ФУНКЦИИ СИ

СИ:
Все подпрограммы - функции

2 аспекта использования функций:

- **Описание**
- **Вызов**

ОПИСАНИЕ ФУНКЦИИ В СИ

Заголовок функции

БЛОК

{

Тело функции

}

```
void main()
{
  Тело
}
```

ОПИСАНИЕ ФУНКЦИИ В СИ

Заголовок

```
Тип ИмяФункции(СписокФормальныхПараметров)
{Описание данных
 Операторы
 return (выражение,возвращаемое функцией)
}
```

Блок

как в главной функции

тип возвращаемого значения; если отсутствует, то **int**;
если **void**, *то значение не возвращается, т. е. имеем аналог подпрограммы общего назначения, в этом случае* **return** *не нужен*

ОПИСАНИЕ ФУНКЦИИ В СИ

Заголовок

Тип ИмяФункции (СписокФормальныхПараметров)

Два смысла: 1) имя алгоритма,точнее - указатель на функцию, его значение - адрес точки входа в функцию ;
2) возвращаемое значение (имя функции можно использовать в выражениях).
Если имя функции **main**, *то это главная функция, она первой получает управление после запуска программы.* **main** *обязательно присутствует в программе. Пока рассматриваем main без параметров.*

ОПИСАНИЕ ФУНКЦИИ В СИ

- *В список формальных параметров включаются данные, которые передаются из вызывающей функции в данную и из данной функции в вызывающую (вход-выход функции).*
- *Замена формальных параметров на фактические - **только по значению**, поэтому у переменных - **результатов** функции в список формальных параметров следует включать **адрес**.*
- *Так как имя массива - указатель на его начало, в Си массивы можно передавать только по ссылке.*
- *У массивов - формальных параметров можно не указывать число значений первого индекса.*
- *Если список формальных параметров отсутствует или вместо него стоит слово* **void**, *то нет передачи значений в функцию.*

Примеры заголовков функций Си

1. Функция вычисления минимального значения среди элементов одномерного массива:
float min(float a[], int n) или float min(float *a, int n)

2. Функция решения квадратного уравнения $ax^2+bx+c=0$:

void kv_ur(float a, float b, float c, float *d, float *x1, float *x2)
d - дискриминант,
x1, x2 - корни уравнения, если d≥0, вещественная и мнимая части корней, если d<0.

3. Функция вычисления минимальных значений элементов строк матрицы из n строк и m (m≤5) столбцов:
void minmatr(float a[][5],int n,int m,float min[])

4. Функция вычисления минимальных значений элементов строк матрицы из n строк и m столбцов:
void minmatr(float **a, int n, int m, float *min)

Вызов (обращение к) функции

1. Как к процедуре общего назначения, используется оператор ИмяФункции (СписокФактическихПараметров);

2. Как к функции:указатель функции ИмяФункции (СписокФактПарам) -
используется в выражениях.

Для функции типа* void *допустим только способ 1.

Место описания функции в программе

1. Функция не может быть описана внутри другой функции.

2. Функция (как все в программе) должна быть описана до ее использования; если это не так, то до использования функции необходимо поместить прототип (шаблон) функции.

Шаблон: заголовок функции, в котором могут отсутствовать имена формальных параметров.

```
Тип ИмяФункции(СписокФормальныхПараметров);
```

Наличие шаблона позволяет компилятору контролировать соответствие типа возвращаемого функцией значения и соответствие количества и типов формальных и фактических параметров.

Практика применения шаблонов

- **Шаблоны ставятся в начало программного файла.**
- **Часто употребляемые шаблоны записываются в заголовочный файл (*.h), который подключается к программе директивой include препроцессора.**

Формы директивы include:

#include <Спецификация файла> //поиск файла в
//стандартных директориях

#include “Спецификация файла”// поиск файла по
//маршруту, заданному спецификацией

Примеры

1. Написать функцию ***kv_ur*** решения квадратного уравнения *ax2+bx+c=0.* Результаты функции: дискриминант уравнения и либо два действительных корня, либо действительная и мнимая часть комплексно-сопряженных корней. Результаты должны передаваться по имени, поэтому в список формальных параметров включены их адреса. В главной функции вводятся значения коэффициентов уравнения, вызывается ***kv_ur*** и выводятся результаты.

Примеры

2. Вычислить полусумму минимальных значений двух одномерных массивов: a[5] и b[8].

Два раза повторяющийся с точностью до обозначений и констант алгоритм вычисления минимального значения одномерного массива оформим как функцию, назовем ее *min*. Ввод массивов также оформим как функцию.

Замечание. Промежуточные данные должны быть внутренними (локальными) данными функции. Пример: счетчик i.

