

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ
Кафедра экономической математики, информатики и статистики (ЭМИС)

ОСНОВЫ АЛГОРИТМИЗАЦИИ. СРЕДА VISUAL STUDIO. ВВЕДЕНИЕ В
ЯЗЫК СИ

Отчёт по лабораторной работе по дисциплине “Введение в
программирование”

Студент гр. 543-1



/А.Е. Мухамеджан

“ ” 2023

Руководитель доцент
кафедры ЭМИС

_____ Е.А. Шельмина

“ ” 2023г.

Цель работы: изучить структуру программы на языке Си, операторы присваивания, ввода и вывода данных, используемые при составлении программ линейной структуры.

Теоретические сведения

Структура программы на языке Си

Выполнение всех программ, написанных на языке Си, начинается с функции, именуемой `main` (рисунок 1).

директивы_препроцессора
`int main()`
{
 определения_объектов;
 исполняемые операторы;
 `return 0;`
}

директивы_препроцессора
`void main()`
{
 определения_объектов;
 исполняемые операторы;
 `return;`
}

Рисунок 1 – Скриншот вида структуры программы

У функции есть имя (`main`), после которого в круглых скобках перечисляются аргументы или параметры функции (в данном случае у функции `main` аргументов нет). У функции может быть результат или возвращаемое значение. Если функция не возвращает никакого значения, то это обозначается ключевым словом `void`. В фигурных скобках записывается тело функции – действия, которые она выполняет. Оператор `return 0;` означает, что функция возвращает результат – целое число 0.

Вслед за заголовком функции `main` в фигурных скобках размещается тело функции, которое представляет последовательность определений, описаний и исполняемых операторов. Как правило, определения и описания размещаются до исполняемых операторов. Каждое определение, описание и оператор завершается «;».

Алфавит языка

Множество символов используемых в языке Си можно разделить на пять групп.

1. Следует отметить, что одинаковые прописные и строчные буквы считаются различными символами, так как имеют различные коды (рисунок 2).

Прописные буквы 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
Строчные буквы 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
Символ подчеркивания	_
Арабские цифры	0 1 2 3 4 5 6 7 8 9

Рисунок 2 – Скриншот таблицы символов, используемых для образования ключевых слов и идентификаторов

2. Знаки нумерации и специальные символы. Эти символы используются с одной стороны для организации процесса вычислений, а с другой – для передачи компилятору определенного набора инструкций (рисунок 3).

Символ	Наименование	Символ	Наименование
,	запятая)	круглая скобка правая
.	точка	(круглая скобка левая
;	точка с запятой	}	фигурная скобка правая
:	двоеточие	{	фигурная скобка левая
?	вопросительный знак	<	меньше
'	апостроф	>	больше
!	восклицательный знак	[квадратная скобка
	вертикальная черта]	квадратная скобка
/	дробная черта	#	номер
\	обратная черта	%	процент
~	тильда	&	амперсанд
*	звездочка	^	логическое не
+	плюс	=	равно
-	минус	"	кавычки

Рисунок 3 – Скриншот таблицы знаков нумерации и специальных символов

3. Управляющие и разделительные символы. К этой группе символов относятся: пробел, символы табуляции, перевода строки, возврата каретки, новая страница и новая строка. Эти символы отделяют друг от друга объекты, определяемые пользователем, к которым относятся константы и идентификаторы. Последовательность разделительных символов рассматривается компилятором как один символ (последовательность пробелов).

4. Кроме выделенных групп символов в языке Си широко используются так называемые, управляющие последовательности, т.е. специальные символьные комбинации, используемые в функциях ввода и вывода информации. Управляющая последовательность строится на основе использования обратной дробной черты (\) (обязательный первый символ) и комбинацией латинских букв и цифр (рисунок 4).

Управляющая последовательность	Наименование
\a	Звонок
\b	Возврат на шаг
\t	Горизонтальная табуляция
\n	Переход на новую строку
\v	Вертикальная табуляция
\r	Возврат каретки
\f	Перевод формата
\"	Кавычки
\'	Апостроф
\0	Ноль-символ
\\	Обратная дробная черта

Рисунок 4 – Скриншот таблицы управляющих последовательностей

Переменные и константы Идентификатор: Для символического обозначения величин, переменных, констант, функций и т.п. используются имена или идентификаторы (рисунок 5).

