Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(ПНИПУ)

ОТЧЁТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

по теме:

РАБОТА С ОДНОМЕРНЫМИ МАССИВАМИ

Выполнила: студентка группы РИС-22-1б

Черкасова А.А\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверил: доцент кафедры ИТАС

Полякова О.А\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_/\_\_\_\_\_\_\_/\_\_\_\_\_г.

Пермь 2023

**Содержание**

[**Введение** 3](#_Toc127366278)

[**Анализ задачи** 3](#_Toc127366279)

[**Блок – схема** 5](#_Toc127366280)

[**Приложение А** 10](#_Toc127366281)

[**Приложение Б** 12](#_Toc127366282)

# **Введение**

**Постановка задачи:**

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

**Цель:** Получение навыков обработки одномерных массивов.

Для достижения поставленной цели, необходимо решить следующие **задачи**:

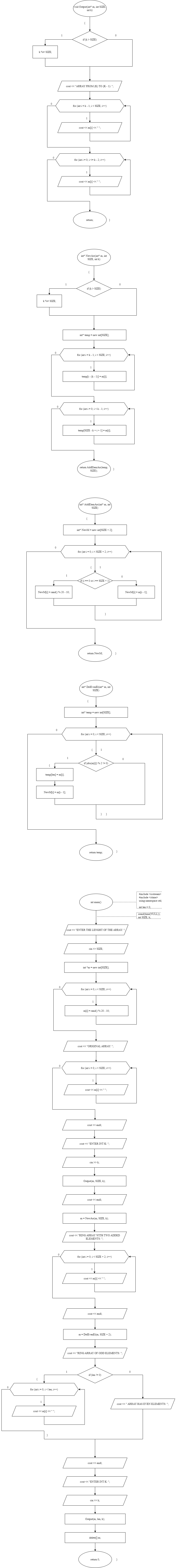
* Провести анализ задачи
* Реализовать задачу на языке С++
* Составить блок-схему

# **Анализ задачи**

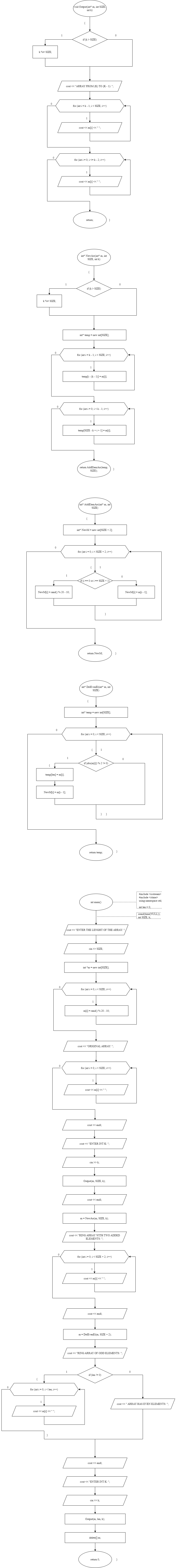
1. Чтобы распечатать массив начиная с k-ого элемента и до k-1 необходимо использовать два цикла *for*, первый будет выводить элементы от k-1 до конца массива, второй – с начала массива до k-1. Данный алгоритм реализован в функции *Output*
2. В функции *NewArr* происходит «развёртка» кольца с k-ого элемента и до k-1. Преобразование его в массив. Алгоритм схож с алгоритмом в функции *Output*. Также здесь происходит вызов функции *AddElemArr,* которая добавляет в переданный, в качестве параметра, массив случайные элементы в начало и конец массива.
3. Функция *DelEvenEl* удаляет четные элементы из массива. Здесь создаётся «новый» массив, который будет содержать в себе нечётные элементы. С помощью цикла *for* выбираются нечетные элементы из переданного массива и добавляются в новый массив.
4. В функции *main* создаётся массив случайных чисел. Также здесь вызываются функции, описанные ранее. В качестве параметра помещается ранее созданный массив.
5. После завершения основного алгоритма, удаляется массив для высвобождения памяти.

