**Практическая работа №1.**

**Тема: «Основы алгоритмизации. Построение блок-схем линейных вычислительных процессов»**

**Цель лабораторной работы:**

Целью данной лабораторной работы является ознакомление с основами

программирования на языке Си, его особенностями, синтаксисом, компиляцией и отладкой.

1. **Теоретические сведения.**

Решение любой задачи на ЭВМ происходит в несколько этапов:

* постановка задачи;
* конструирование алгоритма решения задачи;
* составление программы по разработанному алгоритму;
* ввод в ЭВМ программы и исходных данных;
* отладка и тестирование программы;
* получение решения и анализ результатов.

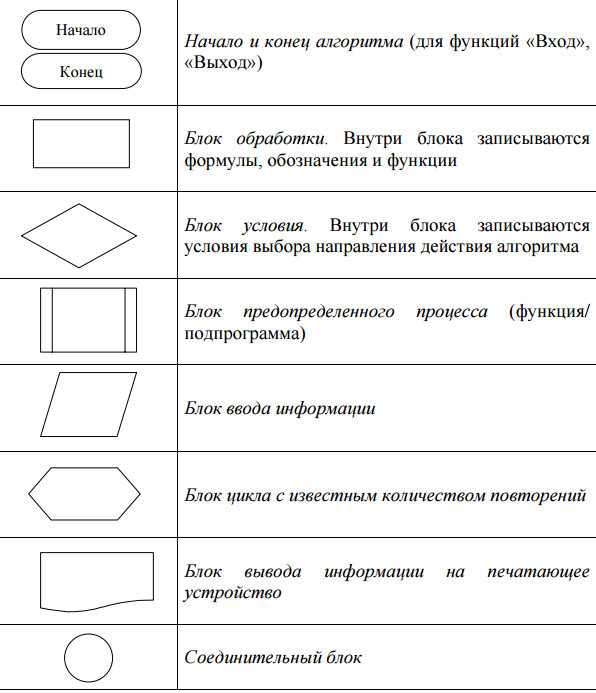
Алгоритм – конечная последовательность точно определенных действий, приводящих к однозначному решению поставленной задачи. Главная особенность любого алгоритма – формальное исполнение, позволяющее выполнять заданные действия (команды) не только человеку, но и различным техническим устройствам (исполнителям). Процесс составления алгоритма называется алгоритмизацией.

Алгоритмы могут быть заданы: словесно, таблично, графически (с помощью блок-схем). Словесное задание описывает алгоритм с помощью слов и предложений. Табличное задание служит для представления алгоритма в форме таблиц и расчетных формул. Графическое задание, или блок-схема, – способ представления алгоритма с помощью геометрических фигур, называемых блоками. Последовательность блоков и соединительных линий образуют блок-схему. Описание алгоритмов с помощью блок-схем является наиболее наглядным и распространенным способом задания алгоритмов. Блок-схемы располагаются сверху вниз. Линии соединения отдельных блоков показывают направление процесса обработки в схеме. Каждое такое направление называется ветвью. Алгоритм независимо от его структуры всегда имеет по одному блоку «Начало» и «Конец». Его ветви должны в конце сойтись, и по какой бы ветви не было бы начато движение, оно всегда должно привести к блоку «Конец».

При задании алгоритма с помощью блок-схемы используются строго определенные блоки.

Основные типы блоков приведены в таблице 1. Следует отметить, что все блоки нумеруются. В этом случае номера проставляются вверху слева от блока (блоки «Начало», «Конец» и соединительные блоки не умеруются). Стрелки на соединяющих линиях обычно не ставят при аправлении сверху вниз и слева направо; если направление противоположное, то его показывают стрелкой на линии.

**Таблица 1. Основные типы блоков.**

****

Алгоритмы бывают линейные, разветвляющиеся и циклические. Линейный алгоритм не содержит логических условий, имеет одну ветвь обработки и изображается линейной последовательностью связанных друг с Начало Конец19 другом блоков. Условное изображение линейного алгоритма может быть представлено на рис. 1.1.

