Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

Пермский национальный исследовательский политехнический университет

Кафедра «Информационные технологии и автоматизированные системы»

Лабораторная работа №5

«Функции и массивы»

Вариант 18

Выполнил студент группы ИВТ-20-1б

Матяж Владимир Олегович

Проверил старший преподаватель

Ярулин Денис Владимирович

Пермь 2020

**Цель работы:**

Получение практических навыков при работе с массивами.

**Постановка задачи:**

Используя функции решить указанную в варианте задачу. Массив должен передаваться в функцию.

**Вариант:**

Задан двумерный массив N x N. Последовательно рассматриваются квадратные подмассивы, правый верхний элемент которых лежит на побочной диагонали. В каждом таком подмассиве находится максимальный элемент. Путем перестановок строк и столбцов (целиком) элемент надо переместить в правый верхний угол подмассива. Проверить получилась ли на побочной диагонали убывающая последовательность элементов.

**Анализ задачи:**

Запрашиваем от пользователя размер массива и сохраняем ее значение в целочисленную переменную, обозначим ее n.

Создадим динамический массив a. Создадим динамический массив b.

Заполним массив a с помощью генератора случайных чисел.

Передадим указатели на массивы a и b, их размер в функцию m.

Запускаем цикл for, который будет уменьшать количество строк и столбцов в подмассивах. Делаем левый верхний элемент подмассива максимальным, сохраняем номер столбца и строки в две целочисленные переменные, обозначим их jmax и imax. С помощью двух вложенных циклов for осуществляем поиск максимального элемента подмассива, сравнивая значения с a[imax][jmax], если они оказываются больше, присваиваем jmax и imax номера их столбца и сроки. Делаем проверку, если максимальный элемент подмассива стоит в строке, находящейся, ниже крайней верхней строки, т.е. номер строки, где находится максимальный элемент, больше номера верхней строки, то сохраняем строку в массив b, строке с максимальным присваиваем значения верхней строки, значения из массива b, присваиваем верхней строке подмассива. Делаем вторую проверку, если максимальный элемент подмассива стоит в столбце, находящимся, левее крайнего правого столбца, т.е. номер крайнего правого столбца подмассива больше номера столбца, в котором находится максимальный элемент, то сохраняем столбец с максимальным элементом в массив b, присваиваем ему значения правого столбца, правому столбцу присваиваем значение из массива b. С каждым разом подмассив становится меньше на 1 строку и 1 столбец. После окончания основного цикла делаем проверку побочной диагонали, получилась ли она убывающей. Реализуем проверку с помощью цикла while, для этого понадобится целочисленная переменная, для перехода по строкам, обозначим ее m, и переменная типа bool, обозначим ее flag. Цикл будет работать, пока m меньше размера массива и flag не поменяет свое значение. В цикле проверяем значения побочной диагонали, начиная с верхнего правого элемента отсортированного массива, вниз по диагонали. Если значение последующего элемента больше предыдущего или равно ему, flag меняет свое значение, иначе продолжаем дальше проверять, m прибавляем единицу. После окончания цикла, проверяем с помощью if, если flag поменяла свое значение, то выводим «Диагональ не упорядочена», иначе «Диагональ упорядочена».

**Код программы:**

#include <iostream>

using namespace std;

int n;

void m(int\*\* g, int\* b, int u)

{

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

{

int imax = i;

int jmax = 0;

for (int k = i; k < u; k++)

{

for (int j = 0; j < u - i; j++)

{

if (g[k][j] > g[imax][jmax])

{

imax = k;

jmax = j;

}

}

}

if (imax > i)

{

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

{

b[j] = g[imax][j];

}

for (int k = imax; k > i; k--)

{

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

{

g[k][j] = g[k - 1][j];

}

}

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

{

g[i][j] = b[j];

}

}

if (jmax < u - i - 1)

{

for (int k = 0; k < u; k++)

{

b[k] = g[k][jmax];

}

for (int j = jmax; j < u - i - 1; j++)

{

for (int k = 0; k < u; k++)

{

g[k][j] = g[k][j + 1];

}

}

for (int k = 0; k < u; k++)

{

g[k][u - i - 1] = b[k];

}

}

}

cout << "Массив после пересьановки" << endl;

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

{

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

{

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

}

cout << endl;

}

bool flag = false;

int m = 1;

while (m < u && flag == false)

{

if (g[m][u - m - 1] >= g[m - 1][u - m])

{

flag = true;

}

else

{

m++;

}

}

if (flag == true)

{

cout << " На побочной диагонали не убывающая последовательность " << endl;

}

else

{

cout << "На побочной диагонали убывающая последовательность " << endl;

}

}

int main()

{

setlocale(LC\_ALL, "Rus");

cout << "Введиите размер массива " << endl;

cin >> n;

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

int\* b = new int [n];

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

{

a[i] = new int[n];

}

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

{

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

{

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

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

}

cout << endl;

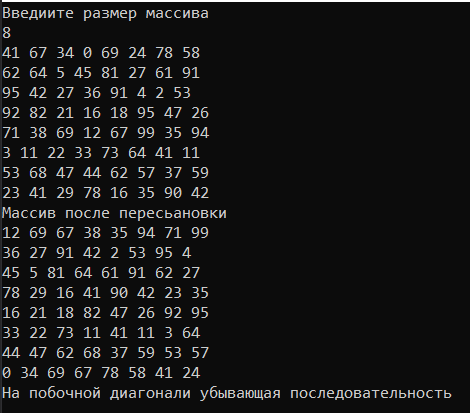
}

m(a, b, n);

return 0;

}

**Результат работы программы:**

****