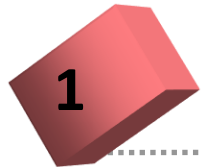




МАССИВЫ И ВЕКТОРЫ

**Дальневосточный государственный университет путей сообщения,
кафедра «Вычислительная техника и компьютерная графика»,
к.т.н., доцент Белозеров Олег Иванович.**

Вопросы лекции:



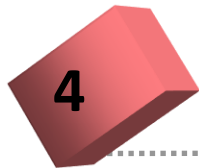
Одномерные массивы



Двумерные массивы



Векторы



Примеры решения задач

1. Одномерные массивы

Определение одномерного массива

Массив – это фиксированный набор объектов одного типа, который обозначается одним групповым именем.

Синтаксис:

тип_элементов имя_массива [размерность];

`int m[10];` // m – массив из 10 целочисленных элементов

Нумерация элементов массива в C++
начинается с нулевого элемента, то
есть, если массив состоит из 10
элементов, то его элементы будут
иметь следующие номера: 0, 1, 2, 3, 4,
5, 6, 7, 8, 9.

Элементы массива могут быть любого
типа, в том числе и структурированного
(кроме файлового).

Замечания:

1. Тип элементов массива и их количество фиксируется в определении массива. Количество элементов называют **размерностью** массива. Размерность массива должна быть константой целого типа.

```
int n=10;
```

```
int m[n]; // ошибка: размерность массива не может быть представлена  
переменной величиной
```

2. Размерность массива можно задавать с помощью переменной целого типа, определённого ключевым словом `const`.

```
const int n=10;
```

```
int m[n];
```

3. В памяти элементы массива располагаются вплотную друг к другу в порядке возрастания индекса.

```
int m[10]; // m – массив из 10 целочисленных элементов
```

4. Элементы массива пронумерованы. Доступ к отдельным элементам осуществляется посредством группового имени и порядкового номера, который называется индексом.

`m[3]=1;`

`m[4]=6;`

5. К элементам массива применимы все операции, совместимые с типом, фиксированным в его определении.

`int i=1;`

`m[i]=m[2*2]+m[2*i+1]; // m[1]=m[4]+m[3]=?`

6. Нумерация элементов массива начинается с **нуля**.

`M[0]=6; // Первый элемент`

`M[9]=7; // Последний элемент`

`M[10]=5; // Ошибка: элемента с таким номером не существует`

7. В C++ для встроенных массивов контроля невыхода за границы массива нет, сам программист должен следить за этим.

Инициализация одномерного массива

Массивы можно *инициализировать* при определении, указывая в фигурных скобках список значений элементов.

Пример (массив с количеством дней в каждом месяце):

```
int days[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
```

Циклы работы с массивами

```
const int n=20;    // размерность массива  
int m[n];          // объявление массива
```

Цикл ввода элементов в массив

```
for(int i = 0; i < n; i++)  
{  
    cout<<"m["<<i<<""]=";  
    cin>>m[i];  
}
```

Цикл вывода элементов из массива

```
for(int i = 0; i < n; i++)  
    cout<<"m["<<i<<""]="<<m[i];
```

Цикл прохода по элементам массива

```
for(int i = 0; i < n; i++)  
{  
    Выражения обработки элементов массива;  
}
```


Программа Поиск максимального и минимального элементов массива и вычисления суммы его элементов

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main() {
    setlocale(LC_ALL, "Russian");
    int x[100];      // массив
    int n;           // Размер массива
    int min, max, sum;
    cout << "Введите количество элементов: \n";
    cin >> n;
    for(int i = 0; i < n; i++)
    {   cout << "x[" << i << "]=";
        cin >> x[i];   }
    cout << "Задан массив: \n";
    for(int i = 0; i < n; i++) // Вывод массива
    cout << x[i] << " ";
    sum=0;
    max=min=x[0];

    // Цикл прохода по массиву элементов
    for(int i = 0; i < n ; i++)
    {
        if(x[i] > max)
            max=x[i];
        else if (x[i] < min)
            min=x[i];
        sum+=x[i];
    }
    cout << "\nМаксимальный элемент: " << max;
    cout << "\nМинимальный элемент: " << min;
    cout << "\nСумма элементов: " << sum;
    system("pause");
    return 0;
}
```

Введите количество элементов:

5

x[0]=1

x[1]=2

x[2]=3

x[3]=4

x[4]=5

Задан массив:

1 2 3 4 5

Максимальный элемент: 5

Минимальный элемент: 1

Сумма элементов: 15Для продолжения нажмите любую клавишу . . .

Программа Удаление отрицательных элементов.

```
#include <iostream>
using namespace std;
int main() {
    setlocale(LC_ALL, "Russian");
    int x[100];      // массив
    int n;           // Размер массива
    cout << "Введите количество элементов: \n";
    cin >> n;
    for(int i = 0; i < n; i++) {
        cout << "x[" << i << "]=";
        cin >> x[i]; }
    cout << "Задан массив: \n";
    for(int i = 0; i < n; i++) // Вывод массива
        cout << x[i] << " ";
    cout << endl;
    // Проход по элементам массива и поиск
    // отрицательных
    for( int i=0;i<n; i++)
        if (x[i]<0) { // если отрицательный элемент
            // сдвигаем все элементы за ним на
            // одну позицию влево
```

```
        // Цикл сдвига элементов
        for (int j = i+1; j < n ; j++)
            x[j-1]=x[j];
        n--; // уменьшение размерности массива
        i--;
    }
    // вывод преобразованного массива
    for(int i = 0; i < n; i++) // Вывод массива
        cout << x[i] << " ";
    system("pause");
    return 0;
}
```

Введите количество элементов:

3

$x[0] = -1$

$x[1] = 2$

$x[2] = 3$

Задан массив:

-1 2 3

2 3 Для продолжения нажмите любую клавишу . . .

Программа Вставка элемента в заданную позицию.

```
#include <iostream>
using namespace std;
int main()
{   setlocale(LC_ALL, "Russian");
int x[100];    // Массив
int n;        // Размер массива
int el, poz;   // Элемент и номер позиции
cout << "Введите количество элементов: \n";
cin>>n;
for(int i = 0; i < n; i++)
{   cout<<"x["<<i<<"]=";
    cin>>x[i]; }
cout << "Задан массив: \n";
for(int i = 0; i < n; i++) // Вывод массива
cout << x[i] << " ";
cout<<endl;
cout << "Введите элемент: \n";
cin>>el;
cout << "Введите номер позиции: \n";
cin>>poz;
```

```
// Цикл сдвига элементов справа налево
for (int j = n-1; j >=poz ; j--)
x[j+1]=x[j];
x[poz]=el;
n++;
// вывод преобразованного массива
for(int i = 0; i < n; i++) // Вывод массива
cout << x[i] << " ";
system("pause");
return 0;
}
```

Введите количество элементов:

5

x[0]=1

x[1]=2

x[2]=3

x[3]=4

x[4]=5

Задан массив:

1 2 3 4 5

Введите элемент:

3

Введите номер позиции:

5

1 2 3 4 5 3 Для продолжения нажмите любую клавишу . . .

Введите количество элементов:

3

x[0]=1

x[1]=2

x[2]=3

Задан массив:

1 2 3

Введите элемент:

0

Введите номер позиции:

0

0 1 2 3 Для продолжения нажмите любую клавишу . . .

Программа Сортировка массива по возрастанию.