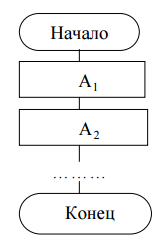


Рис.1.1 Блок-схема линейного алгоритма

Чаще всего линейные алгоритмы используются для программирования вычислений по формулам. В этом случае удобно использовать набор стандартных математических функций языка Си, содержащихся в заголовочном файле math.h : Sin (x) - sin x; cos (x) – cos x ; tan(x) - tg х ; log (x) - ln x; log10(x) – lg x; exp(x) - показательная функция e x ; sqrt (x)- корень квадратный от х; роw(x, y) – x в степени y; abs(x)- модуль x.; acos(x) – arccos x; asin(x) – arcsin x; atan(x)- arctg x; sinh(x)- sh x; cosh (x) – ch x; tanh(x)-th x. Для тригонометрических функций аргумент х измеряется в радианах и имеет тип double, как и значения возвращаемые функцией.

1. **Пример программы линейной структуры:**

Задание: Написать программу для вычисления площади и стороны прямоугольного треугольника, если известны гипотенуза c и угол x. Для вычислений воспользуемся формулами:

a=c\*sin x;

b=c\*cos x;

s=a\*b ;

1. **Реализация программы:**

*//lab1\_1 вычисление высот и площади треугольника*

*#include <stdio.h> //заголовочный файл для организации ввода-вывода*

*#include <math.h> //заголовочный файл математических функций*

*main() //главная функция*

*{ float a,b,c,x, s; //описание переменных вещественного типа*

*printf(“\n Введите гипотенузу и угол треугольника ”);*

*scanf(“%f%f “,&c,&x); // ввод значений вещественного типа*

*a=c\*sin(x);*

*b=c\*cos(x);*

*s=a\*b;*

*printf (“\n Результаты: a=%7.2f b=%7.2f S=%7.2f “,a, b, s);*

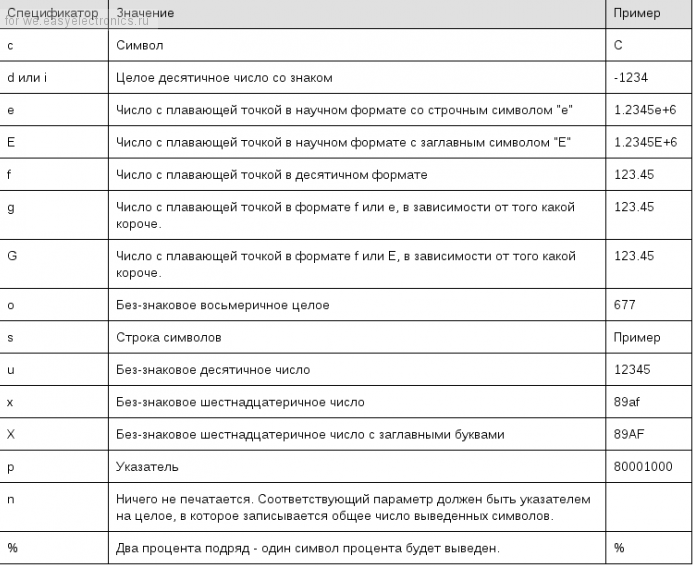
*return 0;*

*}*

Комментарий в программе не влияет на компиляцию программы, а служит для разъяснения действий основных блоков текста и программы в целом. Однострочный комментарий действует от двух символов // до конца строки. Многострочный комментарий заключается в пары символов /\* и \*/. Вторая строка программы #include является директивой препроцессора для подключения заголовочного файла stdio.h , в котором содержатся средства ввода-вывода (связи с внешними устройствами), отсутствующие в самом языке Си. Третья строка - #include является директивой компилятора для включения заголовочного файла math.h, обеспечивающего выполнение математических функций. Программа на языке Си состоит из ряда функций, из которых функция main (главная) является обязательной и служит точкой входа в программу. В круглые скобки заключаются параметры функции, причем наличие круглых скобок обязательно, если даже список параметров пуст. В фигурные скобки заключаются составной оператор (несколько операторов). Для придания тексту программы наглядности открывающая и соответствующая ей закрывающая фигурные скобки печатаются на одном уровне, а заключенный между ними текст, сдвигается на 1-2 символа вправо, вложенный блок также сдвигается вправо и т.