания. Для изменения этого порядка действий применяют скобки. Выражение в скобках вычисляется первым. При применении скобок надо следить за тем, чтобы количество открывающих скобок было равно количеству закрывающих. При записи арифметических выражений надо следить за тем, чтобы друг за другом не стояли два знака действий. В случае нарушения этого правила выдается сигнал об ошибке. Скобочная запись используется и при выполнении действий с отрицательными числами и для изменения знака переменной.

Пример. 5^{-3} надо записать так: $5 \wedge (-3)$. Запись $5 \wedge -3$ неверна, так как два знака арифметических действий стоят рядом.

Скобочная запись может использоваться несколько раз: $(((7,3 + 4,2) \wedge 2 + 6) \wedge .5 + 17) / 22$, при этом сначала вычисляется выражение во внутренних скобках. Правила действия со скобками полностью аналогичны правилам обычной алгебры. Следует следить за тем, чтобы при использовании операции умножения не пропускался знак \times . Например, запись $(Y + X) \wedge (A + B)$ неверно, надо

$$(Y \times X) * (A + B).$$

Примеры.

Выражение на языке БЕЙСИК	Выражение в обычных обозначениях
$A * X \wedge N - B$	$A \cdot X^N - B$
$(A * X) \wedge N - B$	$(AX)^N - B$
$A * (X \wedge N - B)$	$A \cdot (X^N - B)$
$((X+Y) \wedge 2 + Z \wedge 3 / 7) \wedge 8$	$((X+Y)^2 + Z^3 / 7)^8$

Записи вида $ALBLC$ не допускаются. Вообще, когда возможна какая-либо неоднозначность в выполнении действий, лучше поставить скобки, например X^{Y^Z} надо записать в виде: $X \wedge (Y \wedge Z)$.

Упражнение 2

Записать на языке БЕЙСИК следующие выражения (строчные латинские буквы надо заменять прописными):

- 1) $\frac{\alpha + \beta}{2\alpha - \beta} \cdot (\alpha + c)(\alpha^2 + \beta^3)$;
- 2) $1 + x + \frac{x^2}{2!} + \frac{x^3}{3!}$;
- 3) $((x + 3\alpha + y) / 2x)^4 - \frac{\sqrt{x}}{\sqrt{x - 3\alpha + y \frac{2}{3}}}$;
- 4) $(\alpha - \beta) : (c + \frac{a}{c + \beta / (e + \frac{\beta}{c - B})})$;
- 5) $x^4 \cdot (1 + \frac{x - \frac{\alpha}{x}}{x + \frac{\alpha}{x}})$;
- 6) $8,36 \cdot 10^2 + (\frac{\alpha}{a - \beta} - 1)(\frac{\alpha}{a + \beta} + 2,306 a^e)$;
- 7) $(((8,907x + 1)x + 7,01) - 2,06)x + 2$;
- 8) $78,06x^5 + 4,1 \frac{x^4}{-36,04x^3 - \frac{0,98x^2}{15,1x - 10x^e}}$;
- 9) $x(y(z + a) / \beta)^4 x$;
- 10) $(x^y / y^x)^z / z^y / x$;

- 11) $a + \frac{\beta x}{7,3 - 3,2^{\beta} x^4} - \frac{\beta^2 x - x^3 + \beta}{\beta y^{x+3}} + x^4(\beta - 1);$
- 12) $\left\{ \frac{x+1}{x^2+2} + 3,7 \left[\frac{x^3+10x-1}{x+1} \left(\frac{x}{x-1} + 2 \right) \right]^3 + \frac{x}{x^2-25} \right\}^4 - 2;$
- 13) $3,14z^2 + v^2 \cdot h + \frac{v \cdot h + 1}{v} + 3,14v^3;$
- 14) $\left(-a(x+y) - 1 + \frac{3,089x^4 - 2}{1 + \frac{x-1}{x+1}} \right)^2.$

Упражнение 3

Записать на языке БЕЙСИК следующие формулы:

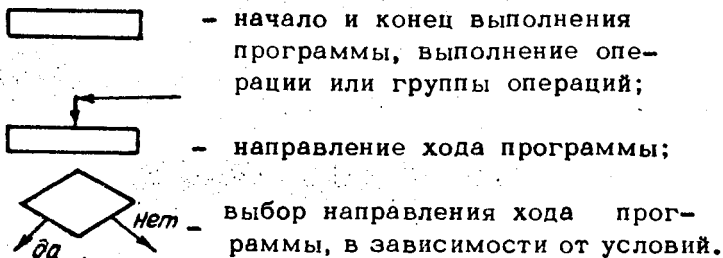
- 1) $100^{\frac{1}{2} \ln x - \ln 2};$
- 2) $e^{\sin^3 y} + \ln(\operatorname{arctg} y);$
- 3) $\operatorname{arctg} \frac{x + \sin y}{1 - x \cos y};$
- 4) $t^3 - e^{t^2/2} + 2,5 \ln(1+t);$
- 5) $\sqrt[5]{\frac{25 + \sqrt[3]{136}}{0,00034} \cdot x};$
- 6) $x \cdot \sin \left[\frac{1}{2} \operatorname{arctg} \left(-\frac{2\sqrt{2}}{5} \right) \right] + n \cdot \operatorname{ctg} \left[\frac{1}{2} \arccos \left(-\frac{4}{7} \right) \right];$
- 7) $\operatorname{ctg}(x^2 + y^2) + |x - \arccos(x + y^2) - e^{x^3}|;$
- 8) $x \cdot \operatorname{tg} \left(5 \operatorname{arc} \cdot \operatorname{tg} \frac{\sqrt{3}}{3} - \frac{1}{4} \arcsin \frac{\sqrt{3}}{2} \right);$
- 9) $99^{\frac{x}{y} - \ln|x|};$
- 10) $x/a - \frac{a}{p} \ln(a + \beta e^{px});$
- 11) $\sqrt[3]{x \cdot 4,2013 \cdot \sqrt[3]{0,1} - 1};$
- 12) $(x + y^2 - 1)(x + y)^4 \sin \frac{x}{y};$
- 13) $\operatorname{arcctg} \frac{x - y}{1 + x \cdot \sin y}.$

6. Структура программы, Режим ввода программы,
Счетный режим, Режим вывода

Программа записанная на языке БЕЙСИК имеет строчную структуру. Каждая строка начинается с номера, который является целым числом от 0 до 9999 включительно. Строка состоит из номера и следующих за ним предложений-операторов. Операторы представляют собой последовательность символов и отделяются в строке друг от друга двоеточием. Текст строки набирается с помощью клавиатуры последовательно, причем на экране отображается набираемый текст, и курсор указывает первое свободное место. Каждая строка заканчивается нажатием клавиши *CR/LE* для передачи текста в ЭВМ, при этом курсор переходит в первую позицию очередной строки.

Очередность выполнения строк в программе определяется их номерами, если нет оператора перехода к другим строкам или обращения к определенной строке. В связи с тем, что в процессе отладки программы может появиться необходимость вставить новые строки, рекомендуется проводить нумерацию строк с определенным шагом, например, 10, 20 ...

Перед написанием программы полезно составить алгоритм решения задачи, т.е. описать последовательность действий, которые должны быть выполнены при счете. Алгоритм можно записать в виде блок-схемы, которая строится с помощью геометрических фигур, внутри которых описаны выполняемые ЭВМ действия. Стрелки указывают последовательность выполнения действий. При составлении блок-схемы пользуются следующими обозначениями:



Если в процессе набора текста строки допущена ошибка, но не нажата клавиша *CR/LF*, то ошибку можно исправить, используя клавиши редактирования при вводе, расположенные во второй зоне клавиатуры:

BACK SPACE - стирание последнего из имеющихся символа;

LINE ERASE - стирание всей строки.

Перед вводом программы необходимо последовательно нажать клавиши: *RESET*; *CLEAR* (очистка памяти), *CR/LF*, дождаться ответа системы и набрать *RUN 1, CR/LF*. После ответа *READY* можно набирать текст программы.

После набора текста программы задать счетный режим нажатием клавиш *RUN, CR/LF*. Можно вывести из памяти текст программы, нажатием клавиш *LIST, CR/LF*.