<i>asm</i>	<i>auto</i>	<i>bad_cast</i>
<i>bad_typeid</i>	<i>bool</i>	<i>break</i>
<i>case</i>	<i>catch</i>	<i>char</i>
<i>class</i>	<i>const</i>	<i>const_cast</i>
<i>continue</i>	<i>default</i>	<i>delete</i>
<i>do</i>	<i>double</i>	<i>dynamic_cast</i>
<i>else</i>	<i>enum</i>	<i>extern</i>
<i>float</i>	<i>for</i>	<i>friend</i>
<i>goto</i>	<i>if</i>	<i>inline</i>
<i>int</i>	<i>long</i>	<i>mutable</i>
<i>namespace</i>	<i>new</i>	<i>operator</i>
<i>private</i>	<i>protected</i>	<i>public</i>
<i>register</i>	<i>reinterpret_cast</i>	<i>return</i>
<i>short</i>	<i>signed</i>	<i>sizeof</i>
<i>static</i>	<i>static_cast</i>	<i>struct</i>
<i>switch</i>	<i>template</i>	<i>then</i>
<i>this</i>	<i>throw</i>	<i>try</i>
<i>type_info</i>	<i>typedef</i>	<i>typeid</i>
<i>union</i>	<i>unsigned</i>	<i>using</i>
<i>virtual</i>	<i>void</i>	<i>volatile</i>
<i>while</i>	<i>xalloc</i>	

Рисунок 5 – Скриншот списка ключевых слов

Идентификаторы в языке Си – это последовательность знаков, начинающаяся с буквы или знака подчеркивания. В идентификаторах можно

использовать заглавные и строчные латинские буквы, цифры и знак подчеркивания. Примеры правильных идентификаторов:

abc A12 NameOfPerson BITES_PER_WORD

abc и Abc – два разных идентификатора, т.е. заглавные и строчные буквы различаются. Примеры неправильных идентификаторов:

- 23X a-b

Ряд слов в языке Си имеет особое значение и не может использоваться в качестве идентификаторов. Такие зарезервированные слова называются ключевыми.

Переменная

Программа оперирует информацией, представленной в виде различных объектов и величин. Переменная – это символическое обозначение величины в программе. Как ясно из названия, значение переменной (или величина, которую она обозначает) во время выполнения программы может изменяться.

С точки зрения архитектуры компьютера, переменная – это символическое обозначение ячейки оперативной памяти программы, в которой хранятся данные. Содержимое этой ячейки – это текущее значение переменной.

В языке Си прежде чем использовать переменную, ее необходимо объявить. Объявить переменную с именем x можно так: `int x;` В объявлении первым стоит название типа переменной `int` (целое число), а затем идентификатор `x` – имя переменной. У переменной `x` есть тип – в данном случае целое число.

Основные типы данных

В языке Си существует несколько стандартных основных типов данных:

char, short, int, long, float, double

Первые четыре типа используются для представления целых, последние два – для представления чисел с плавающей точкой. Переменная типа char имеет размер, естественный для хранения символа на данной машине (обычно, байт), а переменная типа int имеет размер, соответствующий целой арифметике на данной машине (обычно, слово).

Для определения данных целого типа используются различные ключевые слова, которые определяют диапазон значений и размер области памяти, выделяемой под переменные (рисунок 6).

Тип	Размер памяти в байтах	Диапазон значений
<i>char</i>	1	от -128 до 127
<i>int</i>	2	от -32768 до 32767
<i>short</i>	2	от -32768 до 32767
<i>long</i>	4	от -2 147 483 648 до 2 147 483 647
<i>unsigned char</i>	1	от 0 до 255
<i>unsigned int</i>	2	от 0 до 65535
<i>unsigned short</i>	2	от 0 до 65535
<i>unsigned long</i>	4	от 0 до 4 294 967 295

Рисунок 6 – Скриншот таблицы основных типов памяти

Для переменных, представляющих число с плавающей точкой используются следующие типы: float, double, long double.

