Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

ПНИПУ

**ЛАБОРАТОРНАЯ РАБОТА**

«**Статичные одномерные массивы**»

**Выполнил:**

студент группы РИС-23-2б

Борисов Никита Андреевич

**Проверила:**

доцент кафедры ИТАС

О.А. Полякова

Пермь, 2024 г.

**ПОИСК МАСИМАЛЬНЫХ ЧИСЕЛ В МАССИВЕ.**

*Постановка задачи.*

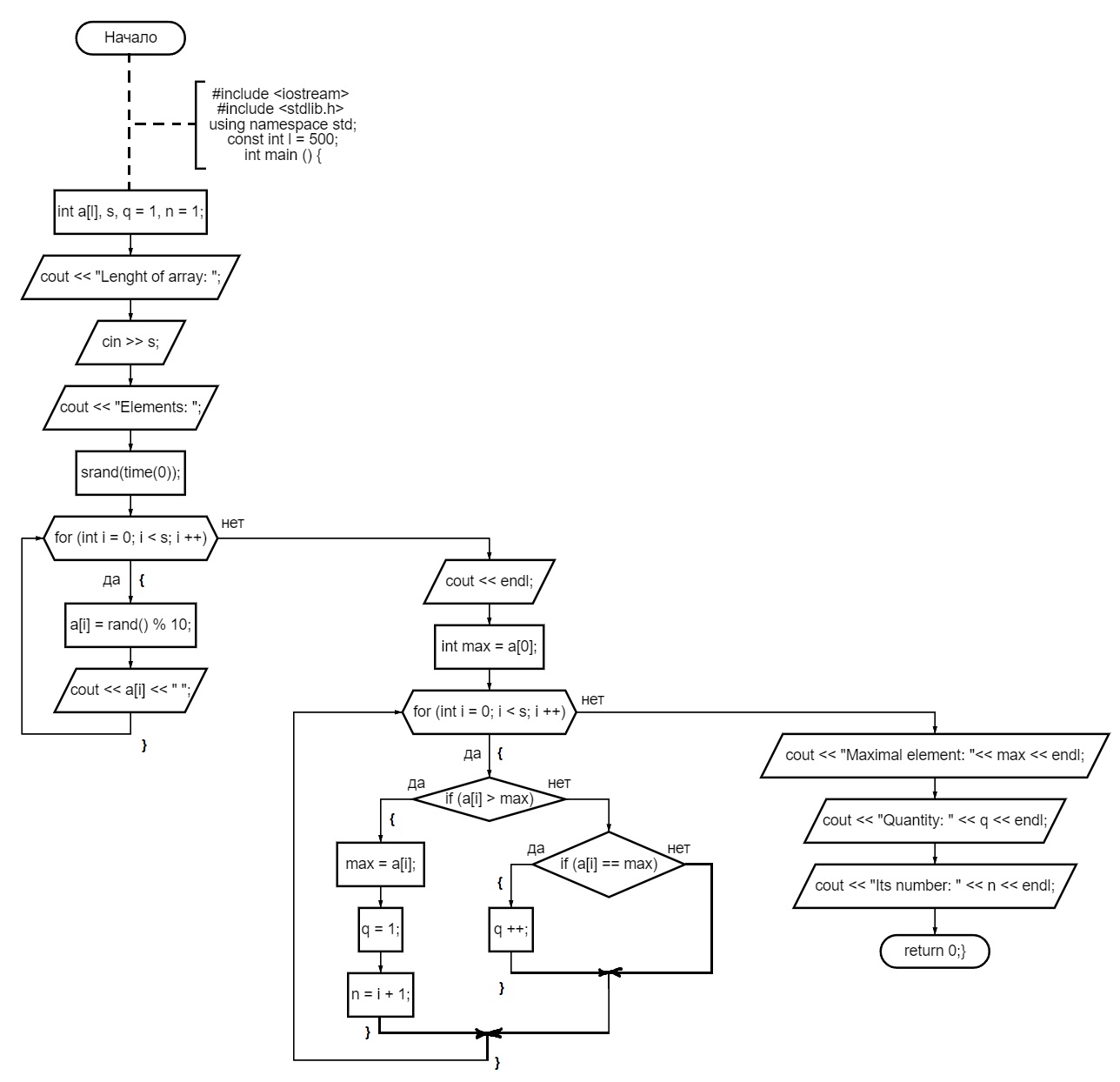
*Дан* массив целых чисел.

*Найти* максимальный элемент, его номер и, если максимальных элементов несколько, то посчитать их количество.