```
#include <iostream>
using namespace std;
int main() {
    setlocale(LC_ALL, "Russian");
    int x[100];      // Массив
    int n;           // Размер массива
    cout << "Введите количество элементов: \n";
    cin >> n;
    for(int i = 0; i < n; i++) {
        cout << "x[" << i << "]=";
        cin >> x[i]; }
    cout << "Исходный массив: \n";
    for(int i = 0; i < n; i++) // Вывод массива
        cout << x[i] << " ";
    cout << endl;

    // Сортировка
    for (int i=0; i<n-1; i++)
        for (int j=i+1; j<n; j++)
            if (x[i]>x[j])
            {
                int tmp=x[i];
                x[i]=x[j];
                x[j]= tmp;
            }
    // вывод отсортированного массива
    for(int i = 0; i < n; i++) // Вывод массива
        cout << x[i] << " ";
    system("pause");
    return 0;
}
```


Введите количество элементов:

5

x[0]=1

x[1]=2

x[2]=5

x[3]=4

x[4]=0

Исходный массив:

1 2 5 4 0

0 1 2 4 5 Для продолжения нажмите любую клавишу . . .

Программа Формирование массива без повторений

```
#include <iostream>
using namespace std;
int main()
{
    setlocale(LC_ALL, "Russian");
    int x[100];    // Исходный массив
    int y[100];    // Массив без повторений
    int n1;        // Размер исходного массива
    int n2;        // Размер полученного массива
    cout << "Введите количество элементов: \n";
    cin>>n1;
    for(int i = 0; i < n1; i++)
    {    cout<<"x["<<i<<"]="";
        cin>>x[i];    }
    cout << "Исходный массив: \n";
    for(int i = 0; i < n1; i++) // Вывод массива
    cout << x[i] << " ";
    cout<<endl;

    // Формирование массива
    n2=0;    // Вначале элементов нет
    for (int i=0; i<n1; i++) { // каждый элемент
        // исходного массива ищем в новом
        int j=0;
        while (j<n2 && x[i]!=y[j])
            j++;
        if (j==n2) // если элемент не найден
        {
            y[n2]=x[i]; // Добавляем новый элемент
            n2++;    // Увеличиваем количество
        }
    }
    // массив без повторений
    cout << "Массив без повторений: \n";
    for(int i = 0; i < n2; i++) // Вывод массива
    cout << y[i] << " ";
    system("pause");
    return 0;
}
```

Введите количество элементов:

5

x[0]=1

x[1]=2

x[2]=3

x[3]=4

x[4]=5

Исходный массив:

1 2 3 4 5

Массив без повторений:

1 2 3 4 5 Для продолжения нажмите любую клавишу . . . ■

Введите количество элементов:

5

x[0]=1

x[1]=1

x[2]=1

x[3]=2

x[4]=3

Исходный массив:

1 1 1 2 3

Массив без повторений:

1 2 3 Для продолжения нажмите любую клавишу . . .

Дан массив из 10 целых чисел. Написать программу, которая заменяет в данном массиве все отрицательные элементы нулями.

Найдите ошибки в программе!!!

```
#include <iostream>
using namespace std;
int main()
{ int n; cout<<"n="; cin>>n;          //ввели количество элементов массива
  int a[n];      // объявляем статический массив размерностью n
  for (int i=0;i<n; ++i)      //ввод и обработка данных
  {cout<<"a["<<i<<"]="; cin>>a[ij);    //ввод очередного элемента
    if (a[i]<0) a[i]=0;} //проверка и замена
  for (int i=0;i<10;++i) cout<<a[i]<<"M"; //вывод массива на экран
  return 0;}
```

```
#include <iostream>
using namespace std;
int main()
{
    int a[100];
    int n;
    cout << "n=";
    cin >> n;    //ввели количество элементов массива
    for (int i = 0; i < n; i++)//ввод и обработка данных
    {
        cout << "a["<<i<<"] = ";
        cin>>a[i];    //ввод очередного элемента

        if (a[i] < 0)
            a[i] = 0;
    }    //проверка и замена
    for (int i = 0; i < n; i++)
        cout << a[i]; //вывод массива на экран
    return 0;
}
```

n=4

a[0] = -1

a[1] = -2

a[2] = 3

a[3] = 4

0034

C:\Users\lenovo\Desktop\Белозеров\ДВГУПС\!Дисциплины\Програ

Чтобы автоматически закрывать консоль при остановке отлад

Нажмите любую клавишу, чтобы закрыть это окно:

Дан массив из n действительных чисел ($n < 100$).
Написать программу для подсчета суммы этих чисел.
Найдите ошибки в программе!

```
#include <iostream>
using namespace std;
int main()
{ int n; cout<<"n="; cin>>n;
float a[n];
float s=0;
for (int i=0;i<n; ++i)
{cout<<"a["<<i<<"]="; cin>>a[i]; //ввод очередного элемента в массив
s+=a[i];} //добавление значения элемента массива к сумме
cout <<"s="<<s<<endl;
return 0;}
```



```
#include <iostream>
using namespace std;
int main()
{
    float a[100];
    int n;
    cout << "n=";
    cin >> n;

    float s = 0;
    for (int i = 0; i < n; ++i)
    {
        cout << "a["<<i<<"] = ";
        cin>>a[i];    //ввод очередного элемента в массив
        s += a[i];
    }//добавление значения элемента массива к сумме
    cout << "s=" <<s<< endl;
    return 0;
}
```

n=5

a[0] = 1

a[1] = 2

a[2] = 3

a[3] = 4

a[4] = 5

s=15

C:\Users\lenovo\Desktop\Белозеров\ДВГУПС\!Дисциплины\Программирование

Чтобы автоматически закрывать консоль при остановке отладки, включите

Нажмите любую клавишу, чтобы закрыть это окно:

Дан массив из n целых чисел ($n < 100$). Написать программу для подсчета среднего арифметического четных значений данного массива.

Найдите ошибки в программе!

