Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«КУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Кафедра программного обеспечения и администрирования информационных систем

Направление подготовки математическое обеспечение и администрирование информационных систем

Форма обучения очная

**Отчет**

**по лабораторной работе №1**

«Программирование на языке C++ с использованием рекурсивных функций и

функций со сложными параметрами»

дисциплина «Основы алгоритмизации и программирования»

Выполнил:

студент группы 113 Гришанов Е.А.

Проверил:

старший преподаватель кафедры ПОиАИС Ураева Е.Е.

Курск, 2024

***Цель работы:*** Изучить особенности написания программ на языке С++ с использованием рекурсивных функций и функций со сложными параметрами.

***Задание***

*Задача 1.* Описать бестиповую функцию Time24ToTime12 (H, M, S, am), которая преобразует время, заданное в часах H, минутах M и секундах S (H, M и S — целые положительные) из 24-часового формата в 12-часовой, при этом для времени в 12-часовом формате устанавливается признак принадлежности момента времени первой половине суток am (am - логическое). Используя эту функцию, привести к новому формату два заданных временных отрезка: (T1, M1, S1) и (T2, M2, S2).

*Задача 2.* Даны два массива целых чисел размеров n и m и целые числа k и t. Если в первом массиве все элементы не превосходят k, а во втором массиве все элементы не превосходят t, то в первом массиве заменить все элементы равные k значением t, в противном случае во втором массиве заменить все элементы, равные t значением k.

*Задача 3.* Используя рекурсивную функцию, для заданного числа n определить значение выраженияx – действительное число, i = 0, 1, … n, j = 1, 2, …n – 1.

***Разработка алгоритма***

*Задача 1*

Входные данные: *T1, T2, M1, M2, S1, S2 –* целые числа.

Выходные данные: *T1, T2, M1, M2, S1, S2 –* целые числа.

Алгоритм решения задачи представлен на рисунке 1.

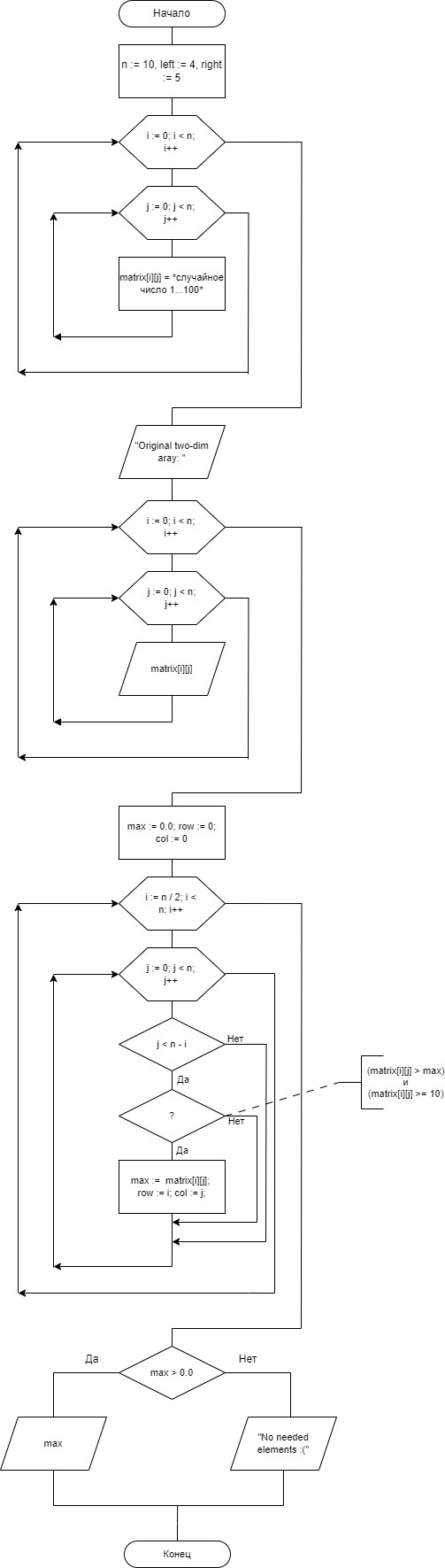


Рисунок 1 – Алгоритм решения задачи 1

*Задача 2*

Входные данные: *col* – целое число.

Выходные данные: *matrix* – двумерный массив.

Алгоритм решения задачи представлен на рисунке 2.