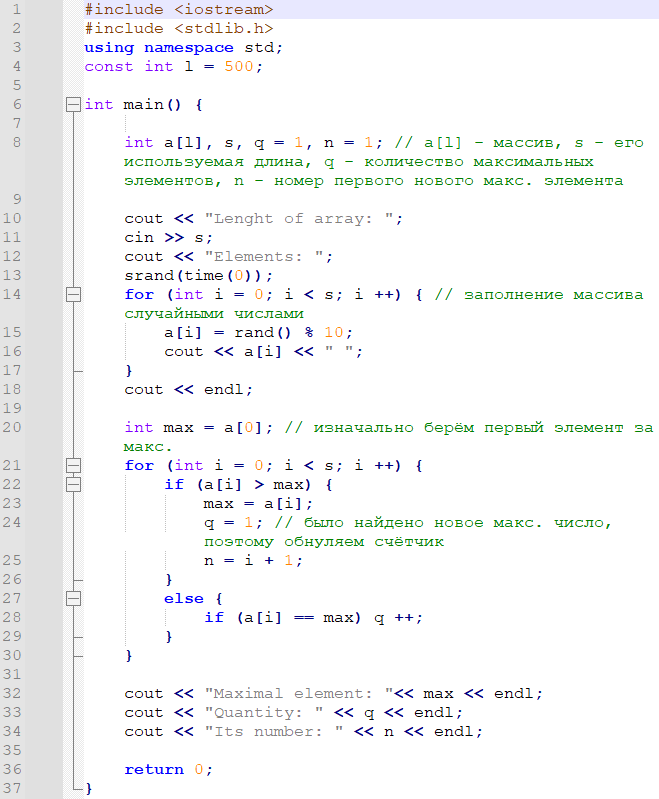
*Анализ задачи.*

1. Изначально первый элемент массива возьмём за максимальный.
2. Дальше будем сравнивать его со всеми последующими элементами и, если найдётся элемент больше текущего максимального, то в переменную, отвечающую за хранения максимального элемента, будет записан найденный.
3. При наличии нескольких максимальных, будет происходить их подсчёт. Когда будет обнаружен новый максимальный элемент, сбрасываем счётчик (исходное значение 1).

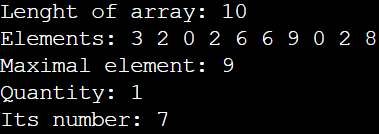
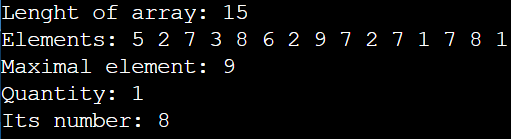
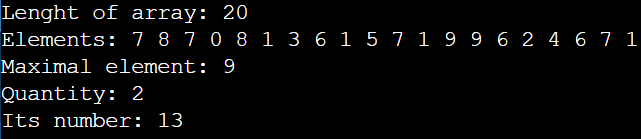
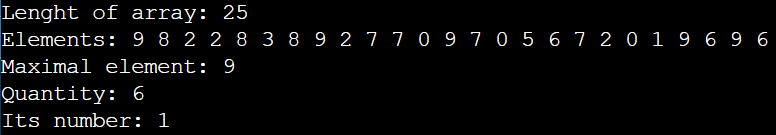
*Блок-схема со вписанным кодом.*

**

*Код программы.*



*Результаты работы программы.*

1. 
2. 
3. 
4. 