7. Операторы ввода данных: *INPUT, READ* и оператор блока данных *DATA*

О п е р а т о р *INPUT* позволяет вводить данные во время выполнения программы и присваивать их переменным списка оператора. Вид оператора:

INPUT < список переменных >

Переменные отделяются друг от друга запятыми.

При исполнении оператора *INPUT* выполнение программы прекращается, а на экране дисплея появляется знак вопроса "?". Пользователь набирает на клавиатуре данные, соответствующие переменным в операторе *INPUT* и нажимает клавишу *CR/LF*.

В случае нескольких элементов списка оператора *INPUT* запрашиваются все новые данные вплоть до исчерпывания списка. В случае ошибочного ввода данных печатается сообщение об ошибке и данное запрашивается вновь.

П р и м е р. 250 *INPUT A,B,C CR/LF*

На пульте набираем 0.1, 0.2, 0.8

В результате *A = 0.1; B = 0.2; C = . . .*

Задание констант в программе может осуществляться с помощью оператора *INPUT* и заданием блока данных. Блок данных — это список констант (чисел), которые можно извлекать последовательно с помощью оператора *READ* и присваивать переменным.

Оператор *DATA* позволяет определить блок данных. Все данные, определенные одним или несколькими операторами *DATA*, воспринимаются как части одного и того же блока *DATA*, при этом они располагаются в порядке их определения.

Вид оператора:

DATA <1-е число> , <2-е число> , ...

Операторы *DATA* обычно пишут в начале программы.

Оператор *READ* имеет вид

READ <переменные, разделенные запятыми>

Оператор *READ* предназначен для выборки констант, заданных в программе операторами *DATA*.

Каждый элемент из списка оператора *DATA* присваивается очередной переменной оператора *READ*. Это продолжается до тех пор, пока всем переменным оператора *READ* не будут присвоены значения из *DATA* или до тех пор, пока все элементы оператора *DATA* не будут использованы по одному разу.

Если оператор *READ* содержит больше переменных, чем имеется значений в *DATA*, то реализуется следующий оператор *DATA*, а в случае его отсутствия выдается сообщение об ошибке.

Если оператор *READ* содержит меньше переменных, чем их есть в *DATA*, то следующий оператор *READ* начинает присвоение с первого неиспользованного значения в операторе *DATA*.

Пример. 10 *DATA* 5,1,8,13,1,14

20 *READ* A, B, C

После выполнения оператора *READ* :

A = 5,1; B = 8,13; C = 1,14.

8. Оператор печати PRINT

Оператор печати *PRINT* служит для вывода информации на экран дисплея или цифроречать. Общий вид оператора *PRINT* :

PRINT < список объектов вывода >

В зависимости от списка объектов оператор печати делится на три типа:

а) вывод результатов вычислений. Элементами списка в этом случае являются имена переменных с индексами или без них, значения которых требуется вывести;

б) вывод сообщения (текста). Элементами списка являются сообщения заключенные в кавычки. В тексте могут быть русские буквы;

в) смешанный тип, объединяющий типы печати а и б.

Элементы списка отделяются друг от друга запятой или точкой с запятой, которые управляют форматом печати. Строка печати делится на пять зон по 16 символов в каждой. Если элементы отделены запятой, то печать очередного элемента начинается с первого места ближайшей свободной зоны (зонная печать) или с новой строки, если таковой нет. Если же точкой с запятой, то с ближайшего свободного места (компактная печать).

Каждое обращение к оператору *PRINT* означает переход на новую строку, если в предыдущем исполненном операторе печати последним символом не являлись ", " "; ". В противном случае печать будет зонной или компактной соответственно. Для пропуска строки можно употребить пустой оператор *PRINT*, т.е. не имеющий объектов вывода.

Примеры.

PRINT A, B, C

PRINT D, E

PRINT A; B; C;

PRINT D; E

1 зона	11	111	1	12	13	4	6
1	12	13					
4	6						
20							

9. Арифметический оператор присваивания LET

Арифметический оператор присваивания имеет вид

$$LETA = \langle a.b \rangle ,$$

где $a.b$ - арифметическое выражение.

В результате выполнения оператора LET вычисляется значение арифметического выражения, стоящего в правой части равенства, и присваивается переменной, стоящей слева. Внешне оператор присваивания напоминает равенство, однако, это не равенство. Нельзя поменять местами правую и левую части, так как они неравноправны: вычисляется всегда значение правой части. Кроме того, возможен, например оператор $LETA = A + 1$, по которому переменной A присваивается новое значение, равное старому, увеличенному на единицу. С точки зрения равенства запись $A = A + 1$ лишена смысла.

З а м е ч а н и е. Слово LET в операторе присваивания для ЭВМ "Искра 226" можно опустить.

П р и м е р ы. 180 $LET X = 18,5$, в результате выполнения этого оператора переменной x будет присвоено значение 18,5.

250 $LETA = SIN (B - 7,8)$ - вычисляется значение $\sin (B - 7,8)$, где значение B известно и результат присваивается переменной A .

380 $LETA = A * E - Z * Y$ - вычисляется значение $a * e - z * y$, где значения a, e, z, y известны, и результат присваивается переменной x .

400 $LET X = Y$. Известное значение переменной Y присваивается также переменной X .

10. Операторы STOP и END

Оператор $STOP$ осуществляет программированный останов счета по программе. Оператор $STOP$ используют при необходимости произвести какие-то действия в ходе выполнения программы, например сменить кассету с MA , оценить результаты и т.д. Для возобновления выполнения программы надо нажать клавишу $CONTINUE$.

Оператор *END* является оператором конца программы. По нему ЭВМ заканчивает выполнение данной программы и переходит в режим ожидания следующего задания. Перед оператором *END* рекомендуется употреблять оператор *STOP*. Оператор *END* в программе должен являться последним.

11. Оператор *SELECT*

Для подключения того или иного устройства вывода служит оператор *SELECT*. В начале работы ЭВМ для вывода информации всегда подключен дисплей (БОСГИ). Подключение цифropечати для вывода результатов осуществляется оператором *SELECT PRINT DC*, который должен стоять в программе. Обратное подключение БОСГИ осуществляется оператором

SELECT PRINT DS.

Для подключения цифropечати с целью вывода программы из памяти ЭВМ необходимо набрать оператор *SELECT LIST DC*, а для обратного подключения БОСГИ -

SELECT LIST DS.

Примечание. Оператор *SELECT* используется и в некоторых других случаях, изучение которых не входит в задачу данного практикума.

Лабораторная работа 1 ВЫЧИСЛЕНИЕ АРИФМЕТИЧЕСКИХ ВЫРАЖЕНИЙ

С помощью оператора *INPUT* или *READ* ввести данные и вычислить значение арифметического выражения.

- Отпечатать: 1. Фамилию, номер группы студента, выполняющего работу.
2. Заголовок: "Лабораторная работа 1".
3. Результат в виде $X = \langle \text{результат} \rangle$.

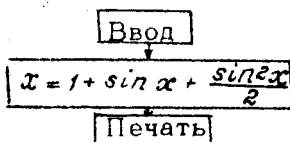
Варианты арифметических выражений представлены в упражнении 3. Значения переменных, входящих в выражения, принять следующими:

- | | | |
|--------------------------|-------------------|--------------|
| 1) $x = 9;$ | 7) $x = 0,01;$ | $y = -0,8;$ |
| 2) $y = 1,1;$ | 8) $z = 2;$ | |
| 3) $x = 0,5; y = \pi/4;$ | 9) $x = -0,25;$ | $y = 2;$ |
| 4) $t = 0,25;$ | 10) $x = -1,456;$ | $p = 2,1;$ |
| 5) $x = 2;$ | $a = 3;$ | $b = 0,1;$ |
| 6) $z = 3; u = 5,1;$ | 11) $x = 3,1415;$ | |
| | 12) $x = 2;$ | $y = 1,5;$ |
| | 13) $x = 0,1;$ | $y = -0,01.$ |

Пример. Вычислить значение.

$$1 + \sin x + \frac{\sin^2 x}{2},$$

где $x = 3,1415$, присвоить результат переменной x и отпечатать. Составляем блок-схему программы.



