Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

Дисциплина: «Основы алгоритмизации и программирования»

Лабораторные № 4 и № 5

Семестр 2

Выполнил работу

Студент группы РИС-22-1Б

Мухаров Тимур Рустамович

Проверил

Доцент кафедры ИТАС

Полякова Ольга Андреевна

Г. Пермь-2023

**Постановка задачи(24 вариант)**

**Лабораторная № 4**

1. Реализовать с использованием массива однонаправленное кольцо (просмотр возможен слева направо, от последнего элемента можно перейти к первому).
2. Распечатать полученный массив, начиная с К-ого элемента и до К-1.
3. Упорядочить элементы по возрастанию
4. Удалить из кольца четные элементы.
5. Распечатать полученный массив, начиная с К-ого элемента и до К-1.

**Лабораторная № 5**

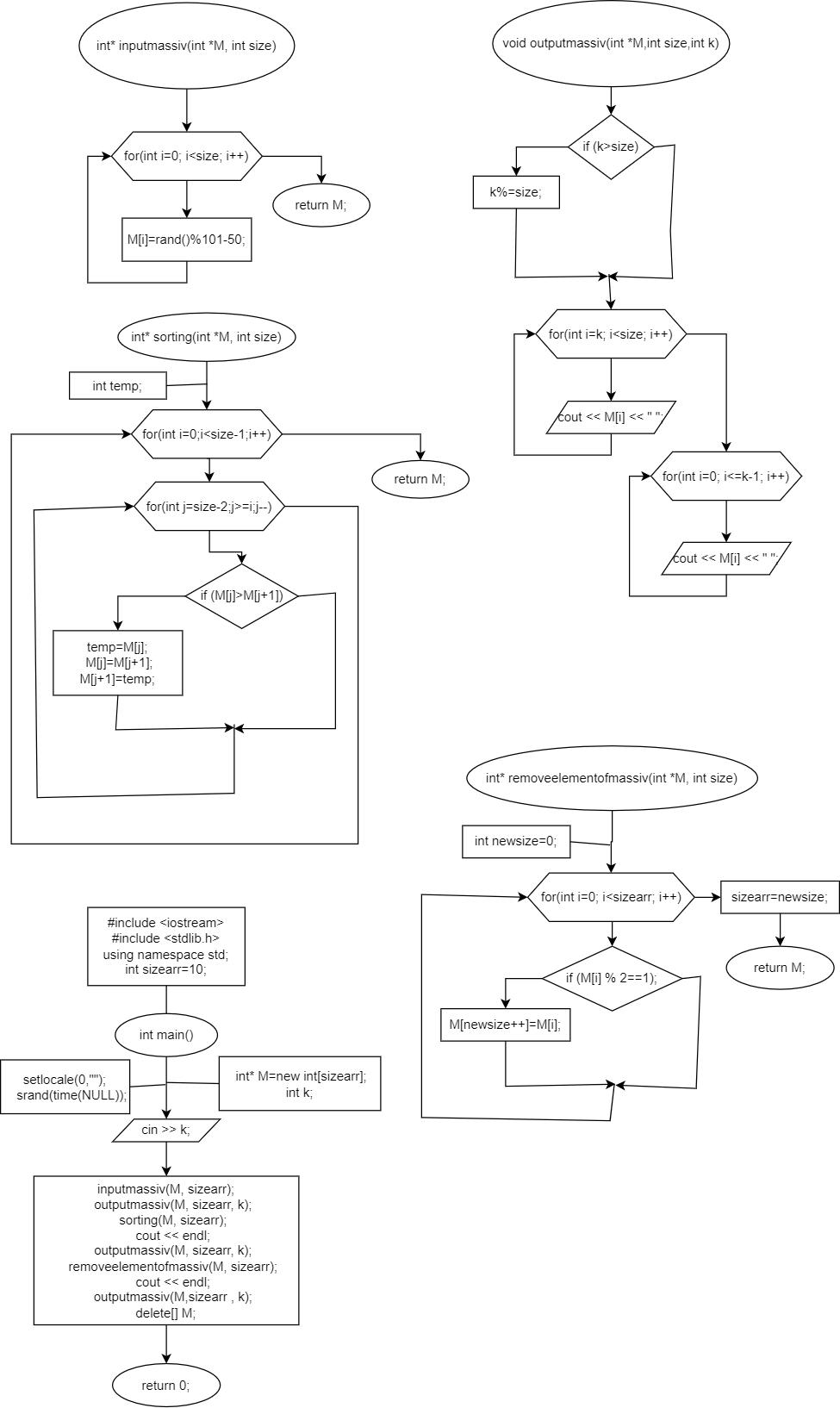
Определить сколько элементов двумерного массива больше любого элемента на главной диагонали.

