

# ПОТОКОВЫЙ ВВОД/ВЫВОД

```
#include <iostream> //библиотека потокового ввода/вывода  
int a,b;
```

```
cout << "input b,a:" << endl;  
cin >> b >> a;  
cout << "a=" << a  
<< endl << "b=" << b;
```

```
input b,a:  
6  
8  
a=8  
b=6
```

```
printf("input b,a:\n");  
scanf("%d%d", &b, &a);  
printf("a=", a,  
"\nb=", b);
```

<< – операция записи в поток

>> – операция чтения из потока

cin – стандартный поток для ввода с клавиатуры

cout – стандартный поток для вывода на экран

endl – функция, включающая в поток символ конца строки  
(аналог "\n", но универсальный).

При форматном вводе (scanf) указываются **адреса** переменных. При потоковом вводе/выводе (cin/cout) и форматном выводе (printf) указываются **имена** переменных.

**Задача 3.** Сформировать массив С, состоящий из элементов массива А, повторяющихся в массиве В несколько раз. Элементы в С не должны повторяться.

```
#include <iostream>
using namespace std;
int main ()
{  setlocale(LC_ALL, "RUS");
  int a[10],b[10],c[10],na,nb,nc=0,i,j,k;
  cout << "Введите длину массива А ";
  cin >> na;
  cout << "Введите " << na << " элементов" << endl;
  for(i=0; i<na; i++)
  {
    cout << "Введите А[" << i << "]" << endl;
    cin >> *(a+i);
  }
  //Ввод массива b аналогично
  for(i=0;i<na;i++)
  {
    //проверка повторения a[i] в b несколько раз
    //k - число повторов a[i] в b
```

```

for(j=0,k=0;j<nb&&k<2;j++)
    if(a[i]==b[j]) k++;
    //k==0 – условие отсутствия a[i] в b
    //k==1 – a[i] присутствует в b 1 раз
if (k==2) //условие повторения
{
    //проверяем, что такого элемента еще нет в C
    for(j=0;j<nc&&a[i]!=c[j];j++);
    if (j==nc) //условие отсутствия a[i] в C
        //заносим этот элемент в C
        c[nc++]=a[i];
    }
}
if (!nc) cout << "Нет массива C" << endl;
else
{
    cout << "Массив C из " << nc << " элементов\n" << endl;
    for(j=0; j<nc; j++)
        cout << " " << *(c+j);
    }
return 0;
}

```

**Задача 4.** Дана последовательность целых чисел  $A[0:n-1]$ .  
Найти длину максимальной последовательности из нулей и  
начало этой последовательности.

Обозначения:

dtp – длина текущей последовательности из нулей

maxdp – максимальная длина последовательности

npmax – начало максимальной последовательности из нулей

nте - номер текущего элемента

*//Алгоритм – вычислительная часть*

maxdp:=0; dtp:=0;

цикл от nте:=0 до n-1

если a[nте]=0 то

dtp:=dtp+1;

иначе

если dtp>maxdp то

maxdp:=dtp;

npmax:=nте-dtp;

всё

всё

кц

если dtp>maxdp то

maxdp:=dtp;

npmax:=nте-dtp;

всё

```

#include <iostream>
using namespace std;
int main()
{
    setlocale(LC_ALL, "RUS");
    int a[100], n, *ua, dtp, maxdp, npmax;
    cout<<"Введите длину массива"<<endl;
    cin>>n;
    cout<<"Введите элементы"<<endl;
    for (ua=a; ua<a+n; ua++) cin>>*ua;
    maxdp=dtp=0;
    for (ua=a; ua<a+n; ua++)
        if (*ua==0) dtp++;
        else
        {
            if (dtp>maxdp) maxdp=dtp, npmax=(ua-a)-dtp;
            dtp=0;
        }
    if (dtp>maxdp) maxdp=dtp, npmax=(ua-a)-dtp;
    if (!maxdp) cout<<"Нет нулей";
    else
        cout<<"Maxdp="<<maxdp<<endl<<"npmax="<<npmax<<endl;
    return 0;
}

```

**Задача 4.** Упорядочить элементы матрицы по всем строкам (по возрастанию). Дополнительный массив не использовать.

**Метод:**

Рассматриваем матрицу как одномерный массив, содержащий  $n \cdot m$  элементов. В сортировке «пузырьком» сравниваем 2 соседних элемента:  $a[i][j]$  и следующий за ним  $a[i1][j1]$ . Индексы  $i1$  и  $j1$  вычисляются в зависимости от местоположения  $a[i][j]$  в матрице:

если  $j=m-1$  /\*  $a[i][j]$  находится в последнем столбце  $i$ -ой строки \*/

то  $i1=i+1$ ;  $j1=0$ ; /\* в качестве  $a[i1][j1]$  берётся первый элемент следующей строки \*/

иначе  $j1=j+1$ ;  $i1=i$ ; /\* в качестве  $a[i1][j1]$  берётся следующий элемент  $i$ -ой строки \*/

все

# Алгоритм решения задачи (вычислительная часть)

f:=1; m1:=n\*m-1;

цикл пока (F)

F:=0; m2:=m1; i:=0; j:=0;

цикл от k:=0 до m2-1

/\* определение пары сравниваемых элементов, т.е. вычисление индексов следующего элемента i1, j1; сравниваем a[i][j] с a[i1][j1] \*/

если j=m-1 то

i1:=i+1; j1:=0;

иначе

j1:=j+1; i1:=i;

всё

если a[i][j]>a[i1][j1] то

b:=a[i][j]; a[i][j]:=a[i1][j1]; a[i1][j1]:=b;

F:=1; m1:=k;

всё

i:=i1; j:=j1;

кц

кц

**Домашнее задание.** Закодировать алгоритм последней задачи. Форматный ввод-вывод, обращение к элементам матрицы при помощи указателей.