**ПРОВЕРКА НА УПОРЯДОЧЕННОСТЬ МАССИВА.**

*Постановка задачи.*

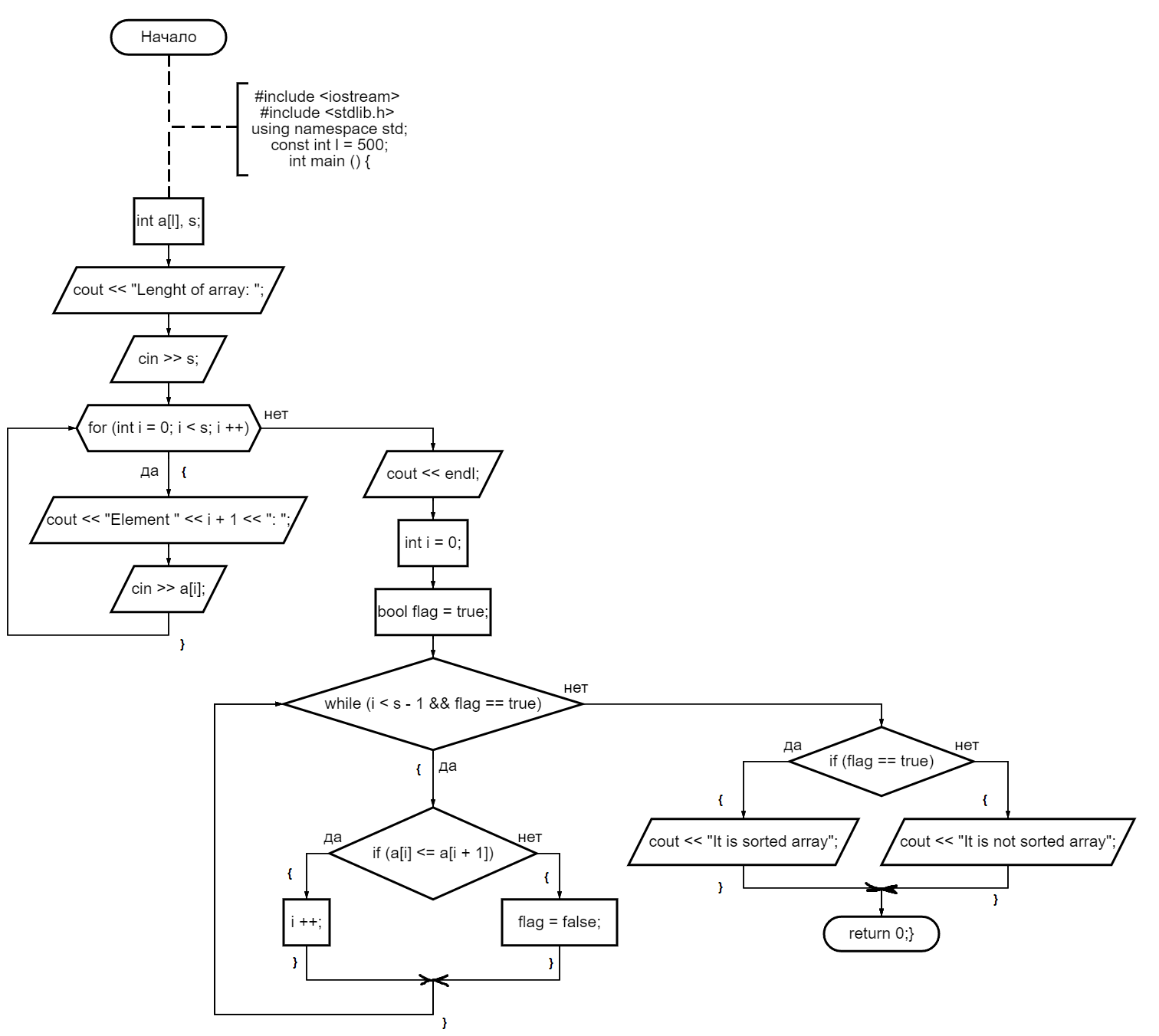
*Дан* массив целых чисел.

*Выяснить*, является ли массив неубывающей последовательностью чисел.

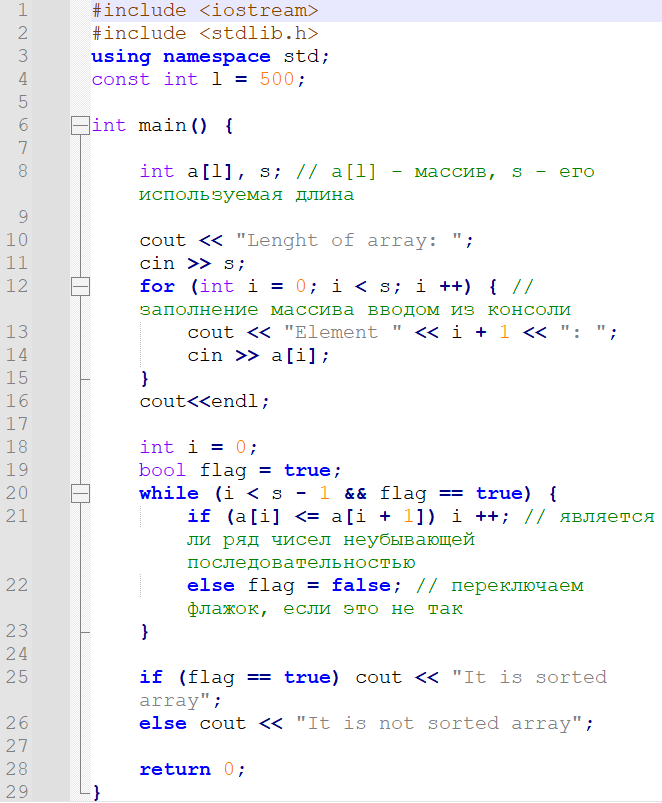
*Анализ задачи.*

1. Цикл является поисковым, т.к. существуют две причины окончания: массив не упорядоченный или произошёл полный перебор массива.
2. Задача с флажком, который меняет значение, если оказывается, что последующий элемент меньше предыдущего.

