

Язык С

Особенности объявлений и некоторые операции

Программа – набор функций, одна из которых с именем `main` является главной.

Тело функции – блок, представляющий собой последовательность описаний и операторов в фигурных скобках `{...}`. Каждый оператор заканчивается точкой с запятой.

Комментарии могут быть приведены в любом месте программы. После символов `//` комментарием считаются все символы до конца строки. Если употребляются символы `/*...*/`, то комментарием считается всё, что заключено между этими символами. Например,

//Это однострочный комментарий

*/*И это тоже комментарий,*

но он может располагаться на нескольких строках/*

Пусть определена переменная типа указатель на величину некоторого типа, например:

```
int *ukaz;  
char *uk;
```

Значением переменной типа **указатель** является **адрес**. С помощью операции взятия содержимого по адресу (*ukaz) можно получить значение, хранящееся в той ячейке памяти, на которую в данный момент показывает ukaz (переменная типа указатель).

Каждая переменная размещается в памяти, начиная с некоторого байта. Операция &<имя переменной> определяет адрес этой переменной. Например,

```
          t      d      ud  
float t, d=1.5, *ud;
```



The diagram illustrates memory addresses t, d, and ud. Above the code, the labels t, d, and ud are positioned. Below, three boxes represent memory cells. The first box, labeled 't', contains the value '1.5'. The second box, labeled 'd', also contains '1.5'. The third box, labeled 'ud', is empty. An arrow points from the 'ud' box to the '1.5' in the 'd' box, indicating that the pointer variable 'ud' holds the address of 'd'.

Выполним присваивание

```
ud=&d;
```

Тогда одинаковый эффект дают следующие операторы:

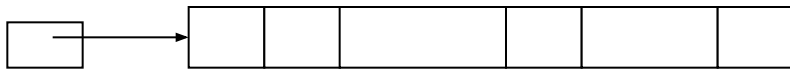
```
t=d;
```

```
t=*ud; //берется значение по адресу ud
```

Массивы и матрицы

Нумерация элементов массива начинается с нуля. Имя массива – указатель на его первый элемент, т.е. $\text{mas} == \&\text{mas}[0]$.

```
int mas[10];
```



mas 0 1... i ... 9

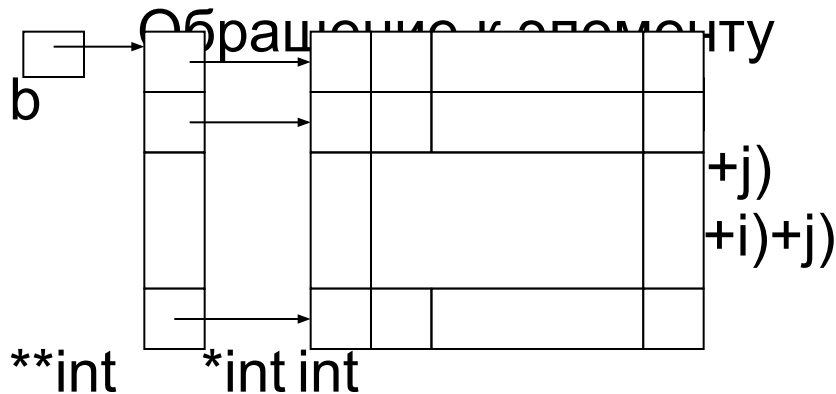
Значение i -ого элемента массива можно представить двумя способами: $\text{mas}[i]$; или *(mas+i) .

Для матрицы при объявлении указывается число строк и столбцов.

Нумерация элементов также начинается с нуля.

0 1 ... 19 int b[10][20].

0



Варианты ввода одномерного массива

```
int n, a[10], i, *uk;
```

Сначала вводится длина массива

```
printf ("Введите длину массива A: ");  
scanf("%d", &n);
```

Затем можно использовать следующие варианты ввода:

- `for (i=0; i<n; i++) scanf("%d", &a[i]);`
- `for (i=0; i<n; i++) scanf("%d", a+i);`
- `for (i=0; i<n; scanf("%d", &a[i++]));`
- `for (i=0; i<n; scanf("%d", a+i++));`
- `for (i=0, uk=a; i<n; i++) scanf("%d", uk++);`
- `for(uk=a; uk<a+n; uk++) scanf("%d", uk);`

Задача 1. Дан массив. Переставить его элементы так, чтобы сначала расположились все неотрицательные элементы, а затем отрицательные. Порядок среди отрицательных и неотрицательных элементов должен быть сохранён. Дополнительный массив не использовать.

Можно провести сортировку по знаку. Если положительный элемент (или 0) стоит правее отрицательного, то их следует переставить. Сравнение производится парами, как в методе "пузырька".

```
// вычислительная часть алгоритма
L:=n-1; //L- верхняя граница цикла
f:=1; //признак окончания перестановок
цикл пока f=1
  m:=L; f:=0;
  цикл от i:=0 до m-1
    если a[i]<0 и a[i+1]>=0 то
      b:=a[i]; a[i]:=a[i+1]; a[i+1]:=b;
      L:=i; f:=1;
  всё
кц
кц
```

```
#include <iostream> //для setlocale
#include <stdio.h> //стандартный ввод и вывод
int main()          //главная функция
{
    setlocale(LC_ALL, ".1251"); //меняем локализацию на русскую
    int a[50], n, b, i, L, m, f;    //описание переменных
    printf("Введите 0<n<=50 n = "); scanf("%d", &n);
    printf("Введите элементы массива\n");
    for (i=0; i<n; i++) scanf("%d", a+i);
    L=n-1; f=1;
    while (f)
    {
        m=L; f=0;
        for (i=0; i<m; i++)
            if (a[i]<0 && a[i+1]>=0)
                b=a[i], a[i]=a[i+1], a[i+1]=b, L=i, f=1;
    }
    printf("Преобразованный массив:\n");
    for (i=0; i<n; i++) printf("%7d", a[i]);
    printf("\n");
    return 0; //признак успешного завершения программы
}
```

Задача 2. Даны два одномерных массива целых чисел А и В. Сформировать массив С, состоящий из повторяющихся элементов массива А, отсутствующих в массиве В.

1. Есть новый массив

а

	1	2	1	2	2	3	4
b	1	3	0	-5			
c	2		2			2	

2. Нет нового массива

а

	1	2	3	3	3	4
b						

б

	1	2	3	4
--	---	---	---	---

Задача 2. Сформировать массив С, состоящий из повторяющихся элементов массива А, отсутствующих в массиве В.

```
#include <stdio.h>
#include <iostream>
using namespace std;
int main ()
{ setlocale(LC_ALL,"RUS");
int a[10],b[10],c[10],na,nb,nc=0,i,j;
printf ("введите длину массива А:");
scanf("%d",&na);
printf ("введите массив А\n");
for (i=0;i<na;i++)
    scanf("%d",&a[i]);
//ввод массива b аналогично
for(i=0;i<na;i++)
    { //проверка повторения a[i]
    for(j=0;j<na&&(a[i]!=a[j]||i==j);j++);
```

```
if (j<na) //повторяется
    { //проверка отсутствия a[i] в b
    for (j=0;j<nb&&a[i]!=b[j];j++);
    //цикл закрыт
    if (j==nb) //условие отсутствия
        c[nc++]=a[i];
    }
}
//вывод результата
if (nc==0)
    printf("массив С пуст\n");
else
    {
    printf("Массив С");
    for (i=0;i<nc;i++)
        printf("%7d",c[i]);
    printf ("\n");
    }
return 0;
}
```