Величина типа float занимает 4 байта. Из них 1 байт отводится для знака, 8 бит для избыточной экспоненты и 23 бита для мантиссы. Отметим, что старший бит мантиссы всегда равен 1, поэтому он не заполняется, в связи с этим диапазон значений переменной с плавающей точкой приблизительно равен от $3.14E-38$ до $3.14E+38$.

Величина типа double занимает 8 бит в памяти. Ее формат аналогичен формату float. Биты памяти распределяются следующим образом: 1 бит для знака, 11 бит для экспоненты и 52 бита для мантиссы. С учетом опущенного старшего бита мантиссы диапазон значений равен от $1.7E-308$ до $1.7E+308$.

Операторы языка программирования

Оператором называется элементарная структурная единица программы. Оператор предназначен как для записи алгоритмических действий по

преобразованию данных, так и для задания порядка выполнения других действий. Операторы выполняются в порядке их следования в программе. Операторы отделятся друг от друга точкой с запятой. Операторы делятся на:

- простые (не содержат в себе других операторов);
- составные (включают в себя один или несколько дополнительных операторов).

Присваивание

Переменной можно присвоить какое-либо значение с помощью операции присваивания. Присвоить – это значит установить текущее значение переменной. По-другому можно объяснить, что операция присваивания запоминает новое значение в ячейке памяти, которая обозначена переменной.

Ввод-вывод с использованием стандартной библиотеки ввода-вывода stdio.h

Все возможности организации ввода-вывода СИ реализованы в библиотечных функциях стандартной библиотеки `stdio.h`.

Для организации вывода используется функция:

`printf(форматная_строка, список_аргументов);`

Форматная строка ограничивается кавычками «”» и может включать произвольный текст, управляющие символы и спецификации преобразования данных.

При организации вывода данных на экран используются спецификации преобразования, которые имеют следующий обобщённый вид:

- %флажки ширина_поля.точность модификатор спецификатор
(рисунок 7)

Модификатор	Назначение
N	Для близкого указателя
F	Для дальнего указателя
h	Для значения short int
l	Для значения long
L	Для значения long double

Рисунок 7 – Скриншот таблицы назначения модификаторов
Список аргументов может отсутствовать.

Обязательными являются «%» и спецификатор (рисунок 8).

Спецификатор	Тип аргумента	Назначение
<i>d</i>	Целого типа	Для целых десятичных чисел (int)
<i>i</i>	Целого типа	Для целых десятичных чисел (int)
<i>o</i>	Целого типа	Для беззнаковых восьмеричных целых
<i>u</i>	Целого типа	Для беззнаковых десятичных целых
<i>x</i>	Целого типа	Для беззнаковых шестнадцатеричных целых (a,b,c,d,e,f)
<i>X</i>	Целого типа	Для беззнаковых шестнадцатеричных целых (A,B,C,D,E,F)
Спецификатор	Тип аргумента	Назначение
<i>f</i>	вещественный	Знаковое вещественное число в формате [+/-]ddd.dddd
<i>e</i>	вещественный	Знаковое вещественное число в формате [+/-]d.dddd или в экспоненциальной форме
<i>g</i>	вещественный	Знаковое вещественное число в формате или f, или e (в зависимости от выводимого значения)
<i>E</i>	вещественный	Такое же, как и e
<i>G</i>	вещественный	Такое же, как и g
<i>s</i>	строковый	ввод-вывод строковых данных
<i>c</i>	символьный	ввод-вывод символов

Рисунок 8 – Скриншот таблицы назначения спецификаторов

Математические функции

Для выполнения математических вычислений в стандартной математической библиотеке описаны следующие функции:

- `int abs (int k) ; double fabs(double x);` Возвращает целое (abs) или дробное (fabs) абсолютное значение аргумента, в качестве которого можно использовать выражение соответствующего типа.
- `double acos (double x);`
- `double asin (double x);`
- `double atan (double x);`
- `long double acosl(long double x) ;`
- `long double asinl(long double x);`

- `long double atanl(long double x);`

Возвращает выраженную в радианах величину угла, арккосинус, арксинус или арктангенс которого передан соответствующей функции в качестве аргумента. Аргумент функции должен находиться в диапазоне от -1 до 1.