Программа.

10 INPUT x	<input checked="" type="checkbox"/>	CR/LF	или	10 DATA 0.1	<input checked="" type="checkbox"/>	CR/LF
20 LET X=1+SIN(X)+SIN(X)^2/2	<input checked="" type="checkbox"/>			20 READ X	<input checked="" type="checkbox"/>	
30 PRINT "СИДОРОВ_С-113"	<input checked="" type="checkbox"/>			30 LET X=1+SIN(X)+SIN(X)^2/2	<input checked="" type="checkbox"/>	
40 PRINT "Л.Б. ПАВ. N1"	<input checked="" type="checkbox"/>			40 PRINT "СИДОРОВ_С-113"	<input checked="" type="checkbox"/>	
50 PRINT "X="; X	<input checked="" type="checkbox"/>			50 PRINT "Л.Б. ПАВ. N1"	<input checked="" type="checkbox"/>	
60 STOP	<input checked="" type="checkbox"/>			60 PRINT "X="; X	<input checked="" type="checkbox"/>	
70 END	<input checked="" type="checkbox"/>			70 STOP	<input checked="" type="checkbox"/>	
				80 END	<input checked="" type="checkbox"/>	

Первый вариант программы использует оператор **INPUT** , а второй - **READ** .

Набираем программу на клавиатуре, для исправления ошибок, используя клавиши **LINE ERASE** и **BACKSPACE** . После того как программа набрана, набираем оператор **RUN CR/LF** для перевода ЭВМ в счетный режим работы. На экране БОСГИ получаем ответ.

СИДОРОВ С 113

ЛАБОРАТОРНАЯ РАБОТА 1

X = ...

Если ответ получен верный, выводим текст программы и результат на цифropечать. Для этого набираем:

SELECT LIST **OC** **CR/LF** , подключая цифropечать для вывода программы;

LIST **CR/LF** , задавая режим вывода содержимого памяти, при этом на бумаге цифropечати будет отпечатана программа, находящаяся в памяти;

IS SELECT PRINT **OC** **CR/LF** , добавляя оператор в программу для подключения цифropечати для вывода результатов;

RUN **CR/LF** , переводя ЭВМ повторно в счетный режим.

После выполнения программы на бумаге цифropечати будет отпечатан ответ.

12. Редактирование программы

После ввода программы пользователь может просмотреть ее с начала до конца с помощью оператора

LIST . При вводе новой строки, следует ввести номер этой строки, новый текст и нажать клавишу *CR/LF* . Новую строку в программе вставляют с номером, находящимся между уже имеющимися номерами соседних строк. Например, для программы

```
10 DIM C(47)
100 FOR A=1 TO 47
110 B=C(A)+A*3
120 NEXT A
```

в которой пропущена печать результатов, следует ввести оператор *PRINT* в строку, имеющую номер между 110 и 120; например 115 *PRINT* в *CR/LF* новая строка автоматически будет вставлена между строками 110 и 120.

Для редактирования программной строки, введенной в память, необходимо перевести ЭВМ в режим редактирования. Надо нажать на клавиатуре клавишу *EDIT CR/LF* , при этом начинают работать клавиши перемещения курсора, набрать номер строки и нажать клавишу *EDIT* . После этого на экране около номера строки слева появляется звездочка. Клавишей *RECALL* производится вызов на экран программной строки для последующего редактирования.

Пр и м е р. Требуется изменить строку 40 *PRINT A, B, C*, находящуюся в памяти машины. Нажмите клавиши *EDIT CR/LF 40 EDIT RECALL CR/LF* (см. выше). После этого на экране появится текст 40 строки * 40 *PRINT A, B, C* .

Строку редактируют следующими клавишами:

DELETE - стирает символ на местоположении курсора, а часть строки справа от курсора сдвигается влево на один символ;

ERASE - стирает часть строки, начиная с местоположения курсора вправо;

INSERT - раздвигают строку для вставки символа. Часть строки, начиная с курсора сдвигается вправо.

Шесть клавиш управляют положением курсора. Клавишами \rightarrow (\leftarrow) смещают курсор вправо (влево) по строке программы на одно знакоместо, клавишами \downarrow (\uparrow) курсор смещают на строку вниз (вверх). Клавишами $\leftarrow \dots \rightarrow$ смещают курсор вправо или влево на пять знакомест. Клавиши редактирования и управления курсором действуют в пределах редактируемой строки.

Пр и м е р. Изменить строку *PRINT A, B, C* на *PRINT C*.

Клавишей $\leftarrow \rightarrow \uparrow \downarrow$ установите курсор под буквой *A* и нажмите клавишу *DELETE*, при этом весь текст, следующий за стираемым символом, смещается влево на одно знакоместо, курсор не перемещается. При повторном нажатии указанной клавиши стирание символа повторяется. Ниже представлены примеры текста до и после четырехкратного нажатия клавиши *DELETE*:

```
READY
:PRINT A,B,C
```

```
READY
*PRINT A,B,C
```

```
READY
*PRINT A,B,C
```

```
READY
*PRINT ,B,C
```

```
READY
*PRINT C
```

Участок текста до и после нажатия клавиши *ERASE*.

```
READY
:PRINT "ПРИМЕР"
```

```
READY
*PRINT
```

Участок текста до и после нажатия клавиши *INSERT*.

```
READY
:PRINT TAN(S)+4.12*4
```

```
READY
*PRINT TAN(S)+4.12*4
```

```
READY
*PRINT TAN(S)+- 4.12*4
```

13. Операторы цикла FOR-TO и NEXT

Во многих случаях возникает необходимость неоднократно выполнить оператор или группу операторов. В этом случае организуют так называемый цикл. Для на-

писания циклов в языке БЕЙСИК имеются два оператора: оператор начала цикла *FOR* и оператор конца цикла *NEXT*. Между ними должны быть заключены операторы (или оператор), которые необходимо неоднократно повторить - так называемое "тело цикла". Для того чтобы контролировать число повторений, в состав операторов начала и конца цикла вводится простая переменная - параметр цикла, значение которой при каждом повторении меняется.

Оператор начала цикла имеет вид

FOR I = A1 TO A2 STEP A3,

где *I* - параметр цикла, *A1* - выражение, задающее начальное значение параметра цикла, *A2* - выражение, задающее конечное значение параметра цикла, *A3* - шаг цикла. Оператор конца цикла имеет вид

NEXT I

Пр и м е р. 120 *FOR B = 0 TO 51 STEP 3*

[тело цикла]

280 NEXT B

Выполнение цикла происходит следующим образом:

1) при исполнении оператора *FOR* вычисляются начальное и конечное значения переменной цикла и значение шага;

2) исполняются операторы, заключенные между операторами *FOR* и *NEXT* - тело цикла;

3) при исполнении оператора *NEXT* переменная цикла изменяется на величину шага цикла (в нашем примере на 3) и полученное новое значение сравнивается с конечным значением;

4) если значение переменной больше конечного значения, при шаге большем нуля, то цикл заканчивается и выполнение программы продолжается со следующего за *NEXT* оператора, в противном случае повторяется исполнение тела цикла;

5) последняя часть оператора *FOR STEP A3* может отсутствовать; в этом случае шаг считается равным единице.

Пример. 3ϕ FOR $I = \phi$ TO 1ϕ .

Тело цикла всегда выполняется по крайней мере один раз, даже если конечное значение переменной меньше начального при положительном шаге. Значение шага цикла может быть отрицательным. Тогда выход из цикла определяется условием $I < A2$.

Циклы могут быть вложены друг в друга, но при этом внутренний цикл должен полностью находиться внутри тела внешнего. Войти в тело цикла можно только через оператор FOR, выход из цикла до его окончания возможен.

Примеры.