```
#include <iosfream>
using namespace std;
int main()
{ int n; cout<<"n="; cin>>n;
  int a[n], k=0;
  float s=0;
  for (int i=0;i<n; ++i)
    { cout<<"a["<<i<<"]="; cin>>a[i]; //ввод очередного элемента в массив
      if (!(a[i]%2)) //если остаток при делении элемента на 2 равен 0
        {s+=a[i];++k;} } //то элемент четный - добавить его к сумме и увеличить
// количество четных элементов на 1
    if (k) //если k не нулевое, то четные числа в последовательности есть
      cout <<"sr="<< s/k<<endl; //и можно вычислить их среднее арифметическое
      значение
    else
      cout<<" четных чисел в последовательности нет "<<endl;
  return 0;}
```

```
#include <iostream>
using namespace std;
int main()
{
    setlocale(LC_ALL, "Russian");
    int a[100]; int n; int k = 0; float s = 0;
    cout << "n=";
    cin >> n;
    for (int i = 0; i < n; ++i)
    {
        cout << "a[" << i << "] = ";
        cin >> a[i];          //ввод очередного элемента в массив
        if (!(a[i] % 2)) //если остаток при делении элемента на 2 равен 0,
        {s+=a[i];++k;} } //то элемент четный – добавить его к сумме и
//увеличить количество четных элементов на 1
    } if (k)//если k не нулевое, то четные числа в последовательности
//есть
        cout << "sr=" << s / k << endl;
    else
        cout << " четных чисел в последовательности нет "<<endl;
    return 0;
}
```

n=5

a[0] = 1

a[1] = 2

a[2] = 3

a[3] = 4

a[4] = 5

sr=3

C:\Users\Ienovo\Desktop\Белозеров\ДВГУПС\!Дисциплины\Программирова

Чтобы автоматически закрывать консоль при остановке отладки, включ

Нажмите любую клавишу, чтобы закрыть это окно:

n=0

четных чисел в последовательности нет

C:\Users\Ienovo\Desktop\Белозеров\ДВГУПС\!Дисциплины\Программиро

Нажмите любую клавишу, чтобы закрыть это окно:

```
n=5  
a[0] = 1  
a[1] = 1  
a[2] = 1  
a[3] = 1  
a[4] = 1  
четных чисел в последовательности нет
```

C:\Users\Ienovo\Desktop\Белозеров\ДВГУПС\!Дисциплины\Программирование-2

Нажмите любую клавишу, чтобы закрыть это окно:

Дан массив из n целых чисел ($n < 100$). Написать программу, которая определяет наименьший элемент в массиве и его порядковый номер.

Найдите ошибки в программе!

```
#include <iostream>
using namespace std;
int main()
{ int n; cout<<"n="; cin>>n;
  int a[n];
  for (int i=0;i<n; ++i) { cout<<"a["<<i<<"]="; cin>>a[i];}
  int min=a[0]; //в качестве наименьш. значения полагаем нулевой элемент массива
  int nmin=0; //соответственно его порядковый номер равен 0
  for (int i=1;i<n; ++i) //перебираем все элементы массива с первого по последний
    if (a[i]<min) //если очередной элемент окажется меньше значения min то в
      //качестве
      {min=a[i]; //нового наименьш. значения запоминаем значение текущего
      //элемента
      nmin=i;} //массива и, соответственно, запоминаем его номер
  cout <<"min="<< min<<"\t nmin=" << nmin<<endl;
  return 0;}
```



```
#include <iostream>
using namespace std;
int main()
{
    int a[100]; int n; cout << "n="; cin >> n;
    for (int i = 0; i < n; ++i) {
        cout << "a["<<i<<" ] = "; cin>>a[i];}
        int min = a[0];    //в качестве наименьш. значения полагаем
//нулевой элемент массива
        int nmin = 0;      //соответственно его порядковый номер равен 0
        for (int i = 1; i < n; ++i)    //перебираем все элементы
//массива с первого по последний
            if (a[i] < min) //если очередной элемент окажется меньше
//значения min то в качестве
            {
                min = a[i];    //нового наименьш. значения запоминаем
//значение текущего элемента
                nmin = i;
            } //массива и, соответственно, запоминаем его номер
        cout << "min="<< min<<"\t nmin = " << nmin << endl;
    return 0;
}
```

n=5

a[0] = 5

a[1] = 4

a[2] = 1

a[3] = 2

a[4] = 3

min=1 nmin = 2

C:\Users\lenovo\Desktop\Белозеров\ДВГУПС\!Дисциплины\Программирование-2 sem

Чтобы автоматически закрывать консоль при остановке отладки, включите параметр

Нажмите любую клавишу, чтобы закрыть это окно:

Дан массив из n действительных чисел ($n < 100$). Написать программу, которая меняет местами в этом массиве наибольший и наименьший элемент местами (считается, что в последовательности только один наибольший и один наименьший элементы).

Найдите ошибки в программе!

```
#include <iostream>
using namespace std;
int main()
{ int n; cout<<"n="; cin>>n;
float a[n];
for (int i=0;i<n; ++i) {cout<<"a["<<i<<"]="; cin>>a[i];}
//первоначально полагаем элемент с номером 0 минимальным и максимальным
float min=a[0], max=a[0];
int nmin=0, nmax=0;
for (int i=1 ;i<n; ++i) //поиск наибольшего и наименьшего значения в массиве и их номеров
{ if (a[i]<min){min=a[i];nmin=i;}
if(a[i]>max){max=a[i];nmax=i;}}
a[nmax]=min; //в позицию наименьшего элемента записываем значение наибольшего
a[nmin]=max; //в позицию наибольшего элемента записываем значение наименьшего
for (int i=0;i<n; ++i) cout<<a[i]<<"\t"; //выводим измененный массив на экран
return 0;}
```

```
#include <iostream>
using namespace std;
int main()
{
float a[100];int n; cout << "n="; cin >> n;
for (int i = 0; i < n; ++i)
{
cout << "a["<<i<<"] = "; cin >> a[i];
}
//первоначально полагаем элемент с номером 0 минимальным и максимальным
float min = a[0], max = a[0];
int nmin = 0, nmax = 0;
for (int i = 1; i < n; ++i) //поиск наибольшего и наименьшего значения в массиве и
//их номеров
{ if (a[i]<min){min=a[i];nmin=i;}
if (a[i] > max) {
max = a[i]; nmax = i; }
}
a[nmax] = min; //в позицию наименьшего элемента записываем значение наибольшего
a[nmin] = max; //в позицию наибольшего элемента записываем значение наименьшего
for (int i = 0; i < n; ++i) cout << a[i] << "\t"; //выводим измененный массив на
//экран
return 0;
}
```

n=5

a[0] = 1

a[1] = 2

a[2] = 3

a[3] = 4

a[4] = 5

5 2 3 4 1

C:\Users\Ienovo\Desktop\Белозеров\ДВГУПС\!Дисциплины\Программирован

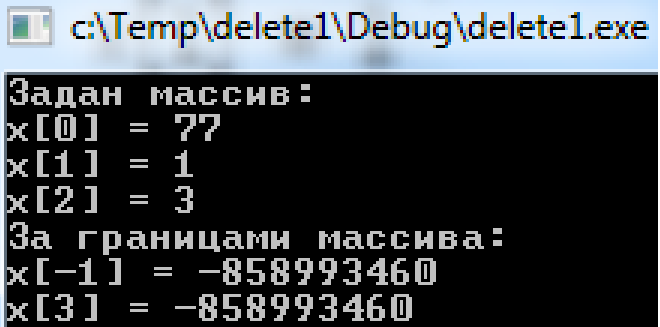
Чтобы автоматически закрывать консоль при остановке отладки, включи

Нажмите любую клавишу, чтобы закрыть это окно:

Проблемы встроенных массивов

Программа Выход за границы массива

