Федеральное агентство по образованию

Государственное общеобразовательное учреждение высшего профессионального образования

**Пермский государственный технический университет**

**Лабораторная работа:**

Двумерные массивы

Выполнил:

студент группы РИС-23-2б

Колосов Данил Евгеньевич

Проверила:

доцент кафедры ИТАС

Ольга Андреевна Полякова

2024 г.

Дано: двумерный массив m целых чисел

Необходимо: последовательно

рассматривая 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

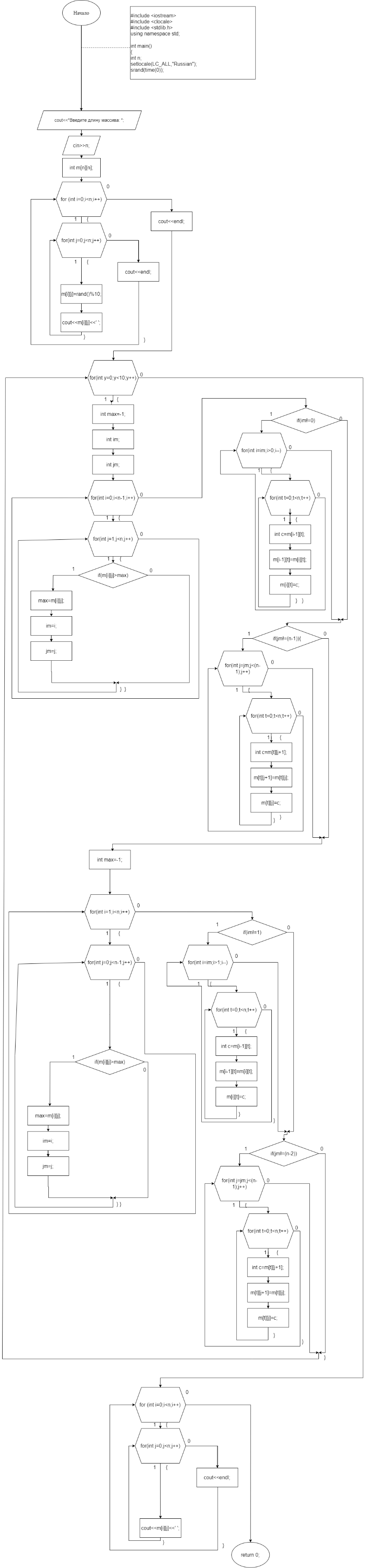
переместить в правый верхний угол подмассива. Проверить

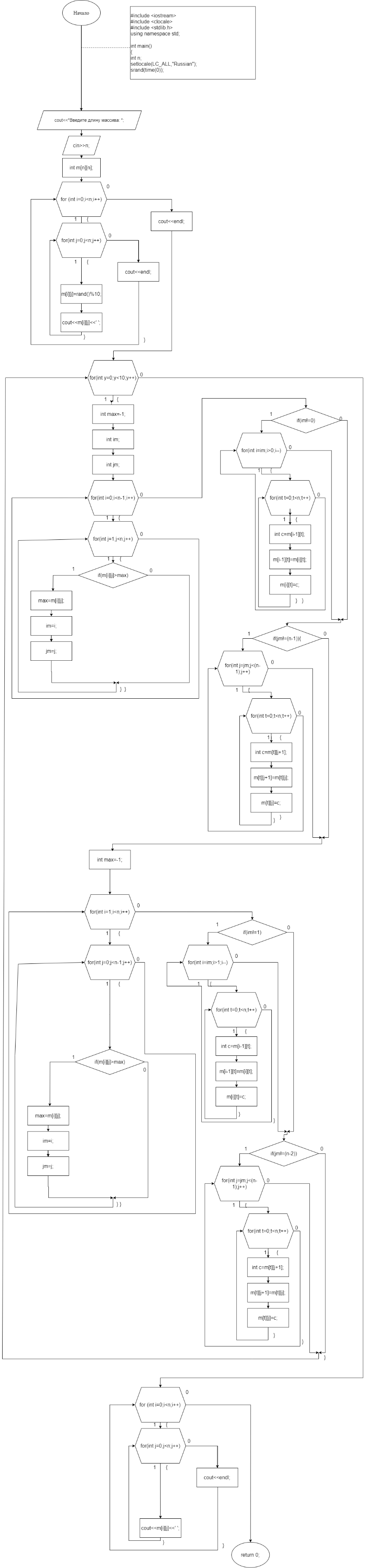
получилась ли на побочной диагонали убывающая

