Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

Направление подготовки: 09.03.01 – «Информатика и вычислительная техника»

**Лабораторная работа № 4**

по дисциплине

**«Основы алгоритмизации и программирования»**

Выполнил студент гр. ИВТ-21-1б

Мифтахов Ильяс Гарифжанович

Проверил:

ст. преп. каф. ИТАС

Яруллин Денис Владимирович

(оценка) (подпись)

(дата)

**Постановка задачи**

Получение навыков обработки одномерных массивов.

**Задание**

1) Реализовать с использованием массива однонаправленное кольцо (просмотр возможен справа налево, от первого элемента можно перейти к последнему).

2) Распечатать полученный массив, начиная с К-ого элемента и до К+1.

3) Добавить в кольцо первый и последний элементы.

4) Удалить из кольца нечетные элементы.

5) Распечатать полученный массив, начиная с К-ого элемента и до К+1.

**Основная часть**

Текст программы представлен ниже:

#include<iostream>

#include<stdio.h>

using namespace std;

void create(int\*&arr, int size)

{

arr = new int[size]();//Динамический массив, заполненный нулями

int j = 9;//начало отсчета для массива

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

{

arr[i] = j;

cout << arr[i] << " ";

j--;

}//заполнили массив числами до 9 до 0

cout << endl;

}

void Print1(int\*&arr, int size, int k)

{

if (k <= size and k>=0)//чтоб не вылезти за память

{

for (int i = k; i <= k + 1; i++)//условия для выполнения цикла 2 раза

{

if (i - 1 == size) { i = 0; }//если предыдущий перед К И равен длине массива, то обнуляем И

cout << arr[i] << " ";//выводим нужное

}

cout << endl;

}

}

void AddFront(int\*&arr, int &size, int element)

{

int\*arr1 = new int[size + 1];//динамический массив с кол-вом символов больше на 1

arr1[0] = element;//нулевому элементу массива присвоить значение нашего элемента

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

{

arr1[i+1] = arr[i];//переписываем массив со сдвигом вправо, так как добавили элемент в начало

cout << arr1[i] << " ";

}

delete[]arr;

arr = arr1;

++size;

cout << endl;

}

void AddBack(int\*&arr, int &size, int element)

{

int\*arr2 = new int[size + 1];//аналогично прошлой функции

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

{

arr2[i] = arr[i];//переобозначили элементы старого массива в динамический

arr2[size] = element;//последнему свободному элементу присвоили значение элемента

cout << arr2[i] << " ";

}

delete[]arr;

arr = arr2;

++size;

cout << endl;

}

void Delete(int\*&arr, int &size, int k)

{

int \*arr3 = new int[size];//динамический массив

int NewArr3[12];//макс кол-во элементов нового массива, так как размер 12, вдруг все числа будут четные

int NewSize=0;//размер нового массива

int length = 0;//переменная для нового массива

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

{

arr3[i] = arr[i];//переобозначение старого массива в динамический

cout << arr3[i] << " ";

if (arr3[i] % 2 == 0)//если число четное,

{

NewArr3[length] = arr3[i];//то помещаем его в новый массив

length++;

}

}

cout << endl;

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

{

cout << NewArr3[i] << " ";//смотрим новый массив

}

delete[]arr;

arr = NewArr3;

cout << endl;

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

{

NewSize++;//считаем новый размер массива

}

cout << NewSize << endl;

size = NewSize;//меняем значение размера

if (k <= size and k >= 0)//вывод к и к+1 элемента осуществляем аналогично функции Print1

{

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

{

if (i - 1 == size) { i = 0; }

cout << arr[i] << " ";

}

cout << endl;

}

}

int main()

{

setlocale(LC\_ALL, "russian");

int \*arr = NULL;

int k = 1;

int element = 793;

int size = 10;

create(arr, size);

Print1(arr, size, k);

AddFront(arr, size, element);

AddBack(arr, size, element);

Delete(arr, size, k);

return 0;

}

Для корректной работы программы с указателями была подключена дополнительная директива препроцессора – <stdio.h>

Итак, для начала были подключены стандартные директивы препроцессора, указано пространство «std». Далее, идет первая функция, все функции будут с типом void, которая отвечает за создание нашего массива, принимает в себя такие значения, как массив и его размер. Вторая функция выводит к и к+1 элементы нашего массива. Третья функция добавляет 1 элемент в начало массива, расширяя его размер и далее сдвигаем весь массив вправо. Четвертая функция отвечает за добавление этого же элемента в конец. Все аналогично предыдущей. Пятая функция отвечает за то, чтобы убрать все нечетные элементы массива, записав четные в новый массив, а потом присвоив основному массиву значения нового, также, посчитан размер нового массива и выведены на экран к и к+1 элементы, аналогично второй функции. Все подфункции вызываются из int main(), все выполнено успешно, без ошибок.