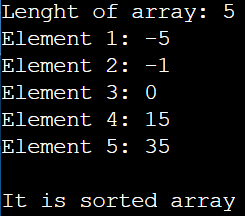
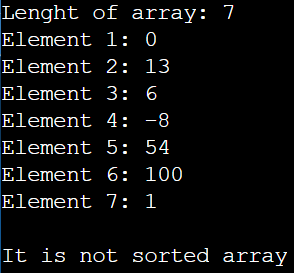
*Блок-схема со вписанным кодом.*

**

*Код программы.*



*Результаты работы программы.*

1. 
2. **РАБОТА С ТЕКУЩИМ РАЗМЕРОМ МАССИВА.**

*Постановка задачи.*

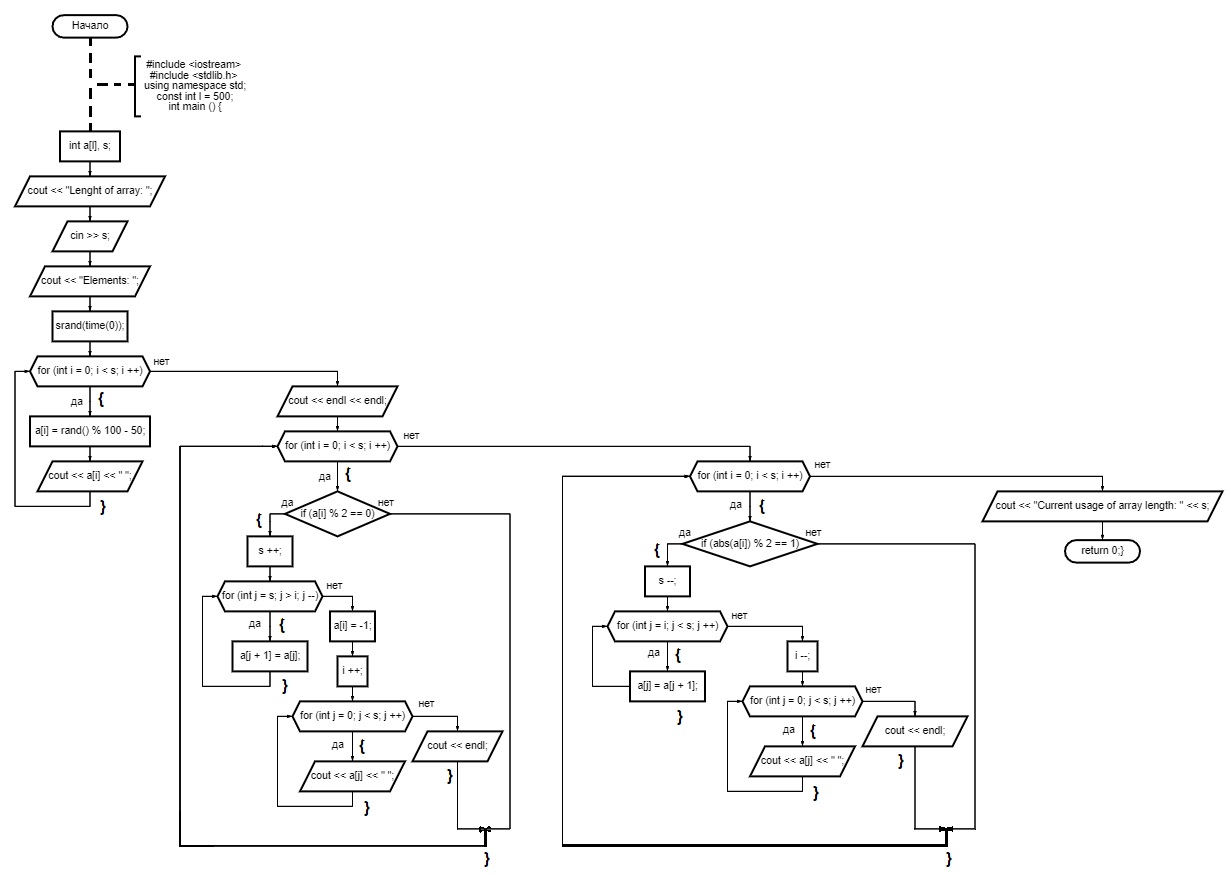
*Дан* массив целых чисел.