1) 10 FOR A=1 TO 10

20 FOR B=5 TO 50 STEP 10

30 LET X=B/A

40 PRINT A, B, X

50 NEXT B

60 NEXT A

2) 50 GOTO 70 запрещенный вход в цикл

60 FOR Y=1 TO 10 STEP 2

70 LET Z(Y)=FNA(Y)-LOG(Y)

80 NEXT Y

100 FOR I=1 TO 4

110 FOR K=1 TO 6

120 IF Z(K)>10 THEN 160 разрешенный выход из цикла

150 NEXT K

160 NEXT I

3) FOR B=SQR(2)+2 TO SQRT(100)+2.

Цикл реализуется 9 раз, B принимает значения 3.414, 4.414, ..., 11.414;

4) FOR B=1φ TO-1φ STEP-5

C принимает последовательно значения 10, 5, 0, -5, -10;

5) Записать программу для вычисления суммы

$$S = \sum_{n=1}^{11} \frac{1}{n},$$

10 LET S=φ

20 FOR I%=1 TO 11

30 LET S=S+1/I%

40 NEXT I%

50 PRINT S

14. Определение функции пользователя, оператор DEFFN

БЕЙСИК дает возможность пользователю задавать в своей программе по своему желанию такие функции, которые могут использоваться аналогично стандартным функциям БЕЙСИК. Такие функции могут быть полезными, например в том случае, если в программе необходимо вычислить какое-либо арифметическое выражение несколько раз. Вид оператора:

DEFFN < имя FN > (< простая цифровая переменная >) = < арифметическое выражение > .

Параметр < имя FN > - это латинская буква.

П р и м е р ы .

DEFFN C(X) = SIN(X/2) + TAN(X/2)

DEFFN B(Y) = 3*Y + Y/2

Параметр < простая цифровая переменная > (в примерах X и Y) представляет собой формальный аргумент, при обращении к функции формальный аргумент заменяют фактическим.

Параметр < арифметическое выражение > может включать в себя функции, вызывающие другие функции, но если например функция A вызывает функцию B, то B не может вызвать A.

Обращение к функции можно совершать из любой точки программы, по имени:

$FNC(X)$, $FNB(Y)$ и т.д.

Значение аргумента, обозначенного в функции FN , автоматически передается оператору $DEFFN$ для вычисления, и результат возвращается функции FN . Выполнение оператора $DEFFN$ осуществляется исключительно обращением к функции FN .

Примеры.

1) 10 $DEFFN$ $C(X) = SIN(X/2) + TAN(X/2)$

20 LET $Y = FNC(2)$

В результате выполнения программы Y будет присвоено значение $\sin(2^2) + \operatorname{tg}(2/2) = \sin 4 + \operatorname{tg} 1$.

2) 10 LET $X = 3$

20 $DEFFN$ $A(Z) = Z^2 - Z$

30 $PRINT$ $X + FNA(2 * X)$

Вычисляется: $2 * X = 2 * 3 = 6$, вычисляется $FNA(6) = 30$, печатается 33.

Лабораторная работа 2

ТАБУЛИРОВАНИЕ ФУНКЦИЙ

Дана функция $f(x)$ с областью определения, содержащей отрезок $[a, b]$. Требуется составить таблицу значений этой функции в точках a_0, a_1, \dots, a_n , где $a_0 = a$, $a_n = b$, $a_{i+1} = a_i + h$, $h = \frac{b-a}{n}$, $i = 0, \dots, n-1$.

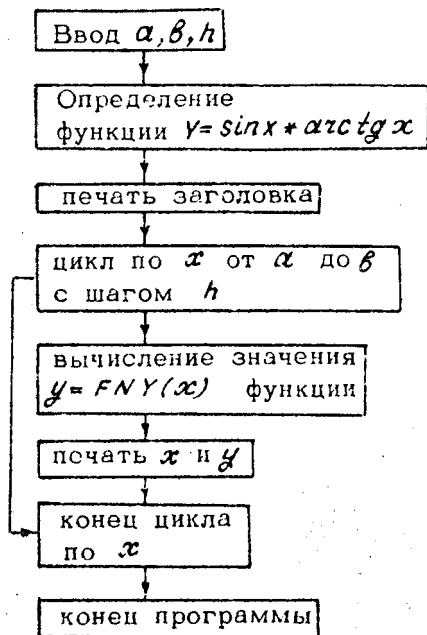
Другими словами надо составить таблицу функции $f(x)$ на отрезке $[a, b]$ с шагом h .

Программу, для решения этой задачи удобно составить с использованием операторов цикла и определения функции.

Пример. Пусть

$f(x) = \sin x + \operatorname{arctg} x$; $x \in [0, 1]$, $h = 0,1$.

Блок-схема программы:



Программа

```
10 INPUT A, B, H [CR/LF]
20 DEFN Y = SIN(X) * ARCTAN(X) [CR/LF]
30 PRINT "Л С И Д О Р О В Ы С - 113" [CR/LF]
40 PRINT "Л Л А Б О Р А Т О Р Н Я Р А Б О Т А Л И Н Ъ Т А Б У Л. Ф У Н К Ц." [CR/LF]
50 FOR X = A TO B STEP H [CR/LF]
60 LET Y = FN Y(X) [CR/LF]
70 PRINT "X = "; X, "Y = "; Y [CR/LF]
80 NEXT X [CR/LF]
90 STOP [CR/LF]
100 END [CR/LF]
```


З а м е ч а н и е. При выполнении этой работы студентам предлагается не вводить свою программу, а получить ее редактированием программы предыдущего товарища, если таковой имелся.

Последовательность работы при вводе программы, запуска ее на счет, исправления ошибок, подключения цифрочести такая же, как и в лабораторной работе.

Оператор печати 70 является оператором смешанного типа, содержащим в качестве объектов и тексты (x , y), и значения переменных (x , y). Точка с запятой между $x = ux$; $y = y$ обеспечивает печать чисел непосредственно после текстов. Так как между x и $y =$ стоит запятая, то текст будет напечатан с первой позиции ближайшей свободной зоны. В результате получим

$$x = \langle \text{число} \rangle \quad y = \langle \text{число} \rangle$$

В связи с тем, что оператор 70 стоит в цикле, он будет выполняться $n + 1$ раз, а так как он не заканчивается ни запятой, ни точкой с запятой, при каждом обращении печати будет начинаться с новой строки. В результате получим таблицу значений аргумента и функции.

Варианты заданий к лабораторной работе 2.

Составить таблицу значений данной функции $y = f(x)$ на данном отрезке $[a, b]$ с данным шагом h .

№ варианта	Функция $y = f(x)$	$[a, b]$	h	Ответ $y(b)$
1	$y = (\arctg x) / x$	[1,2]	0,2	0,553
2	$y = x \cdot e^{\sqrt{x}}$	[2,4]	0,4	29,556
3	$y = (\ln x) / \sqrt{x}$	[4,7]	0,6	0,735
4	$y = (\sin x) / \sqrt{ x }$	[-2,-1]	0,2	0,841
5	$y = x \cdot \sqrt{\ln x}$	[4,6]	0,4	8,031
6	$y = (\arcsin x) / x$	[0,5;1]	0,1	1,571
7	$y = x^x$	[1;2]	0,2	3,999

№ варианта	Функция $y = f(x)$	$[\alpha, \beta]$	h	Ответ $y(x)$
8	$y = \arccos(1/x)$	[4;4,5]	0,2	1,369
9	$y = \sqrt{1 - e^{\sin x}}$	[3,5;4]	0,1	0,728
10	$y = x / \operatorname{tg} x$	[3,5;4]	0,1	3,455
11	$y = \sqrt{\arctg x}$	[4,5]	0,2	1,172
12	$y = \sqrt[3]{\sin x}$	[0,1]	0,2	0,944
13	$y = \arcsin \sqrt[3]{x}$	[0,1]	0,2	1,571
14	$y = \sqrt[4]{\ln x}$	[10;13]	0,6	1,266

15. Оператор перехода GOTO

Строки выполняются в порядке возрастания их номеров. Иногда программисту необходимо изменить этот порядок. Для этого служит оператор перехода, явно показывающий ту строку, которую необходимо выполнить следующей. Оператор перехода имеет вид

GOTO M,

где M — порядковый номер строки, к выполнению которой следует перейти (**GOTO** пишется вместо). Например результатом выполнения оператора **250 GOTO 140** будет переход к выполнению строки с номером 140.