```
#include <iostream>
#include <locale>
#include <cstdlib>
using namespace std;
int main()
{
    setlocale(LC_ALL, "Russian");
    int x[] = {77, 1, 3}; // Массив
    int n = sizeof(x) / sizeof(int); // Размер массива
    cout << "Задан массив: \n";
    for(int i = 0; i < n; i++) // Вывод массива
        cout << "x[" << i << "] = " << x[i] << endl;
    cout << "За границами массива:\n";
    cout << "x[-1] = " << x[-1] << endl; // -1 - ошибочный индекс
    cout << "x[3] = " << x[3] << endl; // 3 - ошибочный индекс
    system("pause");
    return 0;
}
```



```
c:\Temp\delete1\Debug\delete1.exe
Задан массив:
x[0] = 77
x[1] = 1
x[2] = 3
За границами массива:
x[-1] = -858993460
x[3] = -858993460
```

2. Двумерные массивы

Двумерные массивы (матрицы, таблицы) - представляют собой фиксированное количество элементов одного и того же типа, объединенных общим именем, где каждый элемент определяется номером строки и номером столбца, на пересечении которых он находится.

Нумерация строк и столбцов начинается с нулевого номера. Поэтому если массив содержит три строки и четыре столбца, то строки нумеруются: 0, 1, 2; а столбцы: 0, 1, 2, 3.

В C++ двумерный массив реализуется как одномерный, каждый элемент которого также массив.

Синтаксис:

тип имя[размерность1-число строк][размерность2-число столбцов];

Пример:

```
int m[2][3];           //двумерный массив из 2 строк и 3 столбцов,  
                        // элементы массива целые числа.
```

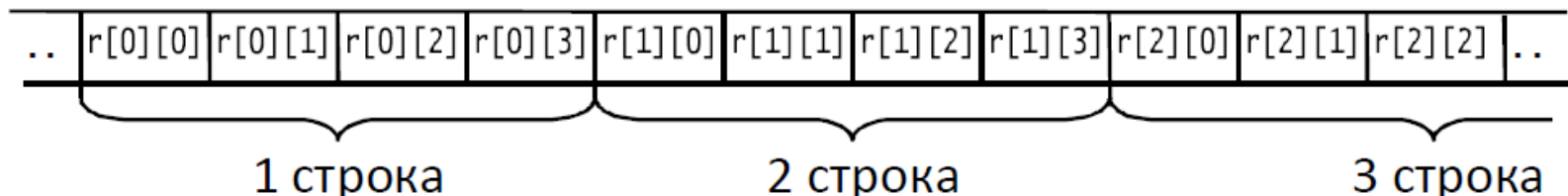
Доступ к элементам массива осуществляется по двум индексам.

Пример:

```
m[i][j]=1;           // i – номер строки, j – номер столбца.
```

Замечания:

1) В оперативной памяти элементы массива располагаются по строкам.



2) Нумерация индексов также начинается с нуля.

Инициализация двумерных массивов

При определении двумерный массив можно инициализировать с помощью списка констант, заключенных в фигурные скобки.

Пример:

```
int m[2][3]={{1,2,3},{4,5,6}};
```

Замечания:

- 1) Заполнение элементов идет по строкам
- 2) Фигурные скобки, отделяющие строки, могут быть опущены.
- 3) Список инициализаторов может быть неполным. В этом случае последние элементы инициализируются нулями.
- 4) При работе с многомерными массивами используются вложенные циклы.

Пример:

```
int m[2][3]={1,2,3,4};
```

$m[0][0]=1$, $m[0][1]=2$, $m[0][2]=3$, $m[1][0]=4$, $m[1][1]=0$, $m[1][2]=0$.

Программа сложения матриц

```
#include <iostream>

#include <locale>

#include <cstdlib>

using namespace std;

int main()
{   setlocale(LC_ALL, "Russian");

    int a[2][3], b[2][3], c[2][3];

    int i, j;

    // ввод матрицы a
    for (i=0; i<2; i++) // проход по строкам
    for (j=0; j<3; j++) // проход по столбцам
    {   cout<<"a["<<i+1<<"]["<<j+1<<"]="";
        cin>>a[i][j];   }

    // ввод матрицы b
    for (i=0; i<2; i++) // проход по строкам
    for (j=0; j<3; j++) // проход по столбцам
    {   cout<<"b["<<i+1<<"]["<<j+1<<"]="";
        cin>>b[i][j];   }
```

```
    // суммирование матриц
    for (i=0; i<2; i++) // проход по строкам
    for (j=0; j<3; j++) // проход по столбцам
    c[i][j]=a[i][j]+b[i][j];
    // вывод матрицы a
    cout<<"a"<<endl;
    for (i=0; i<2; i++) // проход по строкам
    {   for (j=0; j<3; j++) // проход по столбцам
        cout<<a[i][j]<<" ";
        cout<<endl; } // выделение строки
    // вывод матрицы b
    cout<<"b"<<endl;
    for (i=0; i<2; i++) // проход по строкам
    {   for (j=0; j<3; j++) // проход по столбцам
        cout<<b[i][j]<<" ";
        cout<<endl; } // выделение строки
    // вывод матрицы c
    cout<<"a+c"<<endl;
    for (i=0; i<2; i++) // проход по строкам
    {   for (j=0; j<3; j++) // проход по столбцам
        cout<<c[i][j]<<" ";
        cout<<endl; } // выделение строки
    system("pause"); return 0; }
```

```
a[1][1]=2
a[1][2]=2
a[1][3]=2
a[2][1]=2
a[2][2]=3
a[2][3]=4
b[1][1]=3
b[1][2]=2
b[1][3]=1
b[2][1]=2
b[2][2]=3
b[2][3]=4
```

```
a
2    2    2
2    3    4
```

```
b
3    2    1
2    3    4
```

```
a+c
5    4    3
4    6    8
```

Для продолжения нажмите любую клавишу . . .

Задача о предпринимателе

Пусть некий предприниматель имеет *три* магазина, по которым ведет ежедневный учет выручки от продажи *четырёх* видов продуктов (например, чая, сахара, крупы, колбасы). Для учета продаж можно использовать следующий двумерный массив:

double r[3][4];

Например:

```
double r[3][4] = {{500, 700, 450, 1000},  
                  {600, 710, 480, 1100},  
                  {800, 750, 550, 1200}};
```

Пусть предприниматель, о котором речь шла выше, каждый день подводит итоги: вычисляет выручку каждого магазина, выручку от продажи каждого товара и общую выручку за день. Эти расчеты можно выполнить с помощью следующей программы.

Программа о предпринимателе

```
#include <iostream>
#include <locale>
#include <cstdlib>
using namespace std;

int main() {
    setlocale(LC_ALL, "Russian");

    const int NSHOP = 3; // Число магазинов
    const int NGOODS = 4; // Число товаров
    double r[NSHOP][NGOODS]; // Двумерный
                             //массив для выручки

    double srshop, srg, sum; // Выручка по
                             //магазинам, товарам, общая

    cout << "Введите выручку от чая, сахара, крупы, колбасы \n";
    // Ввод данных

    for(int i = 0; i < NSHOP; i++){ // Перебор магазинов
        cout << "Магазин " << i + 1 << ": ";
        for(int j = 0; j < NGOODS; j++) // Перебор товаров
```