# **Блок – схема**

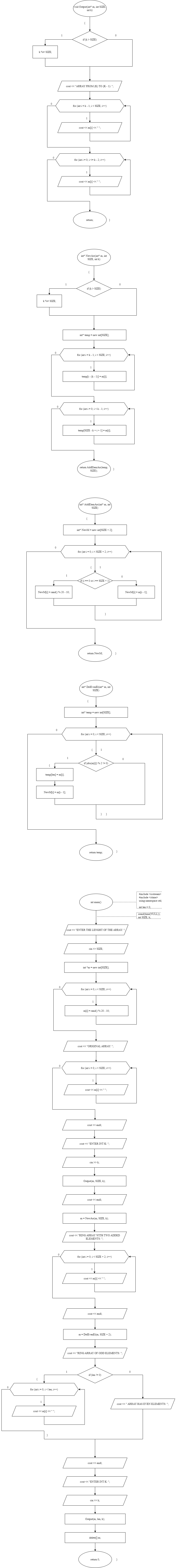
A) Функция *Output:*

**

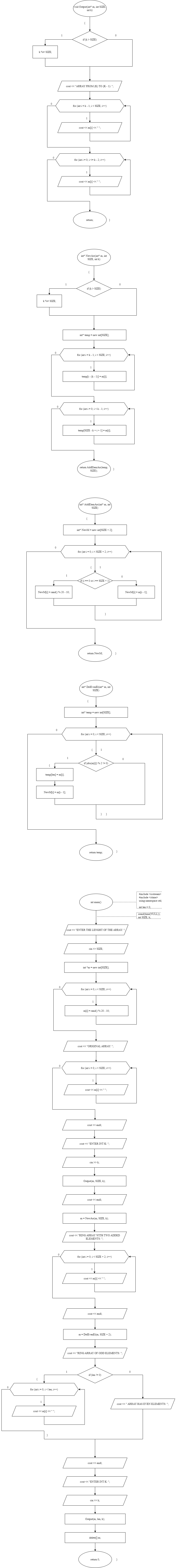
B) Функция *NewArr:*

**

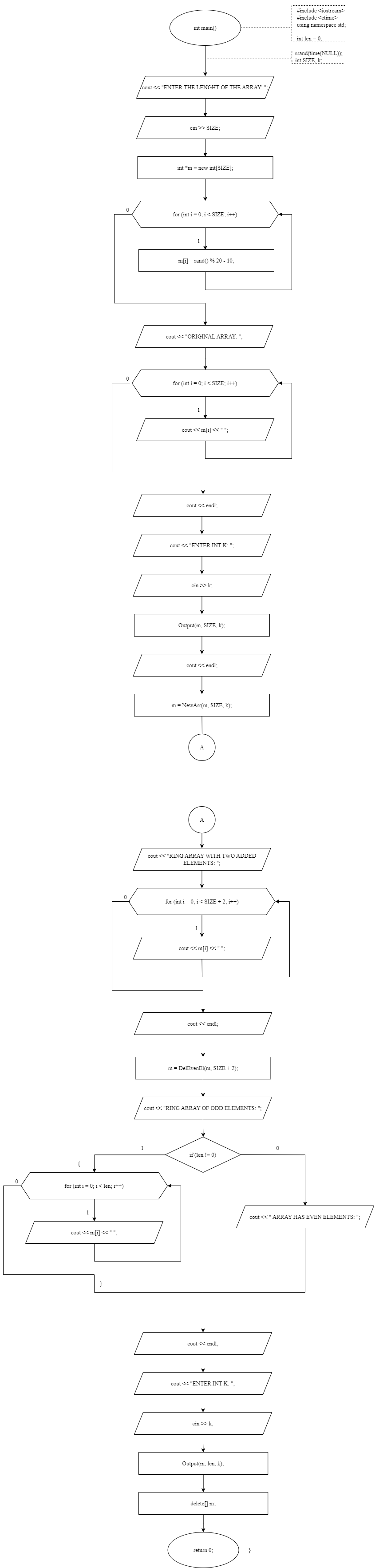
C) Функция *AddElemArr:*

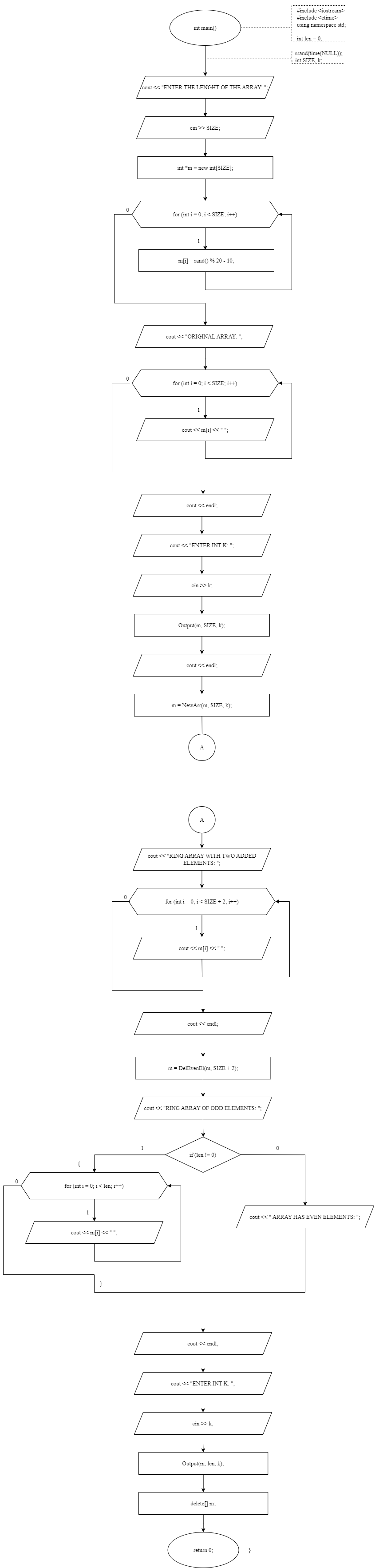
**

D) Функция *DelEvenEl:*

**

E) Функция *main:*

**



# **Приложение А**

**Листинг программы**

#include <iostream>

#include <ctime>

using namespace std;

int len = 0; // длина массива после удаления элементов

//Распечатать полученный массив, начиная с К-ого элемента и до К-1.

void Output(int\* m, int SIZE, int k) // вывод с k-го элемента до k-1

{

if (k > SIZE) k %= SIZE;

cout << "ARRAY FROM (K) TO (K - 1): ";

for (int i = k - 1; i < SIZE; i++) cout << m[i] << " "; // k - 1 т.к. k - номер

for (int i = 0; i <= k - 2; i++) cout << m[i] << " "; // k - 2 = (k - 1) - 1;

}

int\* AddElemArr(int\*, int );

// разворачивание кольца

int\* NewArr(int\* m, int SIZE, int k) // Ring from к to к - 1

{

int\* temp = new int[SIZE];

if (k > SIZE) k %= SIZE;

for (int i = k - 1; i < SIZE; i++) temp[i - (k - 1)] = m[i]; // k - 1 т.к. k - номер

for (int i = 0; i < k - 1; i++) temp[SIZE - k + i + 1] = m[i]; // k - 1;

return AddElemArr(temp, SIZE);

}

//Добавить в заданный массив первый и последний элементы

int\* AddElemArr(int\* m, int SIZE)

{

int\* NewM = new int[SIZE + 2]; // создание массива с добавлением 2х элементов к оригинальному массиву, а не к массиву от к до к - 1, т.к если