- `double cos (double x);`
- `double sin (double x);`
- `double tan (double x);`
- `long double cosl(long double x);`
- `long double sinl(long double x);`
- `long double tanl(long double x);`

Возвращает синус, косинус или тангенс угла. Величина угла должна быть задана в радианах.

- `double exp(double x);`
- `long double exp(long double lx);`

Возвращает значение, равное экспоненте аргумента (e^x , где e — основание натурального логарифма).

- `double pow (double x, double y);`
- `long double powl(long double (x), long double (y));`

Возвращает значение, равное x^y .

- `double sqrt(double k);`

Возвращает значение, равное квадратному корню из аргумента.

- `double log(double x);`
- `double log10(double x);`
- `long double logl(long double (x));`
- `long double log10l(long double (x));`

`log`, `logl` — возвращают значение натурального логарифма аргумента.

`log10`, `log10l` — возвращают значение логарифма аргумента по основанию 10.

В библиотеке `<stdlib.h>` описаны генераторы случайных чисел.

`int rand(void);` Возвращает случайное целое число в диапазоне от 0 до `RAND_MAX`. Перед первым обращением к функции `rand` необходимо инициализировать генератор случайных чисел. Для этого надо вызвать функцию `srand`. `void srand(unsigned k);` Инициализирует генератор случайных чисел. Обычно в качестве параметра функции используют переменную, значение которой предсказать заранее нельзя, например это может быть текущее время.

Ход работы:

Задание 1. Написать программу на языке Си для решения задачи (рисунок 8).

<p>Вариант 12</p>	<p>$a=5.1$ $\alpha=0.1$</p>	$y_1 = \frac{\sqrt{a} + 1}{a\sqrt{a} + a + \sqrt{a}} : \frac{1}{a^2 - \sqrt{a}};$ $y_2 = a - 1;$ $z_1 = \frac{\sin \alpha - \sin 3\alpha + \sin 5\alpha}{\cos \alpha - \cos 3\alpha + \cos 5\alpha};$ $z_2 = \operatorname{tg} 3\alpha$
--------------------------	---	---

Рисунок 8 – Скриншот условия к заданию №1

Программа для решения и вывод задания №1 представлена на рисунке 9.

```

1  #include <stdio.h>
2  #include <math.h>
3
4
5  int main()
6  {
7      float a = 5.1;
8      float alpha = 0.1;
9      printf("y1 = %4.3f\n", ((sqrt(a) + 1) / (a * sqrt(a) + a + sqrt(a)) / (1 / (pow(a,2) - sqrt(a)))));
10     printf("y2 = %4.3f\n", (a - 1));
11     printf("z1 = %4.3f\n", ((sin(alpha) - sin(3 * alpha) + sin(5 * alpha) /
12         (cos(alpha) - cos(3 * alpha) + cos(5 * alpha)))));
13     printf("z2 = %4.3f\n", (tan(3 * alpha)));
14     return 0;
15 }

```

Консоль отладки Microsoft Visual Studio

```

y1 = 4.100
y2 = 4.100
z1 = 0.327
z2 = 0.309

```

Рисунок 9 - Скриншот программы к заданию №1

Задание 2. Разработать блок-схему и составить программу для вычисления результата по приведенной формуле и вывода на печать

исходных данных и результата в виде таблицы (рисунок 10). Исходные данные вводятся с клавиатуры. Приведенные в таблицах значения служат тестом правильной работы программы.

$$2. \quad Y = \left(\sin^2 x + \left(\cos x - \frac{\sin^4 x^2 + x^2}{x} - \frac{x^4 + 0,96x - 1}{x^4 + x^3 - 2,37} \right)^3 + \sqrt{x} \right)^4$$

$x=0,85$

	GRUPPA – N
	VARIANT – N 2
	:
X : 0.85	:
	:

	Y : 4.844E+0000
	YOUR_NAME

Рисунок 10 – Скриншот условия к заданию №2

Блок-схема к заданию №2 представлена на рисунке 11.