*Выполнить:* перед каждым чётным элементом массива поставить «-1»; из полученного массива удалить все нечётные элементы.

*Анализ задачи.*

1. Использование вложенных циклов.
   1. Внешний цикл отвечает за перебор элементов массива и проверку их на чётность или нечётность.
   2. Внутренний будет осуществлять сдвиг необходимых элементов либо вправо, чтобы была возможность поставить «-1» перед нужным элементом, не стирая стоящий перед ним, либо влево, когда необходимо удалять элементы из массива и сокращать его длину.

*Блок-схема со вписанным кодом.*

**

*Код программы.*

#include <iostream>

#include <stdlib.h>

using namespace std;

const int l = 500;

int main() {

int a[l], s;

cout << "Lenght of array: ";

cin >> s;

cout << "Elements: ";

srand(time(0));

for (int i = 0; i < s; i ++) { // заполнение массива случайными числами

a[i] = rand() % 100 - 50;

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

}

cout << endl << endl;

for (int i = 0; i < s; i ++) { // поставить перед всеми чётными числами "-1"

if (a[i] % 2 == 0) {

s ++;

for (int j = s - 1; j >= i; j --) { // сдвиг элементов вправо (сдвинутся все последующие за чётным)

a[j + 1] = a[j];

}

a[i] = -1;

i ++;

for (int j = 0; j < s; j ++) { // отображение изменений в массиве

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

}

cout << endl;

}

}

cout << "Current usage of array length: " << s << endl << endl;

for (int i = 0; i < s; i ++) { // удаление нечётных элементов ряда

if (abs(a[i]) % 2 == 1) {

s --;

for (int j = i; j < s; j ++) {

a[j] = a[j + 1];

}

i --;

for (int j = 0; j < s; j ++) { // отображение изменений в массиве

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

}

cout << endl;

}

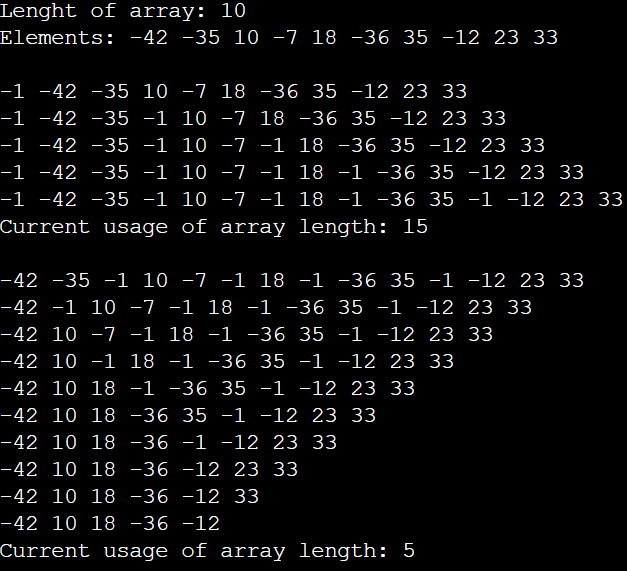
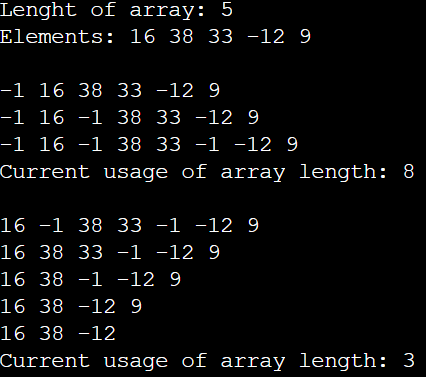
}

cout << "Current usage of array length: " << s;

return 0;

}

*Результаты работы программы.*

1. 
2. 