```
            cin >> r[i][j]; } // Ввод выручки i-го магазина
                             // от продажи j-го товара
        cout << "Выручка по магазинам: \n";
        for(int i = 0; i < NSHOP; i++){ // Перебор
                                         //магазинов

            srshop = 0;
            for(int j = 0; j < NGOODS; j++) // Перебор
                                             //товаров

                srshop += r[i][j];
            cout << srshop << endl; }
        sum = 0;
        cout << "Выручка по товарам: \n";
        for(int j = 0; j < NGOODS; j++){ // Перебор
                                         //товаров

            srg = 0;
            for(int i = 0; i < NSHOP; i++) // Перебор
                                             //магазинов

                srg += r[i][j];
            cout << srg << " ";
            sum += srg; } // Подсчет общей выручки
        cout << "\nВсего продано за день на: " << sum
        << endl;
        system("pause");
        return 0;}
```

Введите выручку от чая, сахара, крупы, колбасы

Магазин 1: 1000

2000

3000

234

Магазин 2: 234

1234

3245

54

Магазин 3: 456

3456

432

12

Выручка по магазинам:

6234

4767

4356

Выручка по товарам:

1690 6690 6677 300

Всего продано за день на: 15357

Для продолжения нажмите любую клавишу . . . █

3. Векторы

Более безопасным типом данных, чем массивы являются **вектора**.

Для того, чтобы использовать вектора надо включить заголовочный файл vector:

```
#include <vector>
```

Пример:

```
vector<int> v(3);    // создает вектор из 3-х элементов типа int.
```

Работа с векторами во многом похоже на работу с массивами:

- 1) обращаться к элементам вектора можно так же как к элементам массива — с помощью индекса, заключенного в квадратные скобки.
- 2) Нумерация элементов вектора начинается с нуля.

Особенность:

При выходе индекса за границы возникает ошибка.

Замечание:

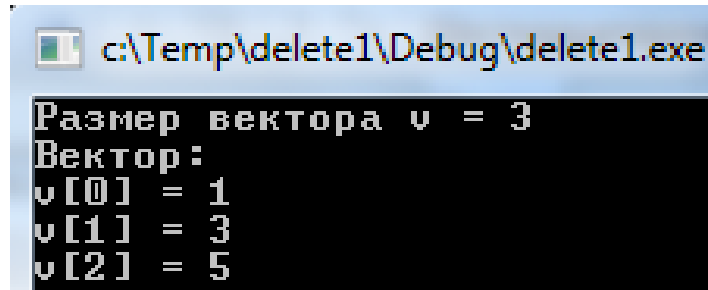
Функция size() возвращает текущий размер вектора.

Пример:

```
cout << v.size();    // выведет 3
```

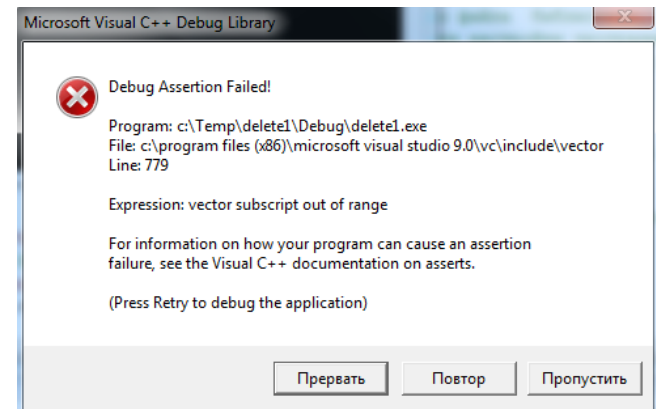
Программа Вектора

```
#include <iostream>
#include <locale>
#include <cstdlib>
#include <vector>
using namespace std;
int main()
{ setlocale(LC_ALL, "Russian");
  vector<int> v(3); // Вектор из 3-х элементов
  cout << "Размер вектора v = " << v.size() << endl;
  for(int i = 0; i < v.size(); i++) // Заполнение вектора
    v[i] = 2 * i + 1;
  cout << "Вектор:\n";
  for(int i = 0; i < v.size(); i++) // Вывод вектора
    cout << "v[" << i << "] = " << v[i] << endl;
  //cout << "v[-1] = " << v[-1] << endl; // -1 - недопустимый индекс
  //cout << "v[3] = " << v[3] << endl; // 3 - недопустимый индекс
  system("pause"); return 0;}
```



```
c:\Temp\delete1\Debug\delete1.exe
Размер вектора v = 3
Вектор:
v[0] = 1
v[1] = 3
v[2] = 5
```

Если убрать комментарии – сообщение об ошибке: vector subscript out of range (индекс вектора вне диапазона)



Возможности Векторов

Вектора можно создавать из элементов любых типов. Если при создании вектора не указывать число элементов, создается пустой вектор, не имеющий элементов.

```
vector<double> v; // Пустой вектор
```

```
v[0] = 3.5;      // Ошибка, вектор v не имеет элементов
```

1) Функция `push_back()` добавляет в конец вектора новый элемент.

```
v.push_back(3.5);
```

```
v.push_back(7.9);
```

Теперь вектор `v` содержит элемент `v[0]` со значение 3.5 и элемент `v[1]` со значение 7.9.

2) Функция `pop_back()` удаляет из конца вектора элемент.

```
v.pop_back();
```

Теперь вектор `v` содержит только элемент `v[0]` со значение 3.5.

3) Текущий размер вектора можно изменить функцией `resize()`. Новые элементы заполняются нулями.

```
v.resize(5);    // Теперь размер v равен 5
```

4) Вектора можно присваивать, подобно одиночным переменным, чего нельзя делать массивами.

Программа Размеры и копирование векторов

```
#include <iostream>
#include <locale>
#include <cstdlib>
#include <vector>
using namespace std;
int main()
{ srand(time(0)); // Инициализация
    //генератора случайных чисел
    setlocale(LC_ALL, "Russian");
    const double Max = 1000.0;
    vector<double> v; // Пустой вектор
    v.push_back(3.5); // Добавление в вектор
    v.push_back(7.9); // элементов
    cout << "Вектор v размера " << v.size() << endl;
    for(int i = 0; i < v.size(); i++) // Вывод
        cout << v[i] << " "; // вектора
    v.resize(5); // Теперь размер v равен 5
    cout << "\nТеперь вектор v имеет размер ";
    cout << v.size() << endl;

