МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

**ИНТЕЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

ОТЧЁТ

По лабораторной работе № \_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил:

Студент группы ИИ-22

Копанчук Евгений Романович

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Брест – 2023

Ход работы

TASK

|  |
| --- |
| 5. Задача о магических квадратах. Написать программу, размещающую числа 1, 2, 3, ... ..., n^2 в квадратной таблице n х n (п < … ) так, чтобы суммы по всем столбцам, строкам и главным диагоналям были одинаковы. |

CODE

|  |
| --- |
| #include <iostream>  #include <vector>  #include <cmath>  #include <Windows.h>  using namespace std;  void generatePermutations(vector<int>& nums, int start, vector<vector<int>>& result) {  if (start == nums.size()) {  result.push\_back(nums);  return;  }  for (int i = start; i < nums.size(); ++i) {  swap(nums[start], nums[i]);  generatePermutations(nums, start + 1, result);  swap(nums[start], nums[i]);  }  }  vector<vector<int>> permute(vector<int>& nums) {  vector<vector<int>> result;  generatePermutations(nums, 0, result);  return result;  }  bool isMagicSquare(const vector<int>& nums) {  int n = (int)sqrt(nums.size());  int magicSum = n \* (n \* n + 1) / 2;  for (int i = 0; i < n; ++i) {  int rowSum = 0;  for (int j = 0; j < n; ++j)  rowSum += nums[i \* n + j];  if (rowSum != magicSum)  return false;  }  for (int i = 0; i < n; ++i) {  int colSum = 0;  for (int j = 0; j < n; ++j)  colSum += nums[j \* n + i];  if (colSum != magicSum)  return false;  }  int diagSum = 0;  for (int i = 0; i < n; ++i)  diagSum += nums[i \* n + i];  if (diagSum != magicSum)  return false;  diagSum = 0;  for (int i = 0; i < n; ++i)  diagSum += nums[i \* n + (n - 1 - i)];  if (diagSum != magicSum)  return false;  return true;  }  int main() {  setlocale(LC\_ALL, "Russian");  int n = 0;  while (pow((int)sqrt(n), 2) != n || n == 0) {  cout << "Введите количество элементов в матрице: ";  cin >> n;  }  system("cls");  vector<int> nums(n);  for (int i = 1; i <= n; ++i) {  nums[i - 1] = i;  cout << i << ' ';  }  cout << endl;  vector<vector<int>> permutations = permute(nums);  cout << permutations.size() << endl;  float i = 0;  vector<int> magic;  for (const auto perm : permutations) {  if (isMagicSquare(perm))  magic.push\_back(i);  i++;  }  for (int i = 0; i < magic.size(); i++) {  for (int j = 0; j < n; j++) {  if (j % (int)sqrt(nums.size()) == 0)  cout << endl;  cout << permutations[magic[i]][j] << ' ';  }  cout << endl;  }  return 0;  } |

OUTPUT

|  |
| --- |
|  |