Отчет

по лабораторной работе № 1

«Разработка, отладка, и тестирование программ,

выполняемых под управлением ОС Windows»

по дисциплине

ОПЕРАЦИОННЫЕ СИСТЕМЫ

Выполнил студент гр. ИС/б-22о

Горбенко К.Н.

Проверила:

Шалимова Е.М.

**Цель работы**: изучение возможностей, предоставляемых ОС Windows для разработки, отладки и тестирования программ.

* 1. Постановка задачи

### Задание 1

Разработать алгоритм и программу на языке Pascal/C, в которой ввести одномерный массив x, содержащий n элементов заданного типа, выполнить обработку в соответствии с вариантом. Исходный массив и результаты вывести на экран. Индивидуальное задание для **варианта № 4:** упорядочить элементы массива по убыванию.

### Задание 2

Разработать алгоритм и программу на языке Pascal/C, в которой ввести двумерный массив X, содержащий N строк и M столбцов элементов заданного типа; выполнить обработку по варианту. Исходный массив и результаты вывести на экран, при этом двумерный массив выводить в виде матрицы. Индивидуальное задание для **варианта № 4:** сформировать одномерный массив, состоящий из максимальных элементов каждого столбца.

* 1. Словесное описание алгоритма

### Алгоритм сортировки пузырьком для задания № 1

Для упорядочивания элементов массива воспользуемся методом сортировки пузырьком. Алгоритм состоит из N-1 этапа (N – количество элементов массива), на каждом этапе на первую позицию неотсортированной части массива устанавливается наибольший элемент неотсортированной части массива. Каждый этап состоит из N-K (K – номер текущего этапа) последовательных сравнений соседних элементов.

### Алгоритм для задания № 2

Для получения наибольшего элемента каждой строки исходная матрица проходится в двух циклах: внутри первого цикла фиксируется номер столбца и инициализируется минимальным значением целого типа переменная, используемая для сравнения элементов, пока второй цикл изменяет номер строки. Внутри второго цикла каждый элемент столбца сравнивается со значением переменной, содержащей текущий максимальный элемент. В конце каждого этапа максимальный элемент столбца помещается в вектор.

* 1. Описание используемых структур данных

Для представления векторов и матриц используются стандартные динамические массивы целого типа языка C++. Для получения минимального значения типа int используется константа INT\_MIN библиотеки <limits.h>.

* 1. Спецификации подпрограмм

В программе используются разработанные подпрограммы:

void viewmatrix(int \*\*array, int size0, int size1)

{

cout << "Matrix" << endl;

for (int i = 0; i < size0; i++) {

for (int j = 0; j < size1; j++)

cout << array[i][j] << " ";

cout << endl;

}

}

viewmatrix принимает двумерный массив типа int (указатель) и его размерности и выводит его на экран.

void viewarray (int \* array, int size0)

{

cout << "Array" << endl;

for (int i = 0; i < size0; i++) {

cout << array[i] << "\t";

}

}

viewarray принимает массив типа int (указатель) и его размер и выводит его на экран.

void sortarray(int \*array, int size)

{

    for (int i = 0; i < size - 1; i++)

    {

        for (int j = size - 1; j > 0; j--)

        {

            if (array[j] > array[j - 1])

            {

                int temp = array[j];

                array[j] = array[j - 1];

                array[j - 1] = temp;

            }

        }

    }

}

sortarray принимает массив типа int (указатель на массив) и его размер и сортирует массив пузырьком.

int \* get\_column\_max\_elements(int \*\*array, int size0, int size1)

{

int \*columnsMaxElements = new int[size0];

for (int j = 0; j < size1; j++)

{

int columnMax = INT\_MIN;

for (int i = 0; i < size0; i++)

{

if (array[i][j] > columnMax)

columnMax = array[i][j];

}

columnsMaxElements[j] = columnMax;

}

return columnsMaxElements;

}

get\_column\_max\_elements принимает двумерный массив (указатель) и его размерности и возвращает массив с максимальными элементами каждого столбца матрицы.