    for(int i = 0; i < v.size(); i++) // Вывод
        cout << v[i] << " "; // вектора
    vector<double> w; // Еще один пустой вектор
    cout << "\nРазмер вектора w " << w.size() <<
        endl;
    w = v; // Присваивание векторов
    cout << "Теперь вектор w - копия вектора
        v\n";
    for(int i = 0; i < w.size(); i++)
        cout << w[i] << " ";
    for(int i = 0; i < v.size(); i++) // Заполнение
        //вектора v
        v[i] = rand() / Max; // случайными числами
    cout << "\nВектор v заполнен случайными
        числами\n";
    for(int i = 0; i < v.size(); i++)
        cout << v[i] << " ";
    cout << endl;
    system("pause");
    return 0;
}
```

Вектор v размера 2

3.5 7.9

Теперь вектор v имеет размер 5

3.5 7.9 0 0 0

Размер вектора w 0

Теперь вектор w - копия вектора v

3.5 7.9 0 0 0

Вектор v заполнен случайными числами

31.926 24.004 31.327 20.011 23.321

Для продолжения нажмите любую клавишу . . .

Генерация (псевдо)случайных чисел

Функция стандартной библиотеки:

- **int rand (void);**
- Она генерирует псевдослучайное целое число на интервале значений от 0 до RAND_MAX. Последнее является константой, которая варьируется в зависимости от реализации языка, но в большинстве случаев составляет 32767.

Генерация чисел от 0 до 9:

- **rand() % 10**
- Если нам нужны числа от 1 (а не от 0) до 9, то можно прибавить 1:
- **rand() % 9 + 1**
- Идея такая: генерируем случайное число от 0 до 8, и после прибавления 1 оно превращается в случайное число от 1 до 9.

- Функция `rand()` генерирует псевдослучайные числа, т.е. числа, которые кажутся случайными, но на самом деле являются последовательностью значений, вычисленных по хитрому алгоритму, в качестве параметра принимающему так называемое зерно (`seed`).
`void srand (unsigned int seed);`
- Она устанавливает зерно равным значению параметра, с которым была вызвана. И последовательность чисел тоже будет другая.

Чтобы сделать зерно всегда разным, используем функцию `time()`.

`time_t time (time_t* timer);`

Она тоже досталась в наследство от языка Си и, будучи вызвана с нулевым указателем в качестве параметра, возвращает количество секунд, прошедших с 1 января 1970 года. Теперь значение этой функции мы можем передать в функцию `srand()` и будет случайное зерно. И числа будут неповторяющиеся.

Для использования функций **`rand()`** и **`srand()`** нужно подключить заголовочный файл **`<cstdlib>`**, а для использования `time()` – файл **`<ctime>`**.

Программа Удаление отрицательных элементов

```
for(int i = 0; i < v.size(); i++)      // Вывод
cout << v[i] << " ";                  // вектора
vector<double>w; // Еще один пустой вектор
cout << "\nРазмер вектора w " << w.size() << endl;
w = v; // Присваивание векторов
cout << "Теперь вектор w - копия вектора v\n";
for(int i = 0; i < w.size(); i++)
cout << w[i] << " ";
for(int i = 0; i < v.size(); i++) // Заполнение
                                //вектора v
v[i] = rand() / Max; // случайными числами
cout << "\nВектор v заполнен случайными числами\n";
for(int i = 0; i < v.size(); i++)
cout << v[i] << " ";
cout << endl;
system("pause");
return 0;
}

// Проход по элементам вектора и поиск
// отрицательных
for( int i=0;i<v.size(); i++)
if (v[i]<0) // если отрицательный элемент
{ // сдвигаем все элементы за ним на
  // одну позицию влево
  // Цикл сдвига элементов
  for (int j = i+1; j < v.size() ; j++)
  v[j-1]=v[j];
  v.pop_back(); // уменьшение размерности
                //вектора
  i--; // перепроверка элемента на этом месте
}
// вывод преобразованного вектора
cout << "Преобразованный вектор: \n";
for(int i = 0; i < v.size(); i++) // Вывод вектора
cout << v[i] << " ";
cout<<endl;
system("pause");
return 0;
}
```

Когда лучше использовать массивы, а когда векторы?

Используйте `std::array` или необработанные массивы для небольшого статического числа элементов.

Если у вас много элементов (более, скажем, 100 Кб), вы загружаете стек. В этом случае или если количество элементов может быть известно только во время выполнения, используйте `std::vector`.

4. Примеры решения задач

Лабораторная работ 4

Общее задание

Для всех задач сформировать одномерный массив N случайных чисел из интервала $(-10; 10)$. Предусмотреть автоматический и ручной способ создания элементов массива. Выполнить операции над массивом по варианту. Вывести на экран в строчку исходный массив и результат с соответствующими заголовками.

Вариант 16

1. Проверить, верно ли, что в одномерном массиве все его элементы меньше 7?
2. В одномерном массиве, состоящем из n вещественных элементов, вычислить:
 - количество отрицательных элементов массива;
 - сумму модулей элементов массива, расположенных после минимального по модулю элемента.
3. Заменить все отрицательные элементы массива их квадратами и упорядочить элементы массива по возрастанию.
4. Отрицательные элементы в массиве возвести в квадрат и полученный массив отсортировать по возрастанию методом вставки.

Лабораторная работа 4.1

```
#include <cstdlib>
#include <iostream>
#include <clocale>
#include <cmath>
#include <ctime>
```

```
using namespace std;
```

```
int main(int argc, char *argv[])
```

```
{
    setlocale(LC_ALL, "Russian");
    int size=0;
    cout << "Введите число элементов массива n=";
    cin >> size;
```

```
    srand (time(NULL)); // инициализация,
// привязка генератора случайных чисел
// к счетчику секунд от 1.01.1970
```

```
    int sposob = 0;
```

```
    cout << "Выберите ручной - 0 или
автоматический (любой другой символ)
способ создания элементов массива "
<< endl;
```

```
    cin >> sposob;
```

```
    float arr[size]{0};
```

```
    int i=0;
```

Лабораторная работа 4.1

```
#include <cstdlib>
#include <iostream>
#include <clocale>
#include <cmath>
#include <ctime>
using namespace std;
int main(int argc, char *argv[])
{
    setlocale(LC_ALL, "Russian");
    const int size=5;
```

```
srand (time(NULL)); // инициализация,
// привязка генератора случайных чисел
// к счетчику секунд от 1.01.1970
```

```
int sposob = 0;
```

```
cout << "Выберите ручной - 0 или
автоматический (любой другой символ)
способ создания элементов массива "
<< endl;
```

```
cin >> sposob;
```

```
float arr[size]{0};
```

```
int i=0;
```

Лабораторная работа 4.1

```
#include <cstdlib>
#include <iostream>
#include <locale>
#include <cmath>
#include <ctime>
using namespace std;

int main(int argc, char *argv[])
{
    setlocale(LC_ALL, "Russian");

    int *arr = new int[size];

    srand (time(NULL)); // инициализация, привязка
                        //генератора случайных чисел к счетчику секунд
                        //от 1.01.1970

    int sposob = 0;

    cout << "Выберите ручной - 0 или
автоматический (любой другой символ) способ
создания элементов массива " << endl;

    cin >> sposob;

    float arr[size]{0};

    int i=0;

    delete [] arr; // очистка памяти
    return 0;
}
```

Окончание нашей программы

Лабораторная работа 4.1

