МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ЛАБОРАТОРНАЯ РАБОТА №3  
на тему «Изменение размера массива»

Выполнила: студентка группы РИС-23-2Б Жилина Анастасия Александровна

Проверила: доцент кафедры ИТАС О.А.Полякова

2024**Разработка алгоритма проверки на упорядоченность массива.**

1. **Постановка задачи** – в одномерном массиве поставить -1 перед каждым четным элементом и удалить все нечетные элементы их полученного массива.
2. **Анализ задачи:**

1) При нахождении четного элемента раздвигаем массив под место для элемента -1: в цикле с конца массива до начала, запоминаем адрес нового места последнего элемента и адрес текущего элемента массива, затем значение последнего становится равным текущему. По завершению цикла вставляем -1 в освободившийся элемент, увеличиваем размер массива.

2) При нахождении нечетного элемента вставляем на это место следующий элемент и уменьшаем размер массива.

**Листинг программы**

#include <iostream>

const int n = 10;

using namespace std;

int main()

{

setlocale(LC\_ALL, "Rus");

int A[n\*2];

int\* p = &A[0];

int\* posl = &A[0];

int\* pred = &A[0];

int sizeA = n;

srand(time(0));

for (int i = 0; i < n; i++)//рандомно формируем массив

{

A[i] = rand() % 100;

cout << A[i] << " ";

}

for (int i = 0; i < sizeA; i++)//ставим перед всеми четными числами "-1"

{

if (\*p % 2 == 0)//если элемент массива четный, то ...

{

for (int j = sizeA; j > i; j--)//для элементов, начиная с последнего до текущего, выполняется:

{

posl = &A[0] + j;//запоминаем адрес нового последнего элемента

pred = posl - 1;//запоминаем адрес нового ПРЕД последнего элемента

\*posl = \*pred;//"двигаем" значение последнего элемента на его новое место, тем самым освобождаем место для "-1"

}

\*(posl - 1) = -1;//вставляем "-1"

sizeA++;//увеличиваем количество элементов в массиве

i = i + 1;//так как кол-во элементов увеличилось, индекс след.проверяемого элемента изменил свое значение. Чтоб повторно не проверять число, увеличим индекс на 1

p = p + 2;//по той же причине передвинем проверяемую ячейку на 2

}

else p = p + 1;//если число нечетное, просто передвигаем проверяемую ячейку на 1

}

cout << endl;

for (int i = 0; i < sizeA; i++)//выводим получившийся массив

{

cout << A[i] << " ";

}

cout << endl;

for (int i = 0; i < sizeA; i++)//удаляем все четные элементы

{

if (A[i] % 2 != 0)//если элемент массива нечетный, то ...

{

for (int j = i; j < sizeA; j++)//цикл по удалению нечетных элементов

{

A[j] = A[j + 1];//на место нечетного встает след.элемент

}

--sizeA;//уменьшаем размер массива

--i;//так как кол-во элементов в массиве уменьшилось, чтоб не перепрыгнуть элемент, уменьшим счетчик (который используется в кач-ве индекс)

}

}

for (int i = 0; i < sizeA; i++)//выводим получившийся массив

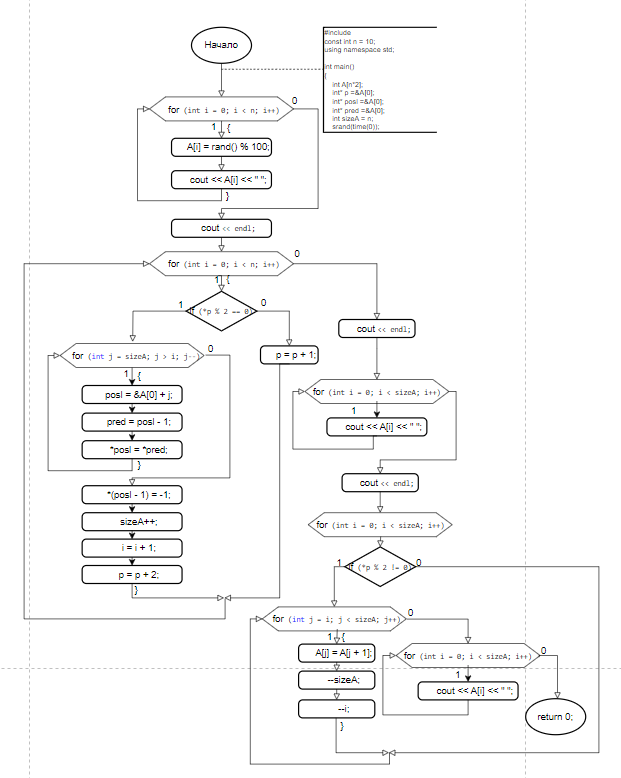
{

cout << A[i] << " ";

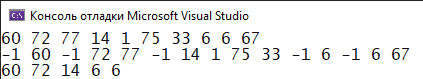
}

}

1. **Блок-схема**

****

1. **Результаты работы**



Разместим готовые исполнимые файлы в созданный репозиторий на GitHub

