**ЛАБОРАТОРНАЯ РАБОТА № 10**

**С++: ФУНКЦИИ**

## ТЕОРИЯ И ПОЯСНЕНИЯ

### *Объявление функции*

Задание прототипа функции, т.е. имени функции, типов параметров и типа возвращаемого значения (имена формальных параметров можно не задавать). Функция, не возвращающая значения, описывается типом .. Если функция не имеет параметров, то перечисление типов параметров заменяется на  (в C++ можно вообще опускать). Функцию нужно объявлять, если по тексту программы вызов функции производится раньше, чем встречается ее определение.

Примеры объявлений функции:

******

### *Определение функции*

Задание исходного кода функции. Все функции в С находятся на одном уровне видимости, поэтому функции не могут быть вложенными, т.е. нельзя определить одну функцию внутри другой. Если функции объявлены, то их определения в файле с исходным текстом программы могут следовать друг за другом в любом порядке.

При определении функции в заголовке должны быть объявлены не только типы, но и имена всех формальных параметров. Областью видимости параметров является тело функции. Областью видимости локальной переменной, объявленной в теле функции, является блок, в котором выполнено это объявление. Наряду с формальными параметрами и локальными переменными в функции можно использовать глобальные данные.

Возвращаемое функцией значение указывается с помощью оператора

 .

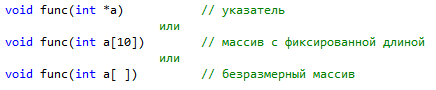
Например,



Выполнение оператора приводит к завершению выполнения функции. В функции можно использовать несколько операторов для создания нескольких точек выхода. Если функция не содержит оператора , то ее выполнение завершается без возврата значения.

### *Формальные параметры - массивы*

Одномерный массив можно объявить как указатель, как статический массив или как безразмерный массив. Например,



Все три способа объявления сообщают компилятору, что предполагается получение указателя на целое.

Двумерный массив (матрица) должен быть объявлен так, чтобы было известно число столбцов для корректной индексации массива. Например,

******

### *Обращение к функции*

Обращение к функции (вызов функции), возвращающей значение, является операндом соответствующего типа. Таким образом, вызов функции допустим в программе везде, где разрешено использовать выражение. Вызов функции, не возвращающей значение, является отдельным оператором.

Фактические параметры должны соответствовать формальным параметрам по типу и порядку следования.

При вызове функции происходит передача ЗНАЧЕНИЙ фактических параметров. Эти значения локализуются внутри функции, их изменения не будут видны после завершения работы функции. Для того чтобы функция могла изменять значения, находящиеся в памяти по определенным адресам, следует передавать адреса этих значений. В этих случаях в качестве формальных параметров используются указатели, а фактические параметры могут быть указателями или вычисляемыми адресами переменных. Например,

|  |  |
| --- | --- |
| Заголовок функции | Вызов функции |
|  |  |
|  |

## ЗАДАНИЯ

### УПРАЖНЕНИЕ 1. Определите (без компиляции и выполнения на компьютере), какие значения будут выведены в результате выполнения каждой из приведенных ниже программ:

|  |  |
| --- | --- |
|  |  |

### УПРАЖНЕНИЕ 2 («Гармошка»). Задан упорядоченный целочисленный массив длины N>>1000. Значения элементов массива находятся в числовом интервале [0, 17].

А. Напишите функцию с параметрами, которая «упаковывает» данные, т.е. размещает информацию о числах из массива, используя как можно меньше памяти.

Б. Напишите функцию с параметрами, которая «распаковывает» данные, т.е. по информации, полученной при выполнении задания А, восстанавливает исходное состояние массива.