**Анализ решения**

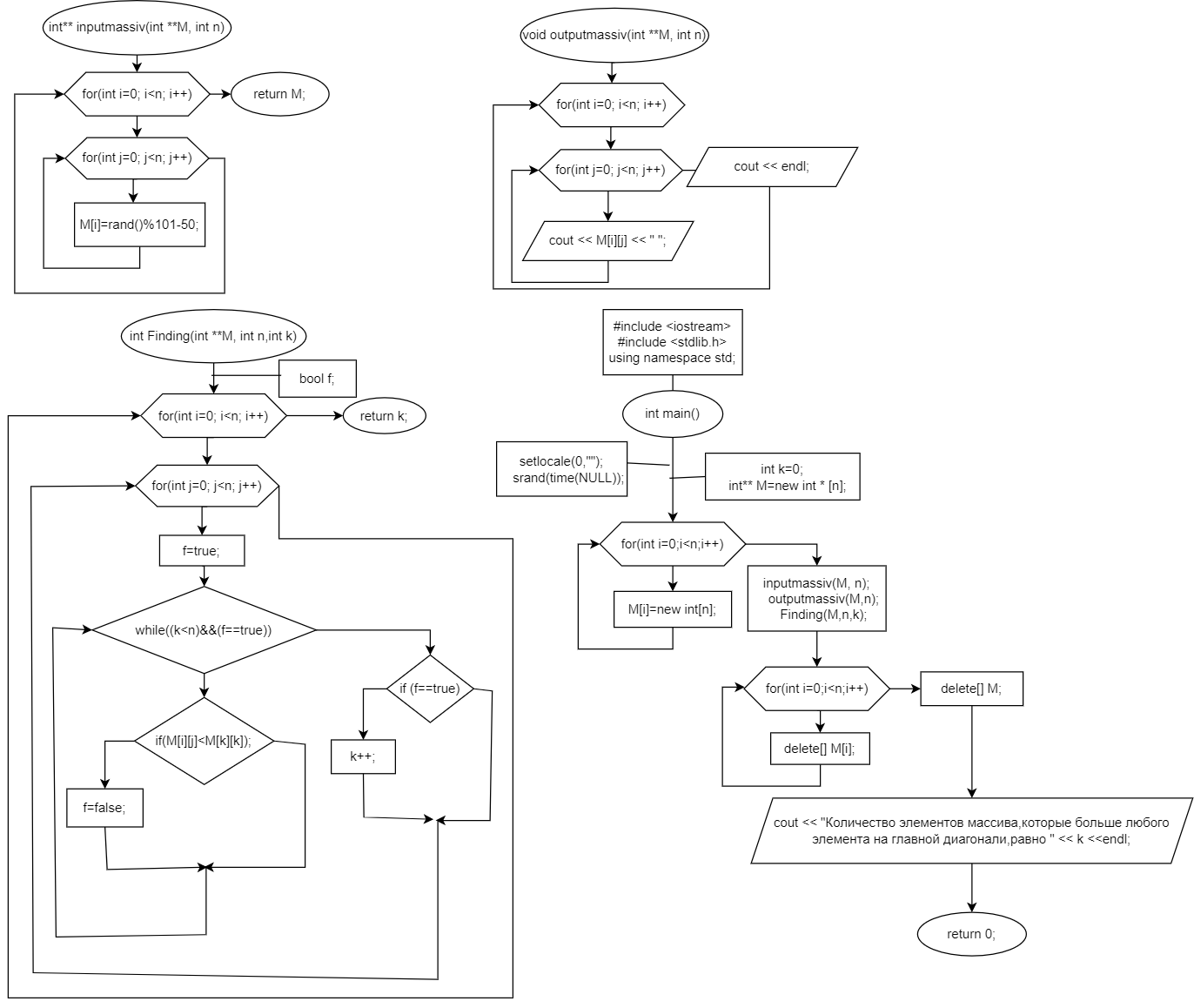
В лабораторной 4 одномерный массив. Функции для заполнения массива, вывода массива, сортировки и удаления элементов массива. В лабораторной 5 двухмерный массив. Функции для заполнения массива и выполнения задачи.