16. Оператор условного перехода IF

Оператор **IF** служит для того, чтобы при выполнении некоторого условия передать управление определенному оператору. Общий вид оператора **IF** следующий:

IF < условие > THEN n,

где < условие > — два арифметических выражения (a, β) соединены одним из знаков арифметических отношений (= равно, < не равно, > больше,

$>$ — больше или равно, $<$ — меньше, \leq — меньше или равно),

n — номер некоторой строки программы.

Действие оператора **IF** состоит в том, что машина переходит к выполнению строки с номером n , если выполнено указанное условие. В противном случае (если условие не выполнено), машина переходит к очередной строке программы.

Примеры.

```
120 IF A >= B THEN 240
230 IF SIN(X) < 0,5 THEN 100
380 IF X^2 + Y^2 < Z THEN 80
```

Составить часть программы для вычисления

$$x = \begin{cases} 0,5 & \text{при } y < 1 \\ y^2 & \text{при } y \geq 1 \end{cases}$$

```
10 IF Y < 1 THEN 40 CR/LF
20 LET X = Y^2 CR/LF
30 GOTO 50 CR/LF
40 LET X = 0,5 CR/LF
50 PRINT X
```

Последний оператор задает печать x .

Упражнение 4

1. Составить программы для вычисления следующих выражений.

а)
$$b = \begin{cases} -1/2, & \text{если } a < 0; \\ 1/2, & \text{если } a \geq 0; \end{cases}$$

б)
$$a = \begin{cases} x+y, & \text{если } w = z; \\ x-y, & \text{если } w \neq z; \end{cases}$$

в)
$$b = \begin{cases} 17 - 0,48 R^2, & \text{если } R - 120 < 0; \\ 0, & \text{если } R - 120 = 0; \\ 18 / (1 + R^2 / 18), & \text{если } R - 120 > 0; \end{cases}$$

$$г) W = \begin{cases} -1, & \text{если } q < 0 \text{ и } h < 0; \\ 0, & \text{если } q = 0 \text{ и } h = 0; \\ +1 & \text{в остальных случаях;} \end{cases}$$

д) $D = \max(A, B, C)$, где $\max(A, B, C)$ означает наибольшее из чисел A , B и C ;

е) $D = \min(A, B, C)$, где $\min(A, B, C)$ означает наименьшее из чисел A , B , C ;

$$ж) \alpha = 0,73 + \begin{cases} x, & \text{если } x > 0; \\ -1, & \text{если } -1 \leq x \leq 0; \\ x^2, & \text{если } x < -1; \end{cases}$$

$$и) v = \begin{cases} a + b \cdot x + c x^2, & \text{если } a = -1; \\ (a \sin x)^2, & \text{если } a = 0; \\ \sqrt{a + b x}, & \text{если } a = 1; \\ a \ln|x|, & \text{в остальных случаях;} \end{cases}$$

где a - целое число.

2. Составить программу вычисления:

а) расстояния $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ между двумя точками на плоскости;

б) корней системы уравнений

$$\begin{cases} a_1 x + b_1 y = c_1 \\ a_2 x + b_2 y = c_2 \end{cases}$$

по формулам

$$x = \frac{c_1 b_2 - c_2 b_1}{a_1 b_2 - a_2 b_1}; \quad y = \frac{a_1 c_2 - a_2 c_1}{a_1 b_2 - a_2 b_1};$$

в) приближенного значения функции

$$\operatorname{tg} x \approx \frac{x}{1 - \sqrt{1 - x^2}};$$

г) площади треугольника с вершинами в точках

$$P_1(x_1, y_1); \quad P_2(x_2, y_2); \quad P_3(x_3, y_3)$$

по формуле

$$S = \frac{1}{2}((x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1));$$

д) значения величины

$$y = \begin{cases} 16,5x + 9x^2 - 1,25x^3, & \text{если } 1 \leq x < 9; \\ 0, & \text{если } x < 1; \\ a - x, & \text{если } x \geq 9. \end{cases}$$

3. Составить программу для вычисления сумм с использованием оператора цикла

а) $\sum_{n=1}^{13} e^{-i\sqrt{n}}$;

в) $\sum_{n=1}^{12} n^2 e^{-\sqrt{n}}$;

д) $\sum_{n=1}^{17} \sin^3 \frac{1}{n}$;

к) $\sum_{n=1}^{23} 2^n \sin \frac{1}{3^n}$;

н) $\sum_{n=1}^{26} \frac{1}{n(n+3)}$;

б) $\sum_{n=1}^{12} \frac{1}{\sqrt{n(n+1)}}$;

г) $\sum_{n=2}^{15} (n^{\frac{1}{n^2-1}} - 1)$;

е) $\sum_{n=1}^{20} \frac{\cos \frac{2\sqrt{n}}{3}}{2^n}, (\pi \approx 3,1416)$;

з) $\sum_{n=1}^{24} \frac{\sin n}{n \sqrt{n}}$;

к) $\sum_{n=1}^{27} \frac{1}{n(n+1)(n+2)}$.

1. Составить программу для решения квадратного уравнения

$$ax^2 + bx + c = 0.$$

5. Составить программу для вычисления:

а) наименьшего по модулю из чисел A, B, C ;

б) наибольшего отрицательного числа из чисел

A, B, C, D ;

в) наименьшего положительного числа из A, B, C, D .

6. Дана последовательность a_n . Составить программу для вычисления наименьшего номера n , при котором $F(a_n) < \varepsilon$:

а) $a_{n+1} = \frac{1}{2} \left(a_n + \frac{1}{a_n} \right); a_1 = 10; |1 - a_n| < 10^{-4}$;

б) $a_n = \frac{n}{2^n}; a_n < 10^{-5}$;

в) $a_{n+1} = \frac{1}{2} \operatorname{tg} a_n, a_1 = 0,1; |a_n| < 10^{-3}$.

7. Определить, сколько элементов последовательности среди первых N удовлетворяют условию

а) $a_n = \sin n, N = 50, |a_n| < 10^{-2}$;

б) $a_n = \frac{n^2}{2^n}, N = 100, |a_n| > \frac{1}{3}$;

в) $a_n = \frac{\sqrt{n}}{10+n}, N = 100, |a_n| > \frac{1}{10}$.

8. Найти наибольший по модулю элемент последовательности среди первых N элементов и его номер:

а) $\alpha_n = \frac{n^2}{2^n}$, $N = 100$;

б) $\alpha_n = \frac{\sqrt{n}}{100+n}$, $N = 200$;

в) $\alpha_n = \frac{10^n}{n!}$, $N = 100$;

г) $\alpha_n = \sin n$, $N = 100$;

д) $\alpha_n = \sec n$, $N = 200$.

Лабораторная работа 3 ПРИБЛИЖЕННОЕ РЕШЕНИЕ УРАВНЕНИЙ

В практических вычислениях довольно часто приходится решать уравнения вида

$$f(x) = 0. \quad (1)$$

К такому виду может быть приведено любое уравнение вида

$$g_1(x) = g_2(x) \quad (2)$$

перенесением всех членов в левую часть.

Пусть функция $f(x)$ в уравнении (1) определена и непрерывна на некотором конечном или бесконечном интервале. Всякое значение x^* из этого интервала такое, что

$$f(x^*) = 0 \quad (3)$$

называется корнем уравнения (1).

Найти корни уравнения (1) точно удается лишь в частных случаях, поэтому разработаны методы численного приближенного решения уравнений вида (1).

При этом приходится решать две задачи:

1) отделение корней, т.е. отыскание достаточно малых областей, в каждой из которых заключен один и только один корень уравнения;

2) вычисление корней с данной точностью.

При выделении областей, в которых находятся действительные корни уравнения (1), можно воспользоваться тем, что если на концах некоторого отрезка непрерывная функция $f(x)$ принимает значения разных знаков, то на этом отрезке уравнение $f(x) = 0$ имеет хотя бы один корень. Такие отрезки можно найти либо путем построения графика функции, либо с помощью табулирования функции с некоторым шагом (см. задание 1).