```
if (sposob == 0)
{
    cout << "Введите "<< size << " элемента/ов
    массива от -10 до 10 "<< endl;
    for (int i = 0; i < size; i++) {
        cin >> arr[i] ;
    }
}
else {
    cout << "Автоматический ввод" << endl;
    for (int i = 0; i < size; i++)
    {
        arr[i] = (rand() % 21) - 10;
        //min+rand()%(max-min+1)
    }
}
```

```
cout << "Вывод результата: " << endl;

for (int i = 0; i < size; i++)
{
    cout << arr[i] << " ";
}
}
```

Лабораторная работа 4.1

```
//Проверим, все ли элементы массива  
//меньше 7
```

```
bool z=true;
```

```
for (int i = 0; i < size; i++)
```

```
{
```

```
if (arr[i]>=7)
```

```
{
```

```
z = false;
```

```
}}
```

```
if (z) // z==true
```

```
{
```

```
cout << endl << "Все элементы массива  
меньше 7 " << endl;
```

```
}
```

```
else
```

```
{
```

```
cout << endl << "Не все элементы  
массива меньше 7 " << endl;
```

```
}
```

```
return 0;
```

```
}
```

Лабораторная работа 4.1

 Консоль отладки Microsoft Visual Studio

Введите число элементов массива $n=5$

Выберите ручной - 0 или автоматический (любой другой символ) способ создания элементов массива
0


Введите 5 элемента/ов массива от -10 до 10

1 2 3 4 5

Вывод результата:

1 2 3 4 5

Все элементы массива меньше 7

 Консоль отладки Microsoft Visual Studio

Введите число элементов массива $n=5$

Выберите ручной - 0 или автоматический (любой другой символ) способ создания элементов массива
0

Введите 5 элемента/ов массива от -10 до 10


8 8 9 9 10

Вывод результата:

8 8 9 9 10

Не все элементы массива меньше 7

Лабораторная работа 4.1

 Консоль отладки Microsoft Visual Studio

Введите число элементов массива $n=5$

Выберите ручной - 0 или автоматический (любой другой символ) способ создания элементов массива
3

Автоматический ввод

Вывод результата:

-7 7 5 8 9

Не все элементы массива меньше 7

Лабораторная работа 4.2

//Подсчитаем количество отрицательных
//элементов массива

```
int z=0;
```

```
for (int i = 0; i < size; i++)
```

```
{
```

```
if (arr[i]<0)
```

```
{
```

```
z=z+1;
```

```
}
```

```
}
```

```
cout << endl << "Количество
```

```
отрицательных элементов массива равно =
```

```
" << z << endl;
```

//Подсчитаем сумму модулей элементов
//массива, расположенных после
//минимального по модулю элемента

// Поиск минимального элемента

```
int m=0;
```

```
int imin = 0;
```

```
for (int i = 0; i < size; i++) {
```

```
if(abs(arr[i]) < abs(arr[imin]))
```

```
{
```

```
imin = i;
```

```
} }
```

```
cout << endl << "Минимальный элемент  
массива равен = " << arr[imin]; // вывод  
минимального значения
```

Лабораторная работа 4.2

```
// Сумма элементов после минимального
```

```
float sum = 0;
```

```
// if (imin+1 < size) {
```

```
    for (int i = imin + 1 ; i < size; i++)
```

```
    {
```

```
        sum += abs(arr[i]);
```

```
    }
```

```
}
```

```
cout << endl << "Сумма элементов после минимального равна = " << sum;
```

Лабораторная работа 4.3

```
//Заменить отрицательные элементы  
//массива их квадратами
```

```
for (int i = 0; i < size; i++)  
{  
    if (arr[i]<0)  
    {  
        arr[i] *= arr[i] ; // arr[i] = arr[i] * arr[i]  
    }  
}
```

```
// Сортировка массива пузырьком
```

```
int temp; // временная переменная для  
//обмена элементов местами
```

```
for (int i = 0; i < size - 1; i++) {  
    for (int j = 0; j < size - i - 1; j++) {  
        if (arr[j] > arr[j + 1]) {  
            // меняем элементы местами  
            temp = arr[j];  
            arr[j] = arr[j + 1];  
            arr[j + 1] = temp;  
        }  
    }  
}
```

Лабораторная работа 4.4

// Заменить отрицательные элементы
массива их квадратами

```
for (int i = 0; i < size; i++)  
{  
    if (arr[i]<0)  
    {  
        arr[i] = arr[i] * arr[i];  
    }  
}
```

// Сортировка массива методом вставки

```
for(int i=1;i<size;i++)  
for(int j=i;j>0 && arr[j-1]>arr[j];j--) // пока  
j>0 и элемент j-1 > j, arr-массив int  
    swap(arr[j-1],arr[j]); // меняем местами  
//элементы j и j-1  
  
cout << "Вывод результата: " << endl;  
for (int i = 0; i < size; i++)  
{  
    cout << arr[i] << " ";  
}
```

Лабораторная работа 5

Составить программу решения задачи. Все числовые значения должны вводиться в процессе исполнения программы. Размерность (единицы измерения) исходных значений должна быть сохранена.

Задать двумерный массив размерности $N \times M$ натуральных случайных чисел из интервала $(-10; 10)$. Предусмотреть автоматический и ручной способ ввода элементов массива. Вывести на экран в виде таблицы исходный массив и преобразованный с соответствующими заголовками.

Вариант №16

1. Дана матрица размера $N \times M$. Вывести номер ее последнего столбца, содержащего только положительные элементы. Если таких столбцов нет, то вывести 0.
2. Дана квадратная матрица порядка M . Зеркально отразить ее элементы относительно побочной диагонали матрицы
3. Дана матрица размера $M \times N$. Поменять местами строку с номером x и первую из строк, содержащих только отрицательные элементы.
4. Найти в матрице A значение максимального элемента и определить его местоположение среди элементов подматрицы $A(i, j)$, расположенной в левом верхнем углу матрицы A . Если таких элементов несколько, то определить их количество.

Лабораторная работа 5.1

```
#include <cstdlib>
#include <iostream>
#include <locale>
#include <cmath>
#include <ctime>

using namespace std;

int main(int argc, char *argv[]) {
    setlocale(LC_ALL, "Russian");

    int rows = 0;    int cols = 0;

    cout << "Введите число строк массива: ";
    cin >> rows;

    cout << "Введите число столбцов массива: ";
    cin >> cols;

    srand (time(NULL));

    int sposob = 0;

    cout << "Выберите ручной - 0 или
автоматический (любой другой символ)
способ создания элементов массива " <<
endl;

    cin >> sposob;

    float arr[rows][cols]{0};

    if (sposob == 0) {

        cout << "Введите "<< rows*cols << "
элемента/ов массива от -10 до 10 "<< endl;

        for (int i = 0; i < rows; i++) {
            for (int j=0; j < cols; j++) {
                cin >> arr[i] [j]; } } }
```

Лабораторная работа 5.1