**ЛАБАРАТОРНАЯ РАБОТА №4. МАССИВ-КОЛЬЦО.**

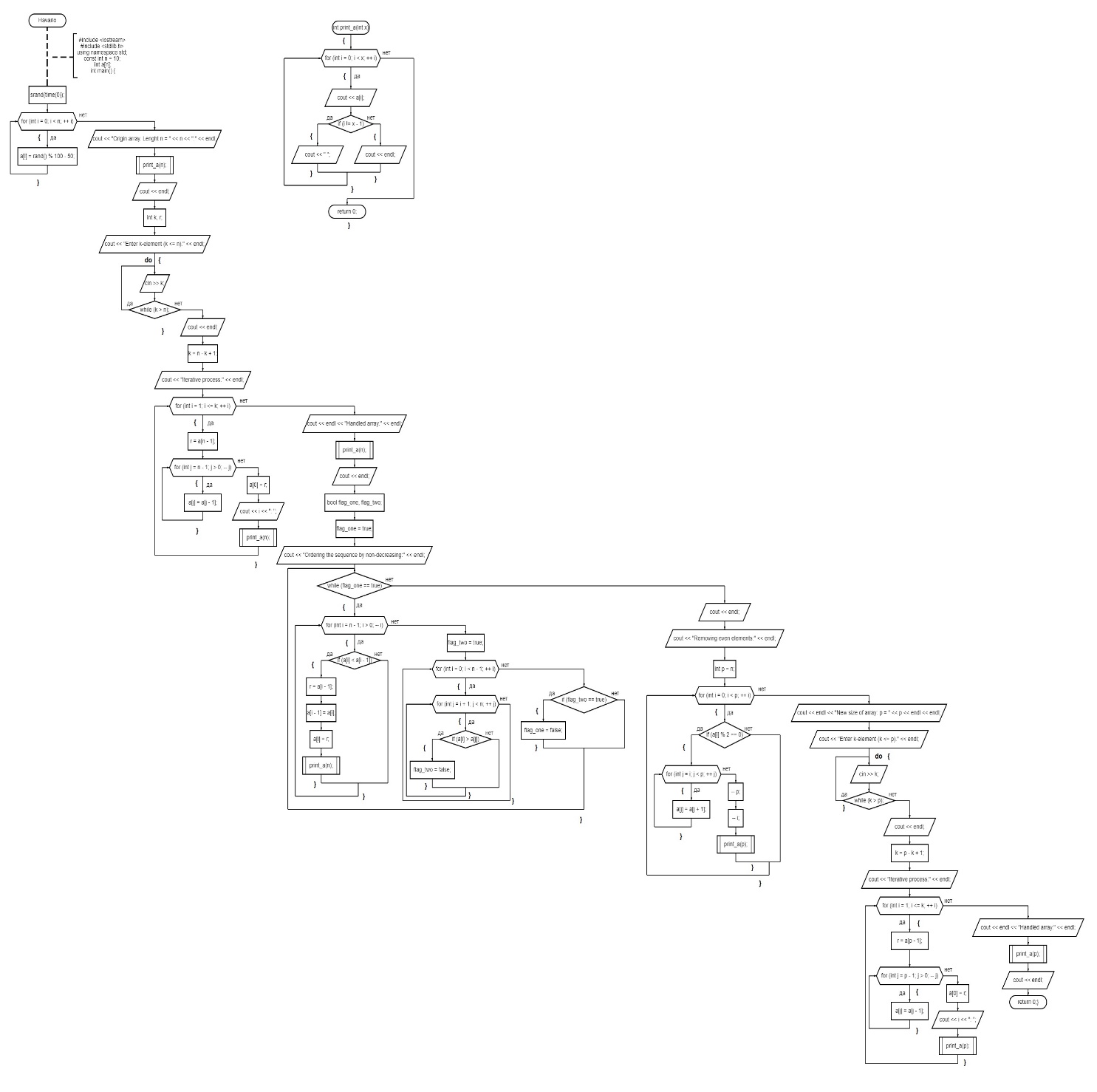
*Постановка задачи.*

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

*Анализ задачи.*

1. Реализация однонаправленного кольца, прокрутка которого возможна слева направо состоит в повторе действий: сдвиг всех элементов на одну позицию вправо, а также перемещение последнего элемента на первое место. Сдвиг осуществим с помощью цикла, который перебирает элементы не с начала, а с конца, иначе все элементы массива станут одинаковыми. Эти действия должны повториться на один раз больше, чем значение того номера элемента массива, с которого нужно по новой начать его после «прокрутки».
2. Для упорядочивания массива можно воспользоваться один из методов его сортировки, например, самым примитивным, методом «пузырька». В этом случае делаем просмотр n – 1 раз от первого элемента до того, который не исключён из просмотра (n – 1, т.к. первый остаётся на своём месте). Просматриваем два соседних элемента, двигаясь от начала к концу. Если a[i] > a[I + 1], то меняем из местами. Когда будет окончен первый проход, на последнем месте в массиве будет находится самый большой элемент. Исключаем его из обработки (с помощью индексов внутреннего цикла). Всего для сортировки стоит использовать два циклических процесса: внешний цикл, который отвечает за количество проходов (n-1); внутренний цикл, который отвечает за сравнения на каждом проходе.
3. Чтобы из полученного массива удалить все нечётные элементы, стоит применить два цикла: внешний цикл отвечает за перебор элементов массива и проверку их на нечётность; внутренний цикл производит сдвиг необходимых элементов влево и сокращает используемую длину.