д. Образуется иерархия вложенных блоков, придающая программе на Си характерный вид. В программе описаны переменные a, b, c, х, s вещественного типа (float). В отличие от других языков в Си учитывается регистр при определении имени переменной, т.е. s и S это разные переменные. Функции printf() и scanf() содержатся в заголовочном файле <stdio.h> и служат для вывода на экран и ввода с клавиатуры соответственно. Управляющие символы \n в функции printf() служат для перевода на новую строку. Символ & в функции scanf() указывает на адрес вводимой переменной. Ввод и вывод переменных вещественного типа производится в формате %f. Признаком форматного вывода в функции printf() является наличие символа %. А буква f означает, что будет выводиться переменная вещественного типа. При выводе между знаком процента и форматной переменной f можно включить общую ширину поля вывода и число позиций после десятичной точки. Например %4.3f означает ширину поля вывода 4 символа, начиная слева и 3 знака после запятой. Оператор return 0; служит для выхода из функции main() и является последним оператором программы, в данном случае возвращаемое значение 0(ноль) является признаком успешного окончания программы. Завершает текст программы или тело функции main() закрывающая фигурная скобка, которая означает конец.

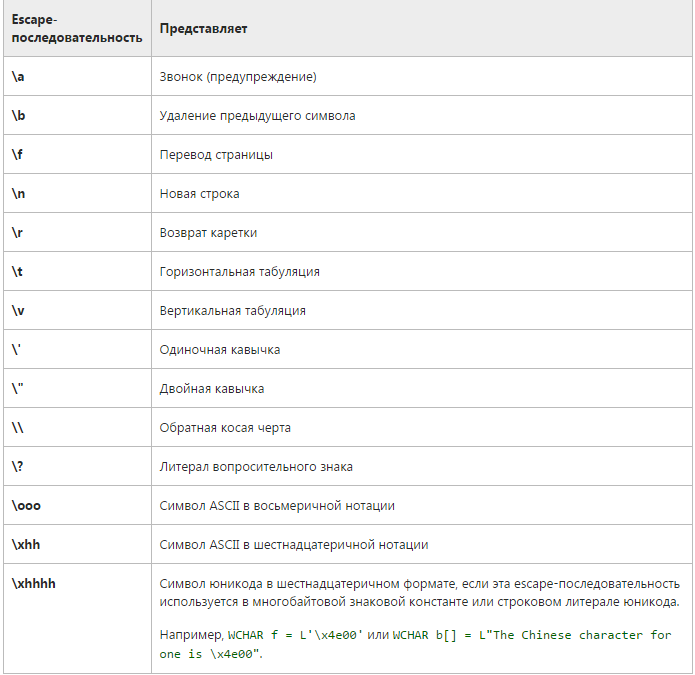
1. **Форматная строка**

Состоит из спецификаторов вывода для различных типов данных, которым предшествует специальный неотображаемый символ процента % и используется в функциях scanf() и printf()

****

1. **Символы, управляющие выводом (эскейп символы)**

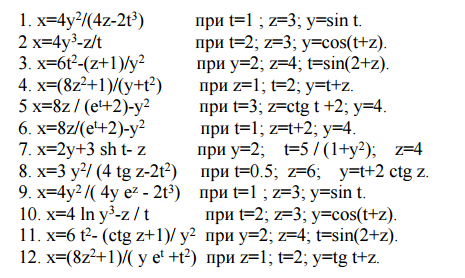
Позволяют из форматной строки управлять выводом на экран

****

1. **Выполнение работы**

Наберите и выполните приведенную выше программу вычисления высот треугольника по заданным сторонам. 3. Составьте и выполните программу линейной структуры согласно вариантам заданий пункта 7.

1. **Варианты заданий.**



1. **Контрольные вопросы**

Что такое алгоритм линейной структуры, программа линейной структуры?

Для чего используются заголовочные файлы?

Структура программы на языке Си

1. **Содержание отчета**

Отчет должен содержать: титульный лист, название и цель и описание работы, задание, код программы, алгоритм в виде блок-схемы и скриншоты с результатом работы программы.