Для того чтобы убедиться, что на данном отрезке имеется только один корень, можно исследовать производную функции. Например, если $f(a) < 0$, $f(b) > 0$, $a < b$ и $f'(x) > 0$ при $x \in [a, b]$, то на отрезке $[a, b]$ существует только один корень функции $f(x)$.

Для решения второй задачи существуют многочисленные методы, из которых мы рассмотрим лишь два: метод итераций и метод половинного деления.

Метод итераций

Уравнение (1) представим в виде

$$x = \varphi(x), \quad (4)$$

что всегда можно сделать и притом многими способами.

Предположим, что нам известен отрезок $[\alpha, \beta]$, на котором имеется один корень и

$$|\varphi'(x)| \leq q < 1 \quad (5)$$

при $x \in [\alpha, \beta]$. Тогда решение уравнения (4) можно найти следующим образом. Выберем на отрезке $[\alpha, \beta]$ произвольную точку x_0 - нулевое приближение и примем в качестве следующего приближения $x_1 = \varphi(x_0)$, далее $x_2 = \varphi(x_1)$ и т.д., вообще $x_n = \varphi(x_{n-1})$.

Последовательность $x_0, x_1, \dots, x_{n-1}, x_n$ сходится к корню уравнения (4), т.е.

$$x_n \rightarrow x^* \text{ при } n \rightarrow \infty \text{ при выполнении условия (5).}$$

Если надо найти корень с погрешностью ε , то достаточно взять x_n такое, что $|x^* - x_n| \leq \varepsilon$. На практике вычисления прерываются по условию

$$|x_n - x_{n-1}| \leq \varepsilon. \quad (6)$$

При практическом нахождении корней по методу итераций нужно при переходе от уравнения (1) к уравнению (4) стремиться представить $\varphi(x)$ так, чтобы производная $\varphi'(x)$ по абсолютной величине была меньше единицы. В ряде случаев это удастся с помощью следующего приема. Пусть

$$0 < \mu_1 \leq f'(x) \leq M_1,$$

при $x \in [\alpha, \beta]$. Заменяем уравнение (1) эквивалентным ему:

$$x = x - \lambda \cdot f(x), \quad (7)$$

где $\lambda > 0$.

Подберем λ , например, таким, чтобы выполнялось неравенство:

$$0 \leq \varphi'(x) = 1 - \lambda f'(x) \leq q < 1$$

при $x \in [\alpha, \beta]$. Для этого достаточно взять $\lambda = \frac{1}{M_1}$,

тогда,

$$q = 1 - \frac{\mu_1}{M_1}.$$

Если производная $f'(x)$ отрицательна, то вместо уравнения $f(x) = 0$ рассматриваем уравнение $-f(x) = 0$.

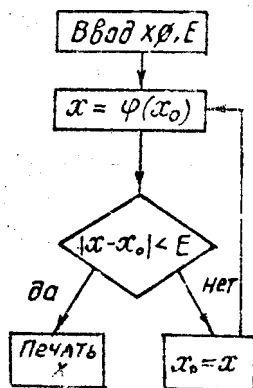
Для написания программы решения уравнения (4) методом итераций надо задать начальное значение x_0 , в программе $x \phi$, погрешность вычислений $\epsilon \rightarrow E$.

Примерный вид программы может быть таким:

```

10 INPUT Xϕ, E
20 DEF FN F(X) = <α.β>
30 LET X = FN F(Xϕ)
40 IF ABS(X - Xϕ) < E THEN 7ϕ
50 LET Xϕ = X
60 GOTO 3ϕ
70 PRINT "СИДОРОВ С-113 ЛАБ. РАБ. №3"
80 PRINT "X = "; X
90 END

```



Здесь x_0 и ε - числа, $\varphi(x)$ - конкретный вид стандартной функции. Предполагается также, что $q \leq 0,5$ и можно пользоваться условием (6) в противном случае можно положить

$$E = \frac{1-q}{q} \cdot \varepsilon.$$

Метод половинного деления

Пусть дано уравнение (1), где функция $f(x)$ непрерывна на $[\alpha, \beta]$ и $f(\alpha) \cdot f(\beta) < 0$.

Для нахождения корня уравнения (1), принадлежащего отрезку $[\alpha, \beta]$, делим отрезок пополам, т.е. выбираем начальное приближение $x_0 = \frac{\alpha + \beta}{2}$. Если $f(x_0) = 0$, то x_0 является корнем уравнения. Если $f(x_0) \neq 0$, то выбираем тот из отрезков $[\alpha, x_0]$ или $[x_0, \beta]$, на концах которого функция $f(x)$ имеет противоположные знаки. Полученный отрезок снова делим пополам и проводим то же рассмотрение и т.д.

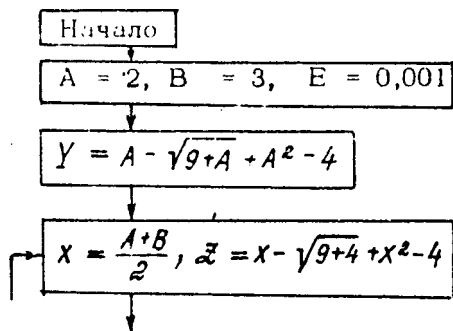
Процесс деления отрезков пополам продолжаем до тех пор, пока длина отрезка, на концах которого функция имеет противоположные знаки, не будет меньше величины ε .

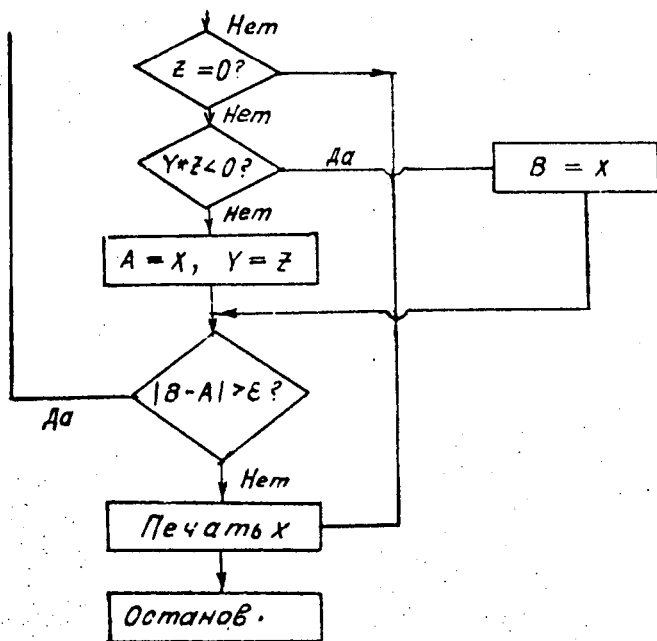
Пример. Методом половинного деления найти корень уравнения

$$x - \sqrt{9+x} + x^2 - 4 = 0$$

на отрезке $[2, 3]$ с погрешностью $\varepsilon = 10^{-3}$.

Блок-схема программы





Программа

```

10 LET A=2; B=3; E=0.001
20 LET Y=A-SQR(9+A)+A*A-4
30 LET X=(A+B)/2
40 LET Z=X-SQR(9+X)+X*X-4
50 IF Z≠0 THEN 110
60 IF Y*Z<0 THEN 90
70 LET A=X; Y=Z
80 GOTO 100
90 LET B=X
100 IF ABS(B-A)>E THEN 30
110 PRINT "X="; X
  
```

Варианты заданий к лабораторной работе 3

1. Решить данное уравнение методом итераций с данной точностью ϵ .

№ варианта	Уравнение	ϵ	Ответ
1	$\ln x = -(1+x)^3$	0,001	0,187
2	$\cos x = 3x-1$	0,01	0,61
3	$2-x = \ln x$	0,01	1,56
4	$x = 2e^{-x}$	0,001	0,853
5	$x^2 = -4 \sin x$	0,01	-1,93
6	$e^{-x} = 1-x^2$	0,01	0,718
7	$e^{x^2} = x+2$	0,01	1,06
8	$\arctg x = 1/x^2$	0,01	1,10
9	$x^2 = \ln(x+1)$	0,001	0,747
10	$e^x = 2-x^2$	0,01	-1,32; 0,54
11	$x^2 = \cos \pi x$	0,001	$\pm 0,438$
12	$x^4 = x+1$	0,001	1,220
13	$x = \operatorname{tg} x \left(-\frac{\pi}{2} < x < \frac{3\pi}{2} \right)$	0,01	4,491
14	$x^3 = 7x+7$	0,001	3,049

2. Найти корень данного уравнения с точностью 0,001 на заданном отрезке $[a, \beta]$ методом деления отрезка пополам.