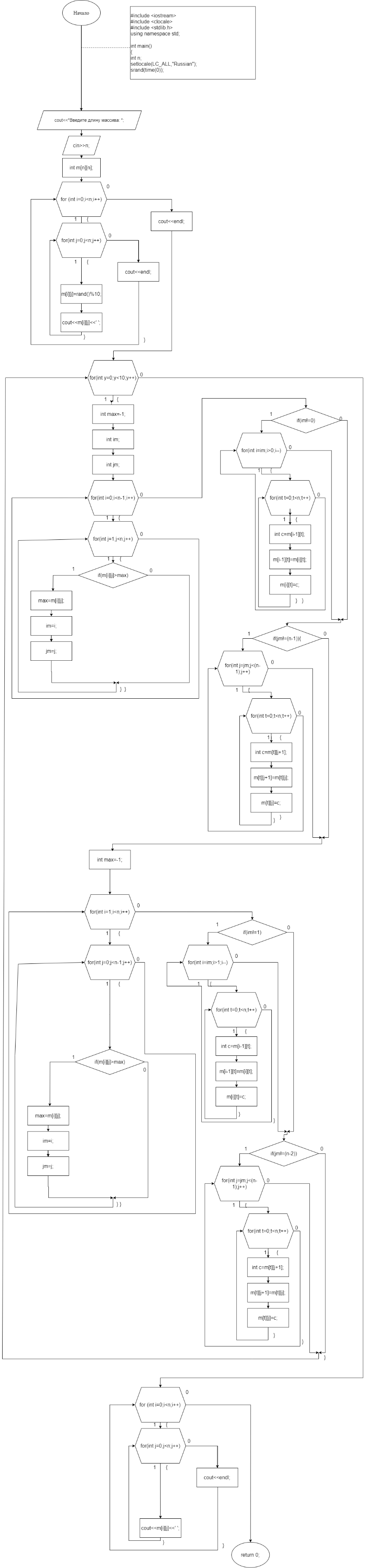
последовательность элементов.

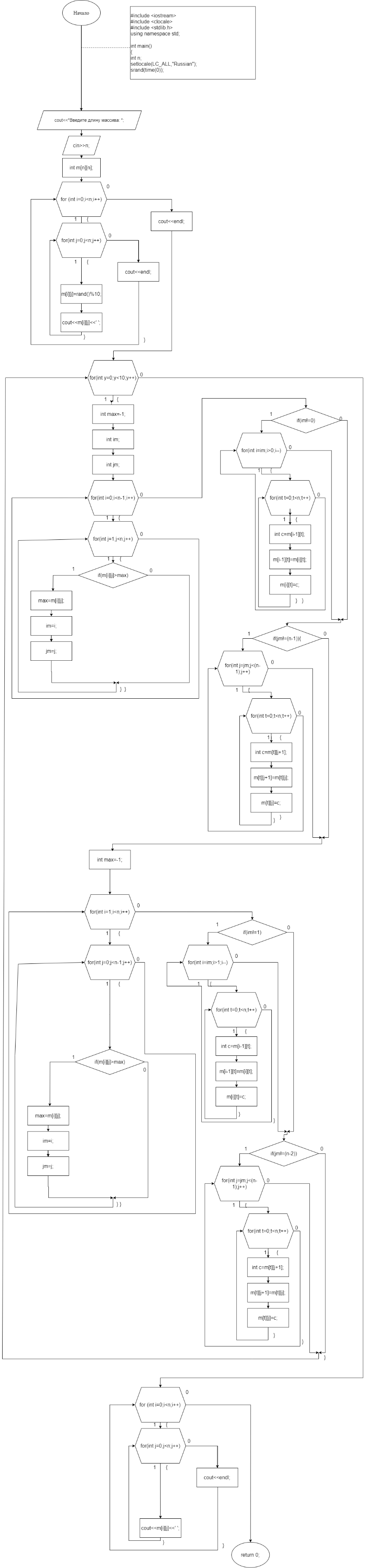
Построение алгоритма

1. За максимальный элемент примем первый элемент подмассива.
2. Дальше будем сравнивать его со всеми последующими элементами и если найдётся элемент больше максимального, то в переменную, отвечающую за хранения максимального элемента, будет записан найденный, таким же образом сохранятся индексы этого элемента.
3. Если нужный максимальный элемент находиться не не а своём месте, то переставляем строки и столбцы, пока он не займёт подлежащее ему место.

Блок схема







Код программы

#include <iostream>

#include <clocale>

#include <stdlib.h>

using namespace std;

int main()

{

int n;

setlocale(LC\_ALL,"Russian");

srand(time(0));

cout<<"Введите длину массива: ";

cin>>n;

int m[n][n];

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

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

m[i][j]=rand()%10;

cout<<m[i][j]<<' ';

}

cout<<endl;

}

cout<<endl;

for(int y=0;y<10;y++)

{int max=-1;

int im;

int jm;

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

for(int j=1;j<n;j++){

if(m[i][j]>max){

max=m[i][j];

im=i;

jm=j;}

}}

if(im!=0){

for(int i=im;i>0;i--){

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

int c=m[i-1][t];

m[i-1][t]=m[i][t];

m[i][t]=c;