// {0, 5, 11, 18,, -8, 9} -> {{0, 5,кон, нач, 11, 18,, -8, 9}, то нач != k, а нач = k + 1 и порядок нарушается

for (int i = 0; i < SIZE + 2; i++)

{

if (i == 0 or i == SIZE + 1)

{

NewM[i] = rand() % 20 - 10;

}

else

{

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

}

}

return NewM;

}

//Удалить из кольца четные элементы.

int\* DelEvenEl(int\* m, int SIZE)

{

int\* temp = new int[SIZE];

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

{

if (abs(m[i]) % 2 != 0)

{

temp[len] = m[i];

len++;

}

}

return temp;

}

int main()

{

srand(time(NULL));

int SIZE, k; // k номер эл-та с которого начнется вывод

cout << "ENTER THE LENGHT OF THE ARRAY: ";

cin >> SIZE;

int \*m = new int[SIZE]; // массив

for (int i = 0; i < SIZE; i++) m[i] = rand() % 20 - 10;

cout << "ORIGINAL ARRAY: ";

for (int i = 0; i < SIZE; i++) cout << m[i] << " "; // вывод заданного массива

cout << endl;

cout << "ENTER INT K: ";

cin >> k;

Output(m, SIZE, k); // вывод массива от k до k -1

cout << endl;

m = NewArr(m, SIZE, k);

cout << "RING ARRAY WITH TWO ADDED ELEMENTS: ";

for (int i = 0; i < SIZE + 2; i++) cout << m[i] << " ";

cout << endl;

m = DelEvenEl(m, SIZE + 2);

cout << "RING ARRAY OF ODD ELEMENTS: ";

if (len != 0)

{

for (int i = 0; i < len; i++) cout << m[i] << " ";

}

else

cout << " ARRAY HAS EVEN ELEMENTS: ";

cout << endl;

cout << "ENTER INT K: ";

cin >> k;

Output(m, len, k);

delete[] m;

}

# **Приложение Б**

**Результаты выполнения программы**

