## Лабораторная работа №9

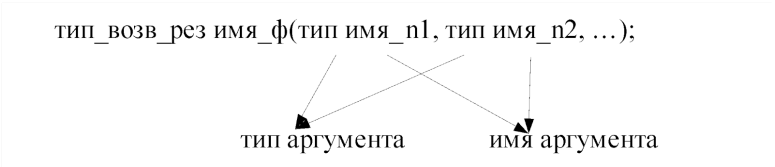
## Функции

**Цель работы:** Научиться создавать и использовать новые функции пользователя

Теоретические сведения

Программа на С составляется из одной или более функций, одна из которых main(), она называется головной функцией. Выполнение программы всегда начинается с функции main().

Функция – самостоятельная единица программы, спроектированная для реализации конкретной задачи. Что нам требуется знать о функции? Необходимо знать, как функцию необходимо определить, т.е. написать код функции, как к функции обращаться, т.е. вызвать функцию на выполнение и как устанавливать связи между функцией и программой ее вызывающей. Чтобы установить связь между функцией и программой ее вызывающей, необходимо знать прототип функции. Прототип функции – это объявление функции. Прототип функции имеет следующую форму:




Например: float s_z(int a, int b);

Список аргументов может быть пустым, содержать один аргумент или несколько аргументов, разделенных запятыми. Если функция не имеет аргументов, допускается использования ключевого слова void. Если функция не имеет аргументов и она ничего не возвращает, то ее прототип можно записать в следующем виде:

void prim(void);

Вызов функции

Вызов функции осуществляется по ее имени. Для вышеописанной функции s_z(), ее надо вызвать на выполнение так:

```
int a=10, b=20;      // объявление и инициализация переменных
float rez;
rez=s_z(a,b);        // вызов функции на выполнение
```

**Определение функции**

В С код, описывающий, что делает функция, называется определением функции. Формально это выглядит так:

```
тип_возв_рез имя_ф(тип имя_n1, тип имя_n2, …)
{
        тело функции
```

}

Синтаксически аргументы – это идентификаторы, они могут использоваться внутри тела функции. Иногда параметры в определении функции называют формальными параметрами. Формальные параметры – это то, вместо чего будут подставлены фактические значения, передаваемые функции в момент ее вызова. После вызова функции значение аргумента, соответствующее формальному параметру, используется в теле выполняемой функции. В С такие параметры являются передаваемыми по значению. Когда применяется вызов по значению, переменные передаются функции как аргументы, их значения копируются в соответствующие параметры функции, а сами переменные не изменяются в вызывающем окружении.

**Инструкция return**

Инструкция return используется для двух целей. Когда она выполняется, управление программой немедленно передается обратно в вызывающее окружение. Кроме того, если за ключевым словом return следует какое-либо выражение, то его значение также передается в вызывающее окружение. Инструкция return имеет следующие формы записи:

return;
return выражение;
return (выражение);

Структура программы, содержащей несколько функций пользователя:

```
#include <stdio.h>
void main(void)
{
        объявление прототипов функций
        объявление переменных
        ввод данных
        вызов функций на выполнение
}
```

Например: написать функцию, которая вычисляет среднее значение двух целых чисел

```
#include <stdio.h>
void main(void)
{
        float s_z(int a, int b);   // прототип функции
        int a,b;                   // переменные
        float rez;                 // результат
        printf("\nВведите два целых числа\n";
        scanf("%d%d",&a,&b);
        rez=s_z(a,b);              // вызов функции на выполнение
        printf("\nСреднее значение равно %3d",rez);
}
float s_z(int a, int b)
{
```

```
        float r;
        r=((float)a+b)/2;        // или return (((float)a+b)/2);
        return r;
}
```