*Блок-схема со вписанным кодом.*

**

*Код программы.*

#include <iostream>

#include <stdlib.h>

using namespace std;

const int n = 10;

int a[n];

// Функция вывода содержимого массива

int print\_a(int x) {

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

cout << a[i];

if (i != x - 1) {

cout << " ";

}

else {

cout << endl;

}

}

return 0;

}

int main() {

// Заполнение массива случайными числами

srand(time(0));

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

a[i] = rand() % 100 - 50;

}

cout << "Origin array. Lenght n = " << n << "." << endl;

print\_a(n);

cout << endl;

// Прокрутка массива слева направо: первым в конечном счёте должен быть элемент k из начального состояния, последним — k - 1

int k, r;

cout << "Enter k-element (k <= n)." << endl;

do {

cin >> k;

} while (k > n); // Элемент с номером k должен быть в рамках массива

cout << endl;

k = n - k + 1; // Расчёт количества прокруток

cout << "Iterative process." << endl;

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

r = a[n - 1];

for (int j = n - 1; j > 0; -- j) { // Перебор элементов идёт справа налево

a[j] = a[j - 1];

}

a[0] = r;

cout << i << ": ";

print\_a(n);

}

cout << endl << "Handled array." << endl;

print\_a(n);

cout << endl;

// Упорядочивание элементов массива по неубыванию

bool flag\_one, flag\_two;

flag\_one = true;

cout << "Ordering the sequence by non-decreasing:" << endl;

while (flag\_one == true) { // Цикл перемещает элементы по массиву

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

if (a[i] < a[i - 1]) {

r = a[i - 1];

a[i - 1] = a[i];

a[i] = r;

print\_a(n);

}

}

flag\_two = true;

for (int i = 0; i < n - 1; ++ i) { // Проверка на монотонность последовательности по неубыванию

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

if (a[i] > a[j]) {

flag\_two = false;

}

}

}

if (flag\_two == true) {

flag\_one = false;

}

}

cout << endl;

// Удаление элементов массива с чётным значением

cout << "Removing even elements:" << endl;

int p = n;

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

if (a[i] % 2 == 0) {

for (int j = i; j < p; ++ j) {

a[j] = a[j + 1];

}

-- p;

-- i;

print\_a(p);

}

}

cout << endl << "New size of array: p = " << p << endl << endl;

// Снова делаем прокрутку массива

cout << "Enter k-element (k <= p)." << endl;

do {

cin >> k;

} while (k > p); // Элемент с номером k должен быть в рамках массива

cout << endl;

k = p - k + 1; // Расчёт количества прокруток

cout << "Iterative process." << endl;

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

r = a[p - 1];

for (int j = p - 1; j > 0; -- j) { // Перебор элементов идёт справа налево

a[j] = a[j - 1];

}

a[0] = r;

cout << i << ": ";

print\_a(p);

}

cout << endl << "Handled array." << endl;