Блок-схема данной программы представлена ниже, на рисунках 1 - 6.

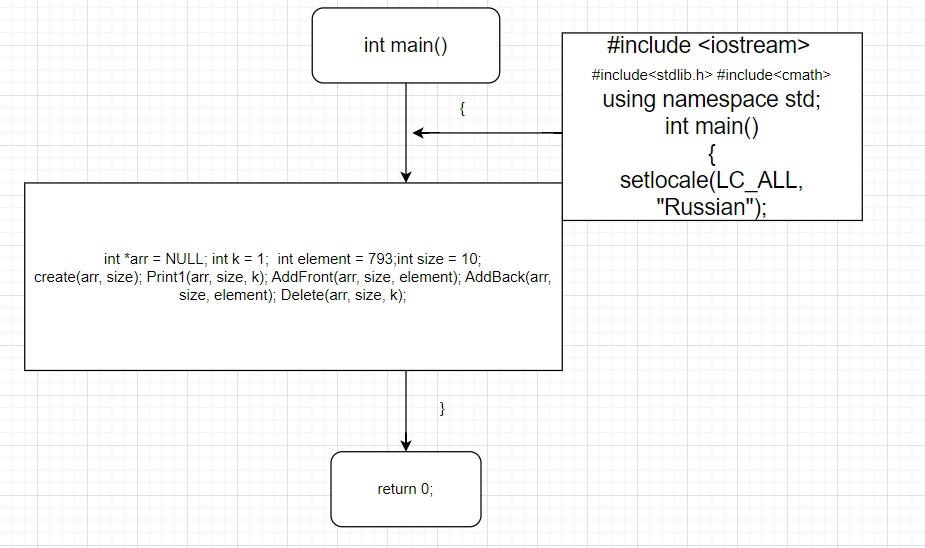


Рисунок 1

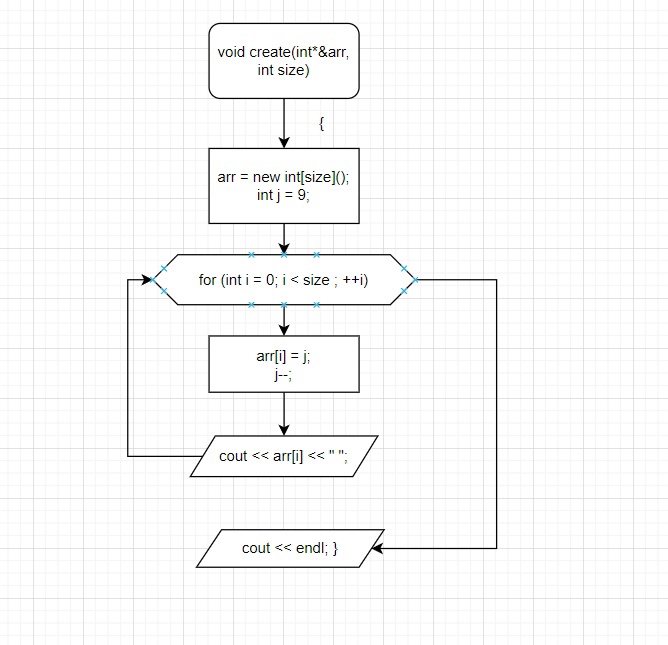


Рисунок 2

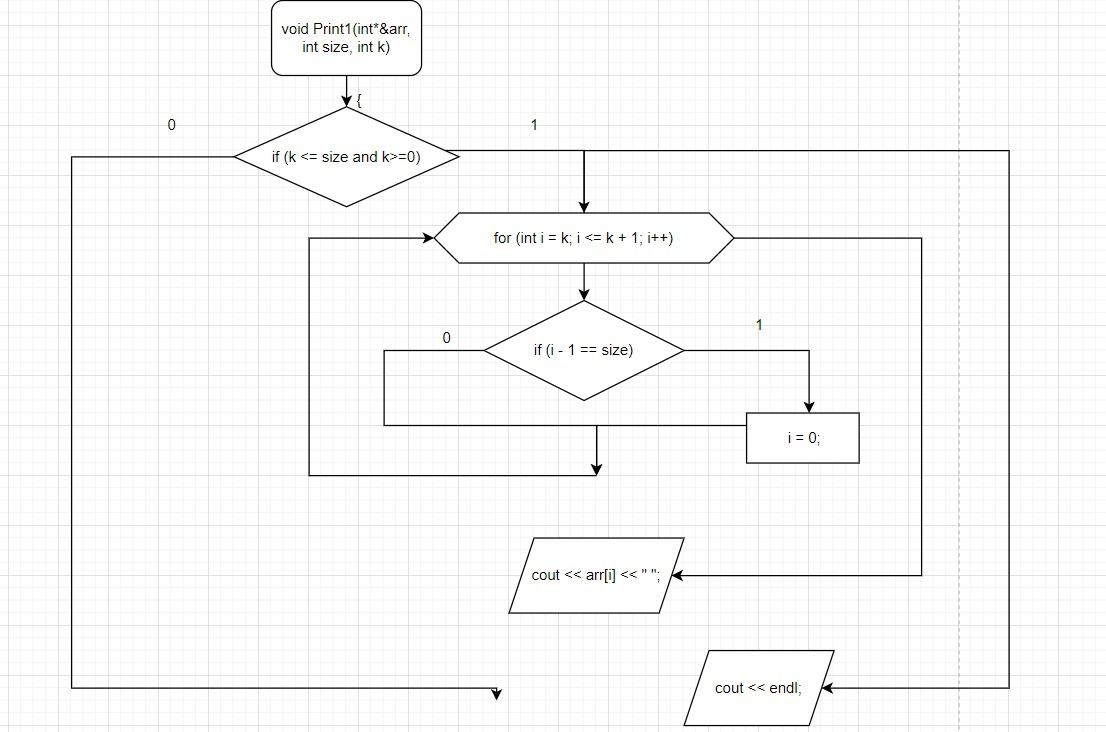


Рисунок 3