* 1. Тестовые примеры

Введем массив из 10 случайных целых чисел в случайном порядке. Он будет отсортирован по убыванию.

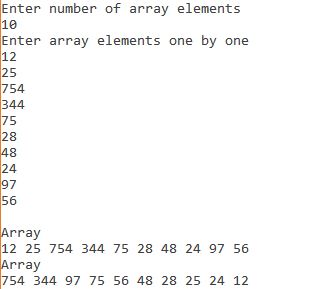


Рис. 1 – Результат работы первой программы

Введем матрицу размера 3x6 и заполним ее случайными числами. Программа выводит максимальные элементы каждого столбца.

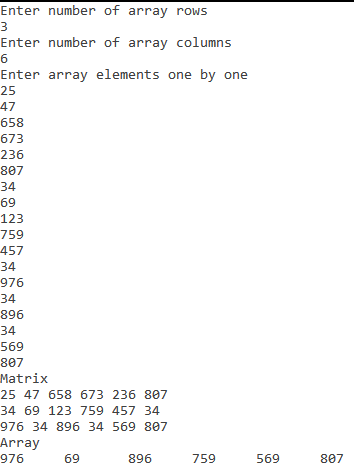


Рис. 2 – Результат работы второй программы

* 1. Тексты программ
     1. Текст программы к заданию № 1

#include <pch.h>

#include <iostream>

using namespace std;

void viewarray(int \*array, int size)

{

cout << endl << "Array" << endl;

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

cout << array[i] << " ";

}

}

void sortarray(int \*array, int size)

{

for (int i = 0; i < size - 1; i++)

{

for (int j = size - 1; j > 0; j--)

{

if (array[j] > array[j - 1])

{

int temp = array[j];

array[j] = array[j - 1];

array[j - 1] = temp;

}

}

}

}

int main()

{

int N;

cout << "Enter number of array elements" << endl;

cin >> N;

int \*arr = new int[N];

cout << "Enter array elements one by one" << endl;

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

cin >> arr[i];

viewarray(arr, N);

sortarray(arr, N);

viewarray(arr, N);

delete[] arr;

}

* + 1. Текст программы к заданию № 2

#include <pch.h>

#include <iostream>

#include <limits.h>

using namespace std;

void viewmatrix(int \*\*array, int size0, int size1)

{

cout << "Matrix" << endl;

for (int i = 0; i < size0; i++) {

for (int j = 0; j < size1; j++)

cout << array[i][j] << " ";

cout << endl;

}

}

int \* get\_column\_max\_elements(int \*\*array, int size0, int size1)

{

int \*columnsMaxElements = new int[size0];

for (int j = 0; j < size1; j++)

{

int columnMax = INT\_MIN;

for (int i = 0; i < size0; i++)

{

if (array[i][j] > columnMax)

columnMax = array[i][j];

}

columnsMaxElements[j] = columnMax;

}

return columnsMaxElements;

}

void viewarray(int \* array, int size0)

{

cout << "Array" << endl;

for (int i = 0; i < size0; i++) {

cout << array[i] << " ";

}

}

int main()

{

int M, N;

cout << "Enter number of array rows" << endl;

cin >> M;

cout << "Enter number of array columns" << endl;

cin >> N;

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

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

arr[i] = new int[N];

cout << "Enter array elements one by one" << endl;

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

for (int j = 0; j < N; j++)

cin >> arr[i][j];

viewmatrix(arr, M, N);

int \* columnsMaxElements = get\_column\_max\_elements(arr, M, N);

viewarray(columnsMaxElements, N);

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

delete[] arr[i];

delete[] arr;

}

* 1. Вывод

В ходе лабораторной работы были изучены средства для разработки, отладки и тестирования программ под управлением ОС Windows. В качестве среды разработки использовалась среда MS Visual Studio, имеющая встроенные средства для компиляции и отладки программ. Отладчик операционной системы позволяет останавливать выполнение программы на любом этапе и просматривать значения переменных, используемых в ней. Эта возможность ускоряет процесс выявления ошибок в программном коде.