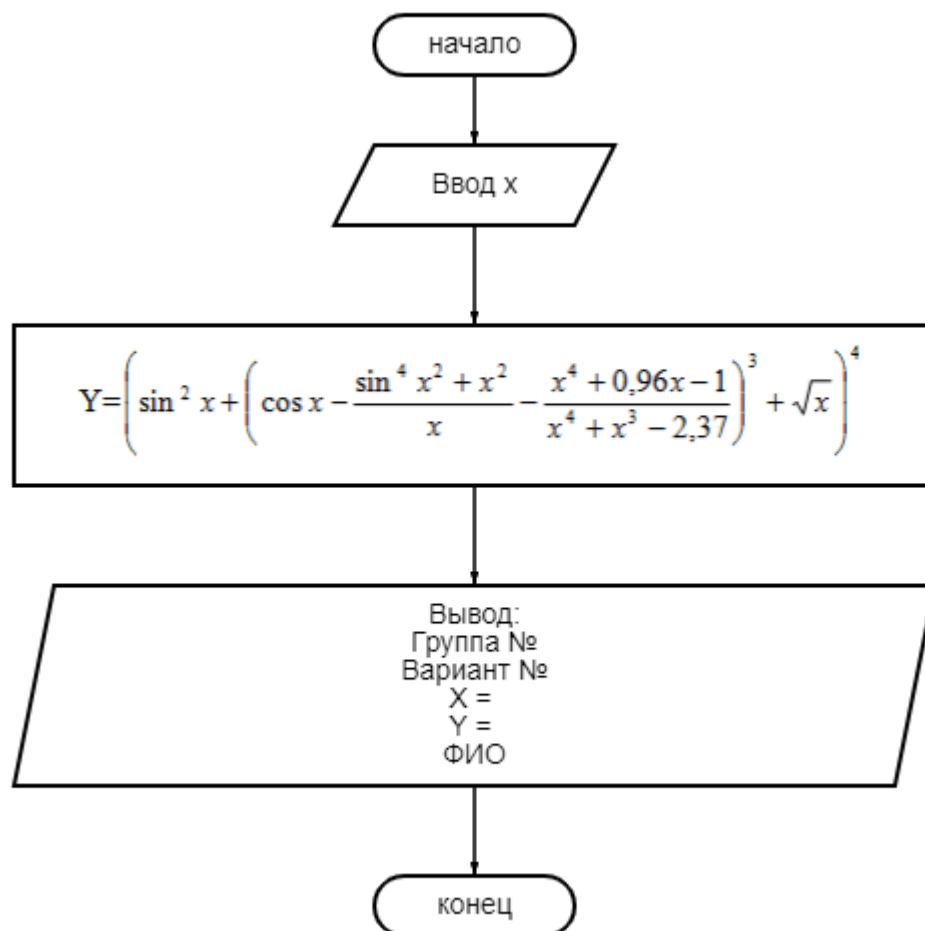


Рисунок 11 – Блок-схема к заданию №2

12.


$$s = 2^{(y^x)} + (3^x)^y - \frac{y \left(\operatorname{arctgz} - \frac{1}{3} \right)}{|x| + \frac{1}{y^2 + 1}}.$$

Блок-схема к заданию №3 представлена на рисунке 14.