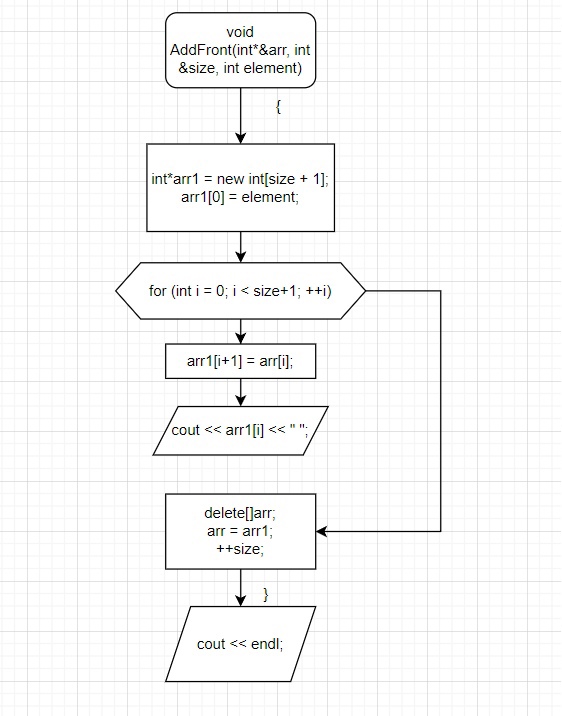


Рисунок 4

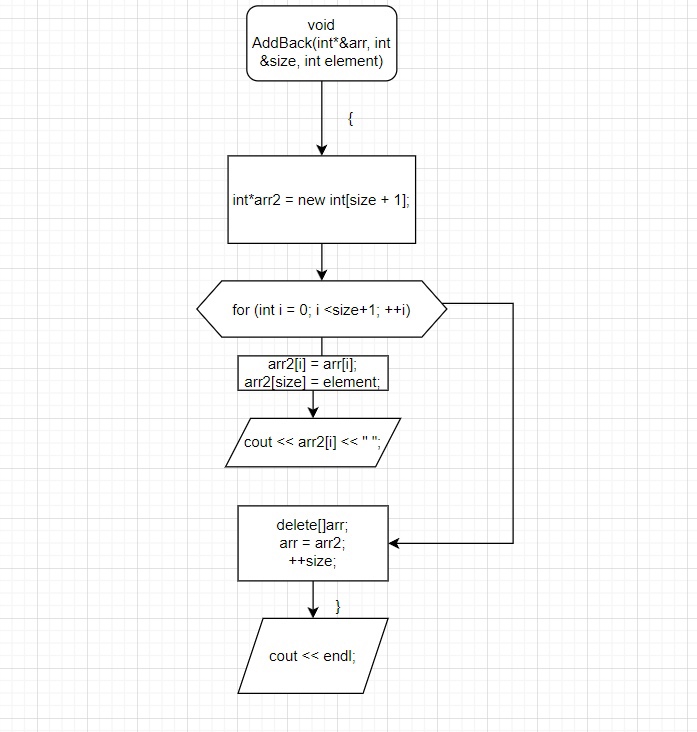


Рисунок 5

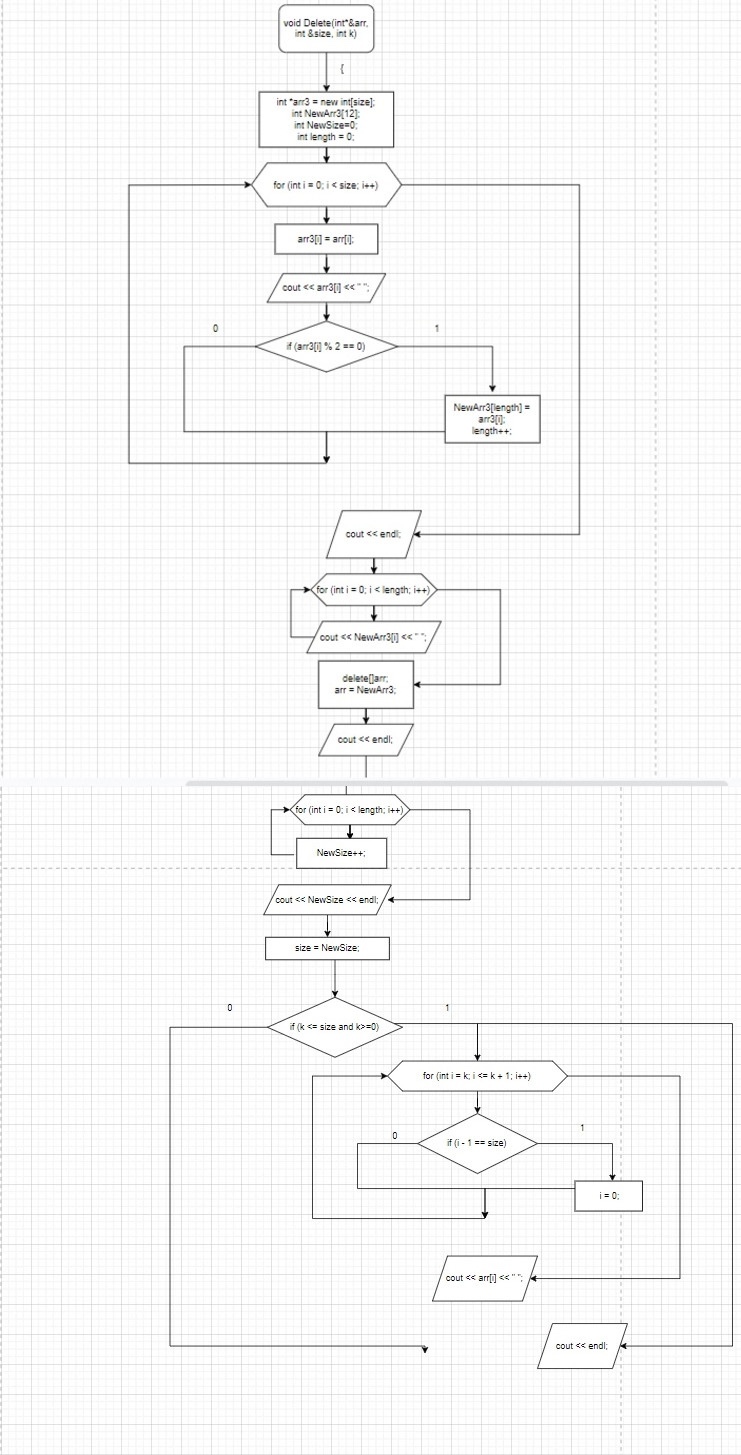


Рисунок 6

На рисунке 7 представлен результат выполнения программы для решения задачи.