}

}

}

//cout<<im<<' '<<jm<<endl;

if(jm!=(n-1)){

for(int j=jm;j<(n-1);j++){

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

int c=m[t][j+1];

m[t][j+1]=m[t][j];

m[t][j]=c;

}

}

}

max=-1;

for(int i=1;i<n;i++){

for(int j=0;j<n-1;j++){

if(m[i][j]>max){

max=m[i][j];

im=i;

jm=j;}}}

if(im!=1){

for(int i=im;i>1;i--){

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

int c=m[i-1][t];

m[i-1][t]=m[i][t];

m[i][t]=c;

}

}

}

if(jm!=(n-2)){

for(int j=jm;j<(n-2);j++){

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

int c=m[t][j+1];

m[t][j+1]=m[t][j];

m[t][j]=c;

}

}

}

//cout<<im<<' '<<jm<<endl;}

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

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

cout<<m[i][j]<<' ';

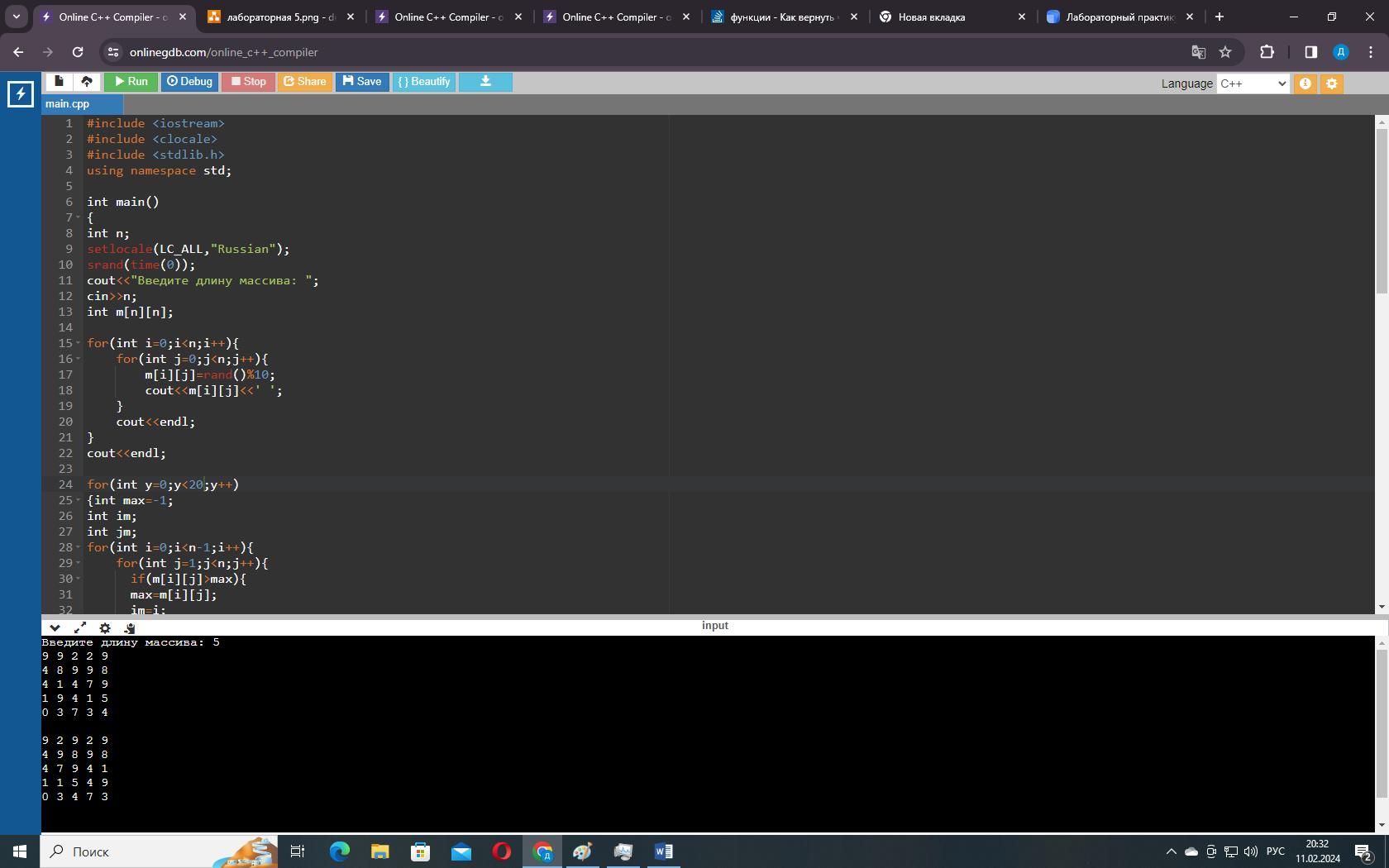
}

cout<<endl;

}

return 0;

}}

Результаты работы

Ссылка на гит