№ варианта	Уравнение	Ответ
1	$x^4 + 2x^2 - 6x + 2 = \phi$	$[0, 1]$
2	$x^3 - 2x - 2 = \phi$	$[1, 2]$
3	$x^4 + 2x^3 - x - 1 = \phi$	$[0, 1]$

№ варианта	Уравнение	Ответ
4	$x^5 - x - 0,2 = 0$	[1,2]
5	$x^3 - 2x^2 + 3x - 5 = 0$	[1,2]
6	$x^3 - 9x^2 + 20x - 1 = 0$	[0,1]
7	$x^4 + 6x^3 + 12x^2 + 9x - 1 = 0$	[0,1]
8	$x^3 - x^2 + 2x - 3 = 0$	[1,2]
9	$x^4 - 4x + 1 = 0$	[0,1]
10	$x^4 - 2x^3 + 1 = 0$	[1,2]
11	$x^3 + x^2 - 0,5 = 0$	[0,1]
12	$x^4 - 3x^2 + 4x - 3 = 0$	[-3, -2]
13	$x^3 + 2x^2 - 4x + 7 = 0$	[-4, -3]
14	$x^3 - 3x + 1 = 0$	[-2, -1]

17. Оператор размерности массива DIM

Наряду с простыми цифровыми переменными ЭВМ позволяет работать с цифровыми массивами. Массив (индексированная переменная) - совокупность чисел, каждому из которых ставится в соответствие его место в массиве, иначе это упорядоченный набор чисел. В языке БЕЙСИК допускается использование одномерных и двумерных массивов. Для каждого массива в ЭВМ должно быть зарезервировано место в памяти. Объем зависит от количества элементов массива, которое в программе определяется оператором размерности массива *DIM*. Если размерность какого-либо массива оператором *DIM* не объявлена, то ЭВМ рассматривает его как имеющего размерность 10 (для одномерных) и 10x10 (для двумерных). Имя массива обозначает массив целиком, а для того чтобы обратиться к элементу, необ-

ходимо указать индексы в виде простой переменной, либо целого числа. Например, $A(I, J)$, $B(5)$, $C(1, J)$. Наименьшее значение индексов равно единице. Оператор размерности *DIM* имеет вид *DIM* <имя> (<размерность>), <имя> (<размерность, размерность>)...

Здесь имя - имя массива, а размерность - наибольшее значение, которое может принимать соответствующий индекс массива.

П р и м е р ы.

а) *150 DIM A (45)*

резервируется место для одномерного массива, содержащего 45 элементов;

б) *10 DIM B (8, 10)* - резервируется место для двумерного массива из 80 чисел;

в) *10 DIM R (80), L1(3), M (8, 7)*.

Оператор *DIM* должен находиться в программе перед первым обращением к массивам. В операторе *DIM A(X, Y)* или *DIM B(X)*, x и Y - целые числа от 1 до 253. Одним оператором можно зарезервировать в памяти машины место для нескольких массивов. Если массив определен как двумерный, использование его как одномерного и наоборот приводит к ошибке.

В одномерном массиве каждый элемент определяется номером, например, $A(5)$, $B(2)$, $B\%(8)$, в двумерном - номером строки и столбца $C(1, 2)$, $D1(3, 4)$, $K\%(5, 6)$.

Язык БЕЙСИК предусматривает обработку массивов поэлементно - строка за строкой. Иногда может обрабатываться и весь массив сразу. При работе с массивами надо помнить, что индексация начинается с единицы.

В одной программе допускается применение одних и тех же обозначений для простых переменных и элементов массива, так как они независимы, но одномерные и двумерные массивы не должны иметь одинаковых идентификаторов.

П р и м е р. Программа вычисления произведения матрицы A размерности $(M \times N)$ на матрицу B размерности $(N \times Q)$.

```

10 DATA ...
20 READ M, N, Q
30 DIM A(M,N), B(N,Q), C(M,Q)
40 FOR I=1 TO M
50 FOR J=1 TO Q
60 LET C(I, J) = 0
70 FOR K=1 TO N
80 C(I, J) = C(I, J) + A(I, K) * B(K, J)
90 NEXT K
100 NEXT J
110 NEXT I

```

Лабораторная работа 4

СОРТИРОВКА ЧИСЛОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Хорошо известно, что для любой пары a и b действительных чисел можно сказать, что либо $a = b$, либо $a < b$, либо $a > b$.

Пусть $x = \{x_1, x_2, \dots, x_n\}$ — некоторая последовательность действительных чисел. Произвести сортировку последовательности x означает так переставить элементы этой последовательности, чтобы полученная последовательность $y = \{y_1, y_2, \dots, y_n\}$ удовлетворяла условию $y_1 \leq y_2 \leq y_3 \leq \dots \leq y_n$.

Иначе говоря, сортировка числовой последовательности это переупорядочение ее элементов в такую последовательность, где все элементы расположены в порядке неубывания.

Методы сортировки принято подразделять на внутренние, когда исходная последовательность размещается полностью в оперативной памяти ЭВМ или внешних устройствах с произвольным доступом, и на внешние, когда исходная последовательность размещена вне памяти

е произвольным поступом. Внутренняя сортировка является составной частью многих алгоритмов.

Существует очень много различных алгоритмов сортировки числовых последовательностей. Опишем здесь один из методов сортировки. Этот метод известен также под названием "метод пузырька". Он наиболее прост и "наивный" среди методов сортировки и используется, в основном, когда число элементов в сортируемой последовательности невелико.

Описание метода крайне просто. Пусть исходная последовательность $X = \{x_1, x_2, \dots, x_n\}$ содержит n чисел. Обозначим через j уменьшенный на единицу номер элемента, который уже занял свое место в отсортированной последовательности. Текущий номер элемента в последовательности обозначим через i .

Приведем описание алгоритма пузырька по шагам.

Шаг 1. Положить $j = n - 1$.

Шаг 2. Положить $i = 1$.

Шаг 3. Если $x_{i+1} < x_i$, то переставить x_i и x_{i+1} местами, иначе-ничего не делать.

Шаг 4. Увеличить i на единицу.

Шаг 5. Если $i \leq j$ перейти к шагу 3.

Шаг 6. Уменьшить j на единицу.

Шаг 7. Если $j \geq 1$ перейти к шагу 2.

Шаг 8. Отпечатать полученную отсортированную последовательность.

Рассмотрим работу алгоритма на примере, при этом будем выполнение шагов алгоритма со второго по шестой называть проходом алгоритма. На каждом проходе значение j уменьшается на единицу. В качестве исходной возьмем последовательность:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
3	5	1	2	7	4	8	10	9	6

Первый проход алгоритма распишем полностью.

Шаг 1. $j = 10 - 1 = 9$.

Шаг 2. $i = 1$.

Шаг 3. 3 5 1 2 7 4 8 10 9 6 (так как $x_{i+j} = 5$, $x_i = 3$, $5 > 3$).

Шаг 4. $i = 2$.

Шаг 5. 3 1 5 2 7 4 8 10 9 6 (так как $x_{i+j} = 1$, $x_i = 5$, $1 < 5$)

$i = 3$; 3 1 2 5 7 4 8 10 9 6

$i = 4$; 3 1 2 5 7 4 8 10 9 6

$i = 5$; 3 1 2 5 7 4 8 10 9 6

$i = 6$; 3 1 2 5 4 7 8 10 9 6

$i = 7$; 3 1 2 5 4 7 8 10 9 6

$i = 8$; 3 1 2 5 4 7 8 9 10 6

$i = 9$; 3 1 2 5 4 7 8 9 6 10

Проход 2. $j = 8$.