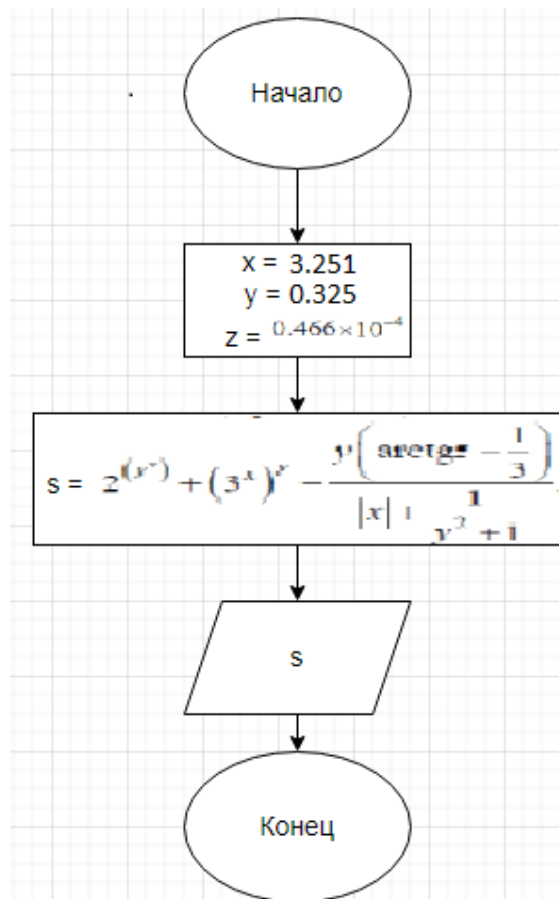


Рисунок 14 – Блок-схема к заданию №3

Программа для решения вывод задания №3 представлена на рисунке 15.

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float x, y, z;
    system("cls");
    printf("x,y,z: ");
    scanf("%f %f %f", &x,&y,&z);
    float s = pow(2, pow(y, x)) + pow(pow(3, x), y) - ((y * (atan(z) - 1 / 3)) / (fabs(x) + 1 / (pow(y, 2) + 1)));
    printf("\tГРУППА - N543-1\t \n\tВАРИАНТ - 12 2\n-----\n\tX = %3.2f\n-----\n\tY = %10.5f\n\t Muhamedzhan Adlet", x, s);
    return 0;
}
  
```

```

x,y,z: 3.251 0.325 0.000466
ГРУППА - N543-1
ВАРИАНТ - 12 2
-----
X = 3.25
-----
Y = 4.21048
Muhamedzhan Adlet
  
```

Рисунок 15 – Скриншот программы к заданию №3

Задание 4. Разработать блок-схему и написать программу на языке Си для решения приведенных ниже задач согласно своему варианту (рисунок 16).

Вариант 12: Ввести радиус, длину ребра и высоту конуса. Найти площадь боковой поверхности, площадь полной поверхности и объем конуса.

Рисунок 16 – Скриншот условия к заданию №4

Блок-схема к заданию №4 представлена на рисунке 17.

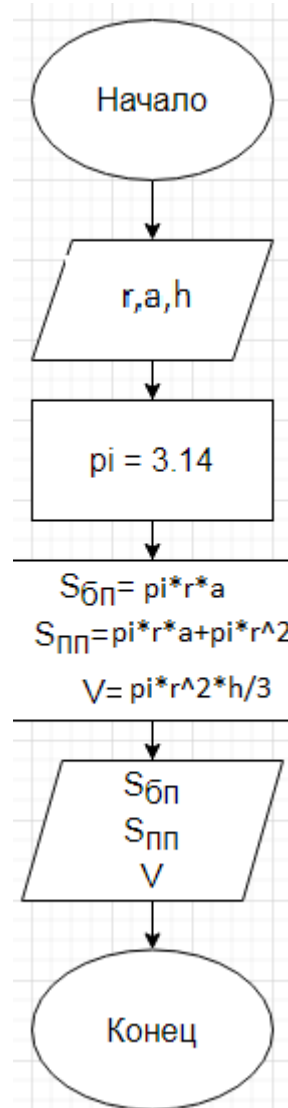


Рисунок 17. – Блок-схема к заданию №4

Программа для решения вывод задания №4 представлена на рисунке 18.

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <locale.h>

int main()
{
    setlocale(LC_CTYPE, "russian");
    float r, a, h, S1, S2, V;
#define pi 3.14159265358979323846
    printf("r,a,h: ");
    scanf("%f %f %f", &r, &a, &h);
    S1 = pi * r * a;
    S2 = pi * r * a + pi * pow(r, 2) ;
    V = pi * pow(r, 2) * h / 3;
    printf("S(бок) = %lf\nS(п.п.) = %lf\nV = %lf", S1, S2, V);
    return 0;
}
```

Консоль отладки Microsoft Visual Studio

r,a,h: 3 2 2
S(бок) = 18.849556
S(п.п.) = 47.123890
V = 18.849556

Рисунок 18 - Скриншот программы к заданию №4

Вывод: в ходе работы изучена структура программы на языке Си, а также операторы присваивания, ввода и вывода данных, используемые при составлении программ линейной структуры.