**Алгоритм программы**

**Лабораторная 4**



**Лабораторная 5**

****

Код программы:

Лабораторная 4

#include <iostream>

#include <stdlib.h>

using namespace std;

int sizearr=10;

int\* inputmassiv(int \*M, int size)//заполнение массива

{

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

{

M[i]=rand()%101-50;

}

return M;

}

void outputmassiv(int \*M,int size,int k)//вывод массива с k элемента до k-1 элемента

{

if (k>size)

k%=size;

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

{

cout << M[i] << " ";

}

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

{

cout << M[i] << " ";

}

}

int\* sorting(int \*M, int size)//сортировка пузырьком

{

int temp;

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

{

for(int j=size-2;j>=i;j--)

{

if (M[j]>M[j+1])

{

temp=M[j];

M[j]=M[j+1];

M[j+1]=temp;

}

}

}

return M;

}

int\* removeelementofmassiv(int \*M, int size)//удаление четных элементов

{

int newsize=0;

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

{

if (M[i] % 2==1);

{

M[newsize++]=M[i];

}

}

sizearr=newsize;

return M;

}

int main()

{

int\* M=new int[sizearr];

setlocale(0,"");

srand(time(NULL));

int k;

cin >> k;

inputmassiv(M, sizearr);

outputmassiv(M, sizearr, k);

sorting(M, sizearr);

cout << endl;

outputmassiv(M, sizearr, k);

removeelementofmassiv(M, sizearr);

cout << endl;

outputmassiv(M,sizearr , k);

delete[] M;

return 0;

}

Лабораторная 5

#include <iostream>

#include <stdlib.h>

using namespace std;

int n=5;

int\*\* inputmassiv(int \*\*M, int n)//заполнение массива

{

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

{

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

{

M[i][j]=rand()%101-50;

}

}

return M;

}

void outputmassiv(int \*\*M, int n)//заполнение массива

{

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

{

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

{

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

}

cout << endl;

}

}

int Finding(int \*\*M, int n,int k)//нахождения сколько элементов больше любого элемента главной диагонали

{

bool f;

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

{

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

{

f=true;

while((k<n)&&(f==true))

{

if(M[i][j]<M[k][k]);

f=false;

}

if (f==true)

k++;

}

}

return k;

}

int main()

{

setlocale(0,"");

int k=0;

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

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

{

M[i]=new int[n];

}

srand(time(NULL));

inputmassiv(M, n);

outputmassiv(M,n);

Finding(M,n,k);

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

{

delete[] M[i];

}

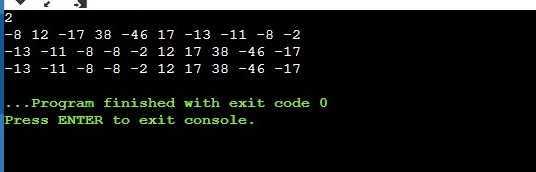
delete[] M;

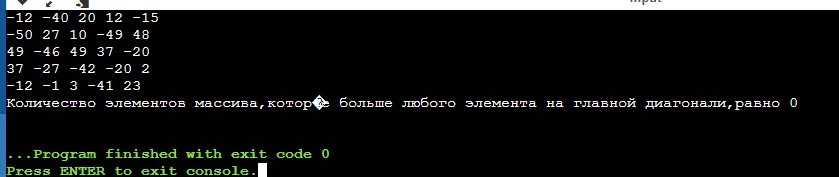
cout << "Количество элементов массива,котоpыe больше любого элемента на главной диагонали,равнo " << k <<endl;

return 0;

}

**Результаты работы программы**

****



**Анализ результатов**

Как видно из результатов работы программ, в пятой лабораторной если на главной диагонали очень большое число, то количество с большей вероятностью равно нулю, в четвертой лабораторной все нужные функции работают.