

Функции

В программе на C может быть любое количество функций, среди них нет иерархии. Но первой будет выполняться функция **main()**, где бы она ни стояла.

Чтобы вызвать функцию необходимо обратиться к ней по имени в качестве оператора или операнда, указав после имени в круглых скобках список аргументов, которые необходимо передать. Имена переменных в этом списке пишутся через запятую без указания типа. Такие аргументы называются **фактическими параметрами**. В результате вызова управление будет передано в вызванную функцию, а затем вернётся к вызывающей функции, когда вызванная функция выполнит все свои инструкции или когда встретится ключевое слово **return**.

Заголовок любой функции имеет вид:

Тип_возвращаемого_значения ИМЯ (список_параметров)

После заголовка должно быть описано тело функции в фигурных скобках.

Параметры, задаваемые в заголовке функции, называются формальными.

Фактические и формальные параметры должны совпадать по типу, количеству и порядку следования, однако могут иметь разные имена.

Если функция возвращает значение, то её можно вызывать и в качестве оператора, и в качестве операнда. Но, если вызвать её в качестве оператора, то возвращаемое значение будет пропадать.

Никакие действия, которые происходят с формальными параметрами в вызываемой функции, не могут привести к изменению значений фактических параметров. Так работает передача по значению.

Если функция ничего не возвращает, то её можно вызвать только в качестве оператора. У таких функций в заголовке в качестве типа возвращаемого значения указывается ключевое слово **void**.

При передаче по значению обратно в вызывающую функцию может вернуться только значение переменной, упомянутой в операторе **return**. И тип именно этой переменной и является типом возвращаемого значения, который указывается в заголовке функции.

Если из одной функции в другую в качестве параметра передаётся массив, то механизм передачи другой. В функцию передаётся **адрес нулевого элемента массива**, и она фактически получает доступ ко всем элементам массива. Такая передача называется **передачей по адресу**, и таким образом можно передавать не только массивы, но и простые переменные.

Если заголовок функции находится ниже ее первого вызова, такая функция нуждается в дополнительном описании, которое называется **прототипом** и имеет вид:

тип_возвращаемого_значения ИМЯ (список_типов_параметров);

obrabotka1(); - вызовы функций в качестве оператора
obrabotka2(x, &y);

a= obrabotka1(); - вызовы функций в качестве операнда
a= obrabotka2(x, &y);

int obrabotka1(); - прототипы функций
int obrabotka2(int, int*);

int obrabotka1() - заголовки функций
int obrabotka2(int n, int* m)

Все переменные, которые описаны внутри каких-либо фигурных скобок, являются локальными, что означает ограничение области действия переменных той функцией или тем блоком, в котором они описаны. При временном выходе из функции такая переменная недоступна, но её значение не теряется. При окончании работы функции переменная прекращает своё существование.

Существуют также глобальные (внешние) переменные. Они существуют вне отдельных функций и **доступны любой функции программы**, которая захочет к ним обратиться. Внешние переменные в обязательном порядке **описываются вне функций**.

Задание:

Решить задачу с разбиением решения на подпрограммы.

- Главная функция, из которой по мере необходимости бет производиться вызов функций ввода, обработки и вывода.
- Функция ввода данных
- Функция обработки данных (можно несколько)
- Функция вывода

До 34 баллов

```
#include <stdio.h>
#include <locale.h>
#pragma warning(disable: 4996)
описание глобальных переменных и массива;
void main(void) {
    setlocale(LC_ALL, "Russian");
    vvod();           // вызовы функций в качестве операторов
    obrabotka();
    vivod();
}
```

До 44 баллов

```
#include <stdio.h>
#include <locale.h>
#pragma warning(disable: 4996)
описание глобального массива;
void main(void) {
    setlocale(LC_ALL, "Russian");
    описание локальных переменных;
    n=vvod();        // вернуть размер массива
    obrabotka(/*передать размер массива*/);
    vivod(/*передать размер массива*/);
}
```

Больше 44 баллов

```
#include <stdio.h>
#include <locale.h>
#pragma warning(disable: 4996)
void main(void) {
    setlocale(LC_ALL, "Russian");
    описание локальных переменных и массива;
    n=vvod(/*передать массив*/);          // вернуть размер массива
    obrabotka(/*передать массив, размер массива*/);
    vivod(/*передать массив, размер массива*/);
}
```

Лабораторная работа №5.

Для выполнения лабораторной работы необходимо:

1. Получить индивидуальное задание.
2. Написать программу на языке программирования Си.
3. **ОТЛАДИТЬ И ПРОТЕСТИРОВАТЬ ПРОГРАММУ.**
4. Составить отчет о лабораторной работе.

Содержание отчета:

- Титульный лист.
- Условие задачи.
- Текст программы.
- Скриншоты, подтверждающие корректную работу программы с разными наборами входных данных.