Примеры

3. Даны две матрицы: a из 3-х строк и 5 столбцов (в дальнейшем будем условно записывать a[3*5]) и b[7*3]. Вычислить минимальные значения для каждой строки каждой из этих матриц.

Естественно, вычисление минимальных значений строк произвольной матрицы следует оформить как функцию (назовем ее *minmatr*), а затем применить ее два раза к конкретным (фактическим) матрицам a и b. Алгоритм ввода одинаков для матриц a и b, поэтому ввод также оформим как функцию (назовем ее *matrin*). Аналогично вывод массива результатов будет процедура *masout*.

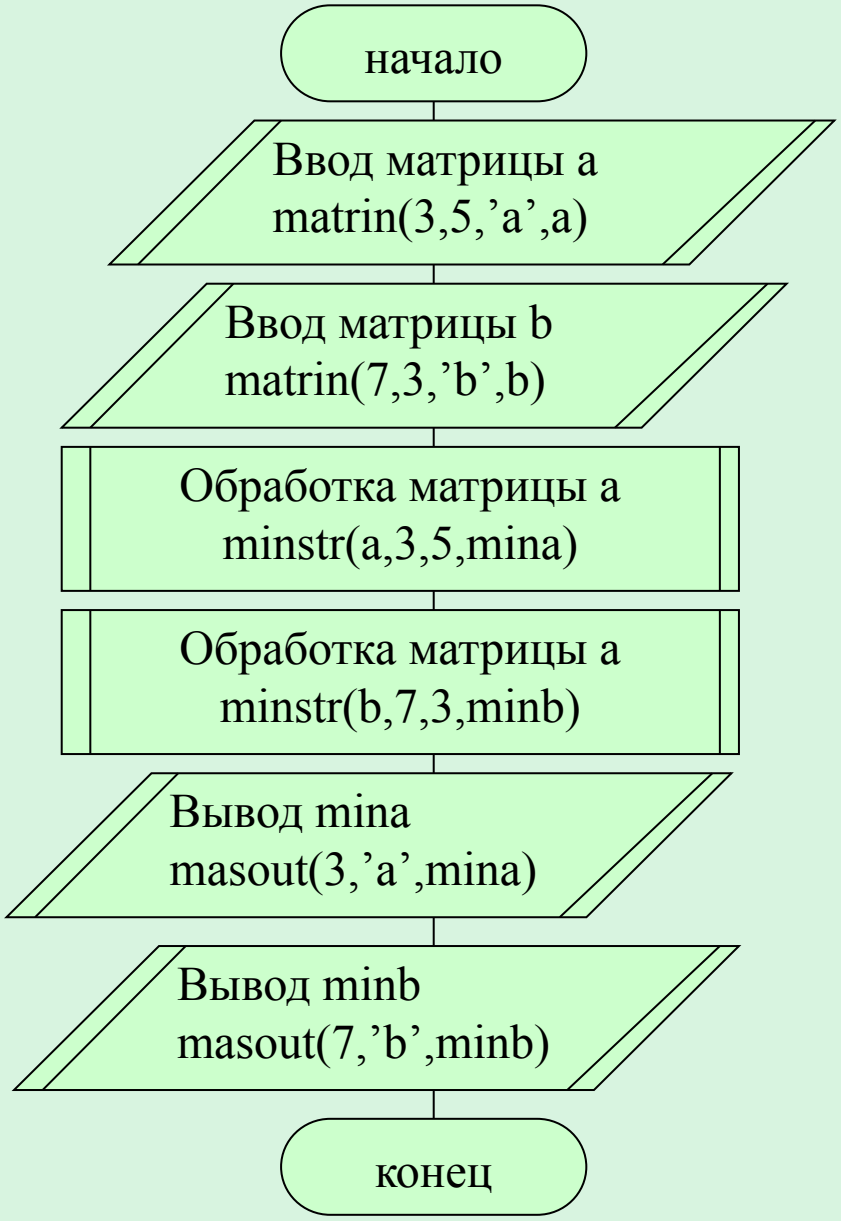
Прототипы функций

- void masout(float a[ ],int n,char c);
- void matrin(float [ ][5],int n,int m,char c);
- void minmatr(float [ ][5],int n,int m,float min[ ]);

Состав данных **программы**

Имя	**Смысл**	**Тип**	**Структура**
Исходные данные			
a	заданные матрицы	Вещественный	двумерный массив **с числом столбцов 5**
b			
Выходные данные			
mina	минимальные значения элементов строк матрицы **a(b)**	вещественный	одномерный массив
minb			

Блок-схема программы



Блок-схема функции определения минимальных значений строк матрицы