# Лабораторная работа № 2

## Одномерные массивы (0,5 баллов за задание)

В заданиях можно использовать статические и динамические массивы, нельзя использовать классы С++ (например, vector и т.д.).

Вводится длина одномерного массива (вектора) и значения его элементов.

1. Поменять местами максимальный и минимальный элементы массива. Вывести измененный массив на экран.
2. Все элементы массива, меньшие заданного значения, и их номера записать в два новых массива (в один значения в другой номера). Вывести новые массивы на экран. Заданное значение вводится с клавиатуры.
3. Вводится дополнительный массив разрешенных значений. Определить и вывести на экран, сколько элементов исходного массива имеют разрешенные значения.
4. Составить и вывести на экран массив из различных (по значению) элементов исходного массива.
5. Составить и вывести на экран два массива: массив повторяющихся элементов исходного массива и массив их частот.

## Двумерные массивы (0,5 баллов за задание)

Ввести размер прямоугольной матрицы и значения ее элементов.

Заполнить и вывести одномерный массив В, где В[i] –

1. число неотрицательных элементов в i-ой строке.
2. среднее арифметическое положительных элементов в i-ом столбце.
3. минимальное значение в i-ой строке.
4. номер максимального значения в i-ой строке.
5. количество простых чисел в i-ой строке.

i – это переменная цикла, которая изменяется от 0 до N-1 (количества строк)

см. следующие страницы!

**Примеры**

**Пример работы с одномерным динамическим массивом**

**Функция для ввода элементов массива. Функции передается адрес массива и количество элементов**

void input(int \*a, int n)

{

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

{

printf("Введите элемент массива A[%d]=", i);

scanf\_s("%d", a + i);

}

}

**Функция вывода элементов массива. Функции передается адрес массива и количество элементов**

void output(int \*a, int n)

{

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

printf("A[%d]=%d\n", i, \*(a+i));

}

**Главная функция программы.**

*Создается указатель для массива (int \*A).*

*Выделяется память для N элементов массива (A=new int[N]).*

*Вызываются функции input и output для ввода и вывода массива*

*Память освобождается (delete [] A)*

int main(int argc, char\*\* argv)

{

int \*A, N;

setlocale(LC\_ALL,"russian");

printf\_s("Введите количество элементов массива: ");

scanf\_s("%d",&N);

A = new int[N]; // выделение памяти под N элементов типа int

input(A,N);

// ... добавьте сюда код для работы с массивом

output(A,N);

delete[] A; // освобождение памяти

return 0;

}

**Пример работы со статическим двумерным массивом**

**Определение массива**

const unsigned int DIM1 = 3;

const unsigned int DIM2 = 5;

int arr[DIM1][DIM2];

**Заполнение массива нулевыми значениями**

for (int i = 0; i < DIM1; i++) {

for (int j = 0; j < DIM2; j++) {

arr[i][j] = 0;

}

}

**Пример работы с динамическим двумерным массивом**

**Определение массива N\*N**

int n;

cin >> n;

// выделение памяти под указатели на строки массива

int \*\*arr = new int \*[n];

**Выделение памяти под строки**

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

    arr[i] = new int [n];

**Заполнение массива нулевыми значениями**

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

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

arr[i][j] = 0;

}

}

**Освобождение памяти, выделенной под массив**

// освобождение памяти, выделенной для строк

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

delete [] arr[i];

//освобождение памяти, выделенной под указатели на строки

delete [] arr;

**Пример работы с двумерным динамическим массивом заданным в другом виде**

На самом деле это одномерный массив, но с ним можно работать как с двумерным (менее эффективно используется фрагментация памяти)

double\* matrix = new double[100, 100];

**Заполнение массива нулевыми значениями**

for (int i = 0; i < 100; i++)

for (int j = 0; j < 100; j++) {

matrix[i,j] = 0;

}

**Освобождение памяти**

delete [] matrix;