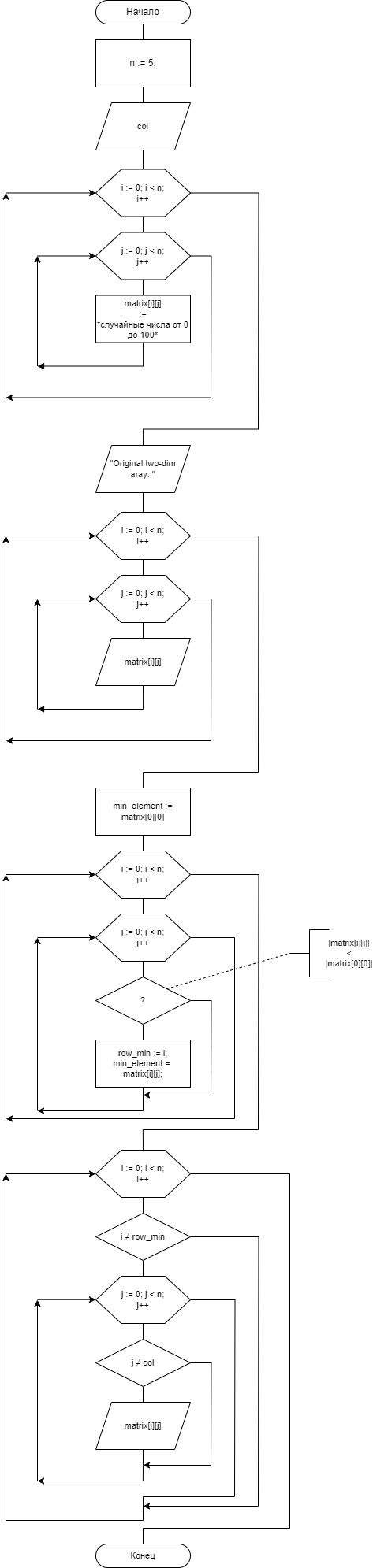


Рисунок 2 – Алгоритм решения задачи 2

***Текст программы***

*Текст программы для решения задачи 1*

#include <iostream>

#include <iomanip>

using namespace std;

int main(){

const int n = 10, left = 4, right = 5;

double matrix[n][n];

// Ввод массива с помощью "рандомных" чисел

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

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

matrix[i][j] = rand() % 100;

}

}

// Вывод заполенного массива

cout << "Original two-dim aray: " << endl;

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

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

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

}

cout << endl;

}

// Поиск наибольшего двузначного значения в заданной области

cout << "\n";

double max = 0.0;

int row, col = 0;

for(int i = n / 2; i < n; i++){

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

if (j < n - i){

if((matrix[i][j] > max) && (matrix[i][j] >= 10))

{

max = matrix[i][j];

row = i;

col = j;

}

}

}

}

// Проверка на нахождение максимального знчения и его индексов

if (max > 0.0){

cout << "Max element in scope: " << max << endl;

cout << row << " " << col;

}

else cout << "No needed elements :(";

}

*Текст программы для решения задачи 2*

// TODO: Дана матрица действительных чисел размера n×n. Вывести минор матрицы на пересечении i-й строки и j-го столбца, где i-номер строки, содержащий минимальный по модулю элемент матрицы, а значение j вводится пользователем.

#include <iostream>

#include <cmath>

using namespace std;

int main(){

int n = 0;

cin >> n;

int col;

cout << "Enter j: " << endl;

cin >> col;

// Объявление и выделение памяти для двумерного динамического массива

double \*\*matrix = new double\* [n];

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

matrix[i] = new double [n];

}

// Заполнение массива с помощью "рандомных" чисел

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

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

matrix[i][j] = (double)(rand() % 100) + 1;

}

}

matrix[0][0] = 0;

// Вывод заполенного массива

cout << "Original two-dim aray: " << endl;

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

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

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

}

cout << endl;

}

// Определение строки содержащей минимальный элемент матрицы

double min\_element = abs(matrix[0][0]);

int row\_min = 0;

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

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

if (abs(matrix[i][j]) < abs(matrix[0][0])){

row\_min = i;

min\_element = matrix[i][j];

}

}

}

// Вывод минора

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

if (i != row\_min)

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

if (j != col)

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

}

cout << endl;

}

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

delete[] matrix[i];

}

delete[] matrix;

}

***Тестирование программы***

Тестирование задачи 1 представлено на рисунках 3, 4.

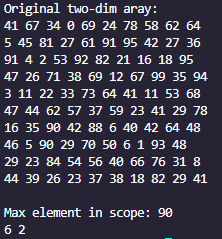


Рисунок 3 –Тест 1 задачи 1

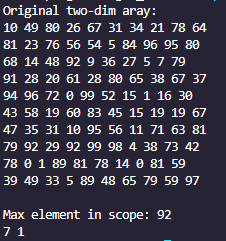


Рисунок 4 – Тест 2 задачи 1

Тестирование задачи 2 представлено на рисунках 5, 6.

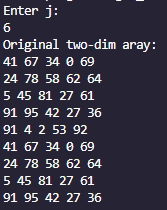


Рисунок 5 –Тест 1 задачи 2

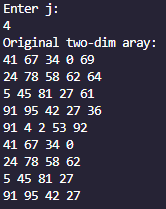


Рисунок 6 –Тест 2 задачи 2