```
#include <cstdlib>

#include <iostream>

#include <clocale>

#include <cmath>

#include <ctime>

using namespace std;

int main(int argc, char *argv[]) {

    setlocale(LC_ALL, "Russian");

    const int rows = 3;

    const int cols = 3;
```

```
srand (time(NULL));

int sposob = 0;

cout << "Выберите ручной - 0 или
автоматический (любой другой символ) способ
создания элементов массива " << endl;

cin >> sposob;

float arr[rows][cols]{0};

if (sposob ==0) {

    cout << "Введите "<< rows*cols << "
элемента/ов массива от -10 до 10 "<< endl;

    for (int i = 0; i < rows; i++) {

        for (int j=0; j < cols; j++) {

            cin >> arr[i] [j]; } } }
```

Лабораторная работа 5.1

```
#include <cstdlib>
#include <iostream>
#include <locale>
#include <cmath>
#include <ctime>
using namespace std;
int main(int argc, char *argv[]) {
    setlocale(LC_ALL, "Russian");
    int rows = 0;
    int cols = 0;
    cout << "Введите число строк массива: ";
    cin >> rows;
    cout << "Введите число столбцов массива: ";
    cin >> cols;
    int** arr = new int* [rows];
    for (int i = 0; i < rows; ++i)
        arr[i] = new int[cols];
```

```
srand (time(NULL));

int sposob = 0;

cout << "Выберите ручной - 0 или
автоматический (любой другой символ)
способ создания элементов массива " <<
endl;

cin >> sposob;

if (sposob == 0) {

    cout << "Введите " << rows*cols << "
элемента/ов массива от -10 до 10 " << endl;

    for (int i = 0; i < rows; i++) {

        for (int j=0; j < cols; j++) {

            cin >> arr[i] [j]; } } }
```


Лабораторная работа 5.1

```
else
{
cout << "Автоматический ввод" << endl;
for (int i = 0; i < rows; i++)
{
for (int j=0; j < cols; j++)
{
cout << arr[i] [j] << "\t" ; // добавляем символ
табуляции
}
cout << endl; // формируем привычную форму
//массива "строки-столбцы"
}
}
```

Лабораторная работа 5.1

```
//Вывод последнего столбца
//массива, содержащего только
//положительные элементы
cout << "Вывод результата" << endl;
int k; // массив содержит
положительные элементы, если "1",
целесообразно выбирать
логический тип данных
for (int j = cols-1; j >= 0; j--) {
    k = 1;
    for (int i = 0; i < rows; i++) {
        if (arr[i] [j]<=0) {
            k = 0;
        }
    }
}
```

```
if (k==1) {
    for (int i = 0; i < rows; i++) {
        cout << endl << arr[i] [j] << " ";
        cout << endl ;
    }
    cout << "Номер столбца: " << j + 1 << endl;
    break;
}
}
if (k==0)
    cout << "0";
```

Лабораторная работа 5.1

Консоль отладки Microsoft Visual Studio

Выберите ручной - 0 или автоматический (любой другой символ) способ создания элементов массива

2

Автоматический ввод

Вывод исходного массива:

10 -10 -7

-10 -4 10

-5 -8 -9

Вывод результата

0

Консоль отладки Microsoft Visual Studio

Введите число строк массива: 3

Введите число столбцов массива: 3

Выберите ручной - 0 или автоматический (любой другой символ) способ создания элементов массива

0

Введите 9 элемента/ов массива от -10 до 10

1 1 1 1 1 1 1 1 1

Вывод исходного массива:

1 1 1

1 1 1

1 1 1

Вывод результата

1

1

1

Номер столбца: 3

Лабораторная работа 5.2

```
// Вывод обновленной матрицы
//(зеркально отраженные элементы)
int buff = 0;
for (int i=0; i<n-1; i++) {
    for (int j=0; j<n-1-i; j++) {
        buff = arr[i] [j];
        arr[i][j]=arr[n-1-i][n-1-j];
        arr[n-1-i][n-1-j] = buff;
    }
}
```

```
cout << endl;
cout << "Вывод преобразованной матрицы: " << endl;
for (int i = 0; i < n; i++) {
    for (int j=0; j < n; j++) {
        cout << arr[i] [j] << "\t" ; // добавляем символ
        //табуляции
    }
    cout << endl; // формируем привычную форму
    //массива "строки-столбцы"
}
```

Лабораторная работа 5.3

//Ищем первую строку, содержащую только

//отрицательные элементы

int k = 0; // строка содержит только отрицательные
элементы, если "1" (целесообразно выбирать
логический тип данных)

int y = 0; // номер первой строки, которая содержит
только отрицательные элементы

int i = 0;

int j = 0;

for (i = 0; i < rows; i++) {

k = 1;

for (j = 0; j < cols; j++) {

if (arr[i] [j]>=0)

{

k = 0;

}

}

if (k==1)

{

y = i;

break;

}

}

Лабораторная работа 5.3

```
if (k==0){  
  
    cout << "Строки с заданными условиями  
отсутствуют " << endl; // можно предусмотреть  
//дополнительно условия для продолжения или  
//выхода  
}  
  
else {  
  
    cout << endl << "Номер первой строки с  
отрицательными значениями: " << y + 1 << endl;}  
  
    cout << endl;  
  
    cout << "Введите номер строки X " << endl;  
  
    int x = 0;  
  
    cin >> x ;  
  
    float buff = 0; // для хранения промежуточных  
//результатов
```

```
    cout << "Вывод результата" << endl;  
  
    x=x-1;  
  
    for (int j = 0; j < cols; j++) {  
  
        buff = arr[y][j] ;  
  
        arr[y][j] = arr[x][j] ;  
  
        arr[x][j] = buff; }  
  
    for (int i = 0; i < rows; i++) {  
  
        for (int j=0; j < cols; j++) {  
  
            cout << arr[i] [j] << "\t" ;  
  
        }  
  
        cout << endl;  
  
    }
```

Лабораторная работа 5.4

```
//Вывод последнего столбца массива,  
//содержащего только положительные  
//элементы
```

```
cout << "Вывод результата" << endl;
```

```
//нахождение максимального элемента
```

```
int mi = 0, mj = 0; // индекс i и j
```

```
//максимального элемента
```

```
int n = 0, m = 0;
```

```
//меняем размеры подматриц
```

```
for(int n = rows; n > 1; n--) // n>1, т.к.
```

```
//минимальный размер матрицы 2 {
```

```
    for(int m = cols; m > 1; m--) {
```

```
        mi = 0;
```

```
        mj = 0;
```

```
//пробегаем по всем элементам
```

```
//подматрицы в поисках максимального
```

```
//элемента
```

```
    for(int i = 0; i < n; i++) {
```

```
        for(int j = 0; j < m; j++) {
```

```
            if(arr[i][j]>arr[mi][mj]) {
```

```
                mi = i;
```

```
                mj = j;
```

```
            }}}
```

```
cout << "Положение максимального  
элемента в подматрице arr(" << n << ", "  
<< m << ") : " << "arr(" <<mi << ", " <<  
mj << ") = " << arr[mi][mj] << endl;
```

```
}}
```

СПАСИБО ЗА ВНИМАНИЕ!