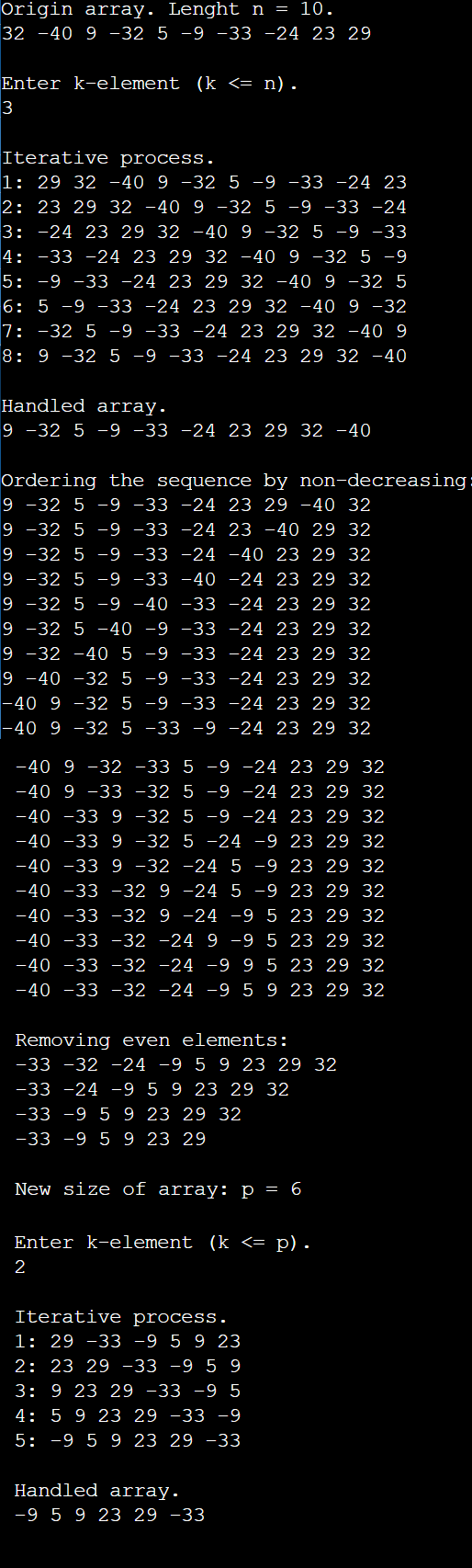
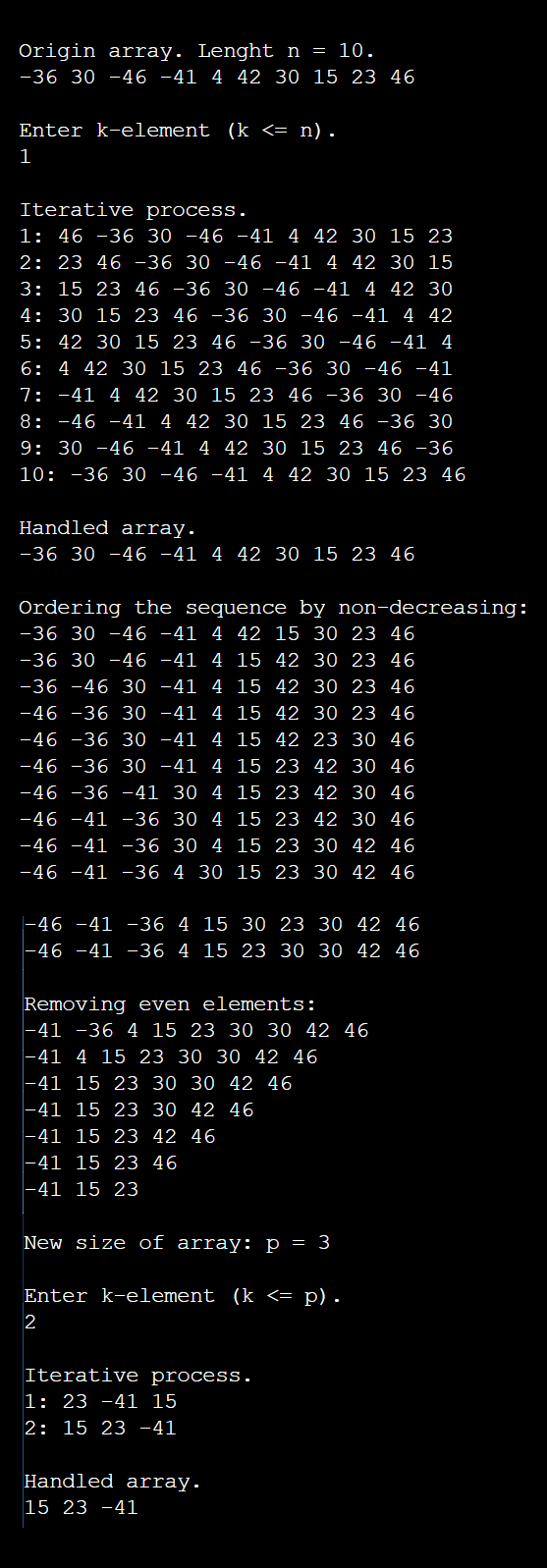
print\_a(p);

cout << endl;

return 0;

}

*Результаты работы программы.*

1. 
2. 

**ССЫЛКА НА ФАЙЛЫ РАБОТЫ.**

<https://github.com/Exateym/Study>