Запишем конечную последовательность прохода 2 и всех последующих проходов:

1 2 3 4 5 7 8 6 9 10

Проход 3. $j = 7$.

1 2 3 4 5 7 6 8 9 10

Проход 4. $j = 6$.

1 2 3 4 5 6 7 8 9 10

Хорошо видно, что исходная последовательность отсортирована и остаются пять проходов алгоритма, в данном случае лишние.

Название метода объясняется тем, что на каждом проходе очередное большое число поднимается в область больших номеров последовательности, как "пузырек воздуха со дна водоема".

Приведем текст программы модернизированного алгоритма пузырька. Отметим, что программа сортирует последовательность "на месте" с использованием всего одной вспомогательной переменной A . Переменная A служит для указания была ли выполнена в данном проходе алгоритма хотя бы одна перестановка элементов последовательности, если "да" то $A = 1$, если "нет", то $A = 0$. Остальные обозначения такие же как в описании алгоритма.


```

10 DIM X (50)
20 INPUT N
30 FOR K=1 TO N
40 INPUT X(K)
50 NEXT K
60 FOR J=N-1 TO 1 STEP -1
70 LET A=0
80 FOR I=1 TO J
90 IF X(I) <= X(I+1) THEN 140
100 LET Z = X(I)
110 LET X(I) = X(I+1)
120 LET X(I+1) = Z
130 LET A=1
140 NEXT I
150 IF A=0 THEN 170
160 NEXT J
170 FOR L=1 TO N
180 PRINT X(L)
190 NEXT L
200 STOP
210 END

```

Варианты заданий к лабораторной работе 4

Произвести сортировку последовательностей:

- 1) 1, 7, 12, 2, 4, 3, 1, 8, 2, 7, 6, 11, 4, 3;
- 2) 9, 4, 3, 1, 3, 7, 4, 5, 2, 2, 8, 7, 5;
- 3) 6, 4, 2, 1, 6, 8, 2, 4, 5, 7, 3, 1;
- 4) 1, 3, 2, 5, 4, 7, 1, 5, 2, 4, 3, 1, 5;

- 5) 6,3,2,5,7,9 11, 6, 1, 4, 2, 10;
 - 6) 7, 5, 3, 1, 2, 4, 6, 8, 1, 2, 3, 10, 1;
 - 7) 8, 6, 4, 2, 1, 3, 5, 2, 3, 4, 5, 10, 9, 8;
 - 8) 2, 1, 4, 3, 6, 5, 8, 7, 10, 1, 9, 2, 8, 3;
 - 9) 10, 1, 2, 9, 3, 8, 7, 4, 5, 6, 10, 1, 2, 8, 9;
 - 10) 7, 4, 1, 10, 2, 4, 8, 9, 3, 5, 3, 1, 8;
 - 11) 1, 8, 2, 1, 7, 8, 7, 2, 1, 9, 2, 10, 5, 4;
 - 12) 2, 6, 10, 7, 4, 1, 3, 5, 7, 9, 3, 7, 6, 5;
 - 13) 8, 9, 10, 8, 6, 4, 2, 3, 4, 5, 7, 9;
 - 14) 5, 6, 5, 1, 5, 2, 5, 4, 4, 8, 8, 10, 2;
 - 15) 1, 4, 7, 10, 8, 6, 4, 2, 3, 1, 5, 2, 6, 3, 7.
-

Перевод основных ключевых слов языка БЕЙСИК

<i>LINE ERASE</i>	- стирание строки
<i>ERASE</i>	- стереть
<i>BACKSPACE</i>	- на одно место назад
<i>INSERT</i>	-вставка
<i>DELETE</i>	- убрать, стереть
<i>EDIT</i>	- редактировать
<i>EDIT RECALL</i>	- вызвать на редактирование
<i>CARRIAGE RETURN / LINE FEED (CR/LF)</i>	-
	возврат каретки/перевод строки
<i>NEXT</i>	- следующий
<i>LET</i>	- пусть
<i>INPUT</i>	-ввести
<i>READ</i>	- читать
<i>DATA</i>	- данные
<i>FOR</i>	- для
<i>TO</i>	- к, до
<i>STEP</i>	- шаг
<i>IF</i>	- если
<i>THEN</i>	-то
<i>GO</i>	- идти
<i>STOP</i>	- остановиться
<i>END</i>	- конец

<i>DEFINITION (DEF)</i>	- определение
<i>FUNCTION (FN)</i>	- функция
<i>DIMENSION (DIM)</i>	- размерность
<i>RUN</i>	- выполнять
<i>LIST</i>	- список
<i>SELECT</i>	- выбрать
<i>PRINT</i>	- печатать

ПРИЛОЖЕНИЕ 2

Операторы языка БЕЙСИК, используемые в практикуме

№ п/п	Название	Форма оператора и примеры
1	Оператор присваивания	$LET \langle \text{переменная} \rangle = \langle a.b \rangle$ $LETA = B$ $LETA = SIN(X) * Y/Z$
2	Оператор безусловного перехода	$GOTO m$ $GOTO 50$
3	Оператор условного перехода	$IF \langle a.b \rangle < \text{знак отношения} \rangle \langle a.b \rangle THEN m$ $IF X + Y < X * SIN Y THEN 50$ $IF A <= 1.5 THEN 100$
4	Оператор начала цикла	$FOR \langle \text{простая переменная} \rangle = \langle a.b \rangle TO \langle a.b \rangle STEP \langle a.b \rangle$ $FOR I = 1 TO 5 STEP 2$ $FOR X = 4 TO Y STEP -3$ $FOR I = 1 TO 10$

Управляющие операторы

№ п/п	Название оператора	Форма оператора и примеры
1	Задание счетного режима	<i>RUN</i>
2	Вывод программы из памяти	<i>LIST</i>
3	Задание режима редактирования	<i>EDIT</i>
4	Вызов строки из памяти для редактирования	<i>m EDIT RECALL</i>
5	Подключение цифропечати для вывода программы из памяти	<i>SELECT LIST 05</i>
6	Подключение БОСГИ для вывода программы из памяти	<i>SELECT LIST 05</i>

С п и с о к л и т е р а т у р ы

1. Б а х в а л о в И.С. Численные методы. - М.: Наука, 1973.
 2. У о р т Т. Программирование на языке БЕЙСИК.- М.: Машиностроение, 1981.
 3. Ч е р н у х и н С.И., Ш а х у н я н ц Т.Г. Программирование на языке БЕЙСИК. Методические указания к практическим и лабораторным работам.-М.: Изд. МИНГ, 1983.
 4. Д е м и д о в и ч Г.П., М а р о н И.А., Ш у - в а л о в а Э.З. Численные методы анализа. - М.: На- ука, 1982.
 5. К е т к о в Ю.Л. Программирование на БЕЙСИК.- М.: Статистика, 1978.
-

СО Д Е Р Ж А Н И Е

Введение	3
Описание языка БЕЙСИК	6
Упражнение 1	8
Упражнение 2	15
Упражнение 3	16
Лабораторная работа 1	22
Лабораторная работа 2	29
Упражнение 4	34
Лабораторная работа 3	37
Варианты заданий к лабораторной работе 3	42
Лабораторная работа 4	45
Варианты заданий к лабораторной работе 4	48
Приложение 1. Перевод основных ключевых слов языка БЕЙСИК	50
Приложение 2. Операторы языка БЕЙСИК, ис- пользуемые в практике	51
Приложение 3. Управляющие операторы	53
Список литературы	54

Михаил Борисович Аверинцев, Юрий Павлович Власов,
Галина Федоровна Канаева, Игорь Ильич Меламед

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К ЛАБОРАТОРНОМУ ПРАКТИКУМУ
ПО ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКЕ

Редактор Г.Е.П е р к о в с к а я
Техн.редактор О.А.О в е ч к и н а
Корректор Т.С.И в а н о в а

Подп. в печ. 20.02.86
Формат 60x90 1/16. Печ.л. 3,5 Уч.-изд.л. 2,4
Заказ 616 Тираж 500 Бесплатно
Редакционно-издательский отдел МИИТа

Типография МИИТа, Москва, ул. Образцова, д. 15