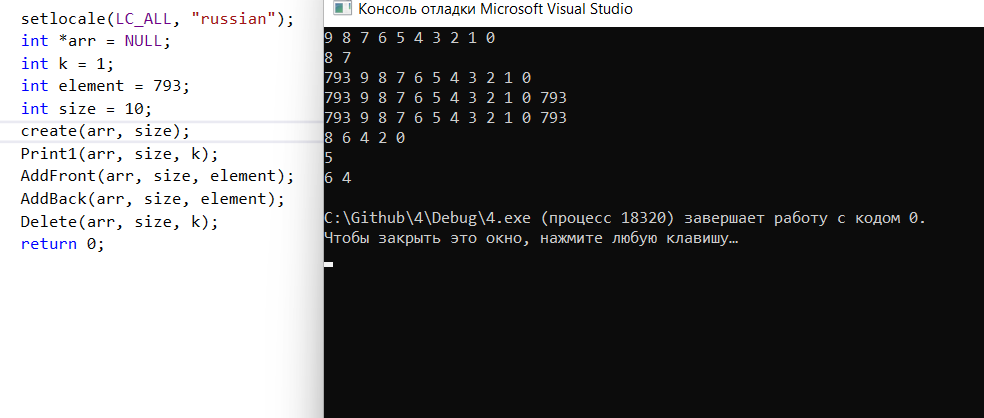


Рисунок 7- результат выполнения программы

**Вывод**

В лабораторной работе №4 поставленная задача выполнена, решена с использованием динамических одномерных массивов. Программа выполняется корректно, без ошибок.