МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(ПНИПУ)

Факультет: Электротехнический

Кафедра: «Информационные технологии и автоматизированные системы» (ИТАС)

Направление: Разработка информационных систем (РИС)

**О Т Ч Ё Т**   
**о лабораторной работе №4**

Дисциплина: «Основы алгоритмизации и программирования»

Тема: «Работа с одномерными массивами»

Выполнил:

Студент группы РИС-24-3б

Суяргулов А.О.

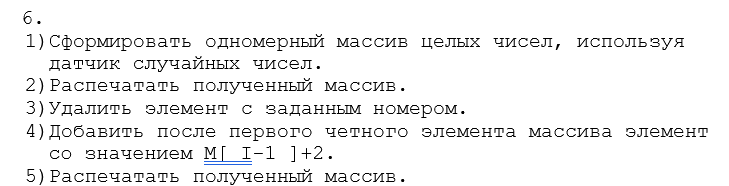
Проверил:

Доцент кафедры ИТАС

Полякова О.А.

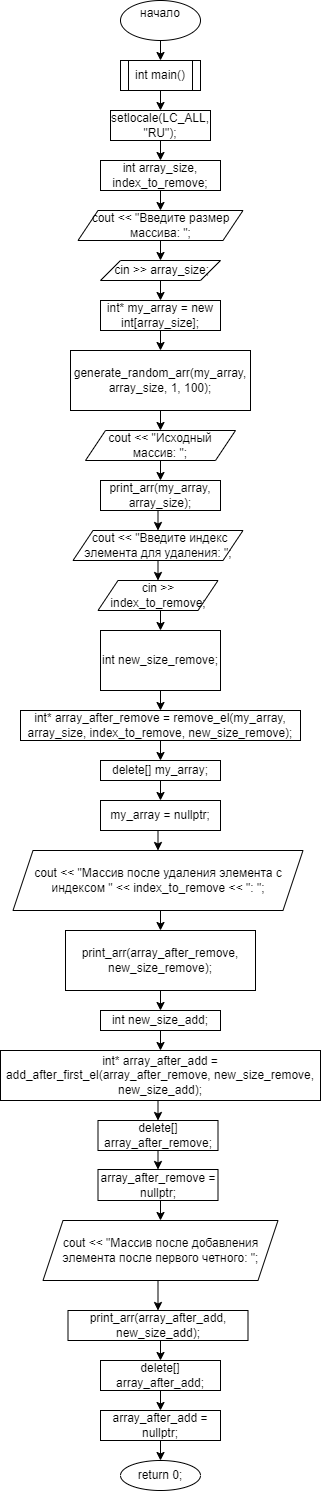
Пермь, 2025

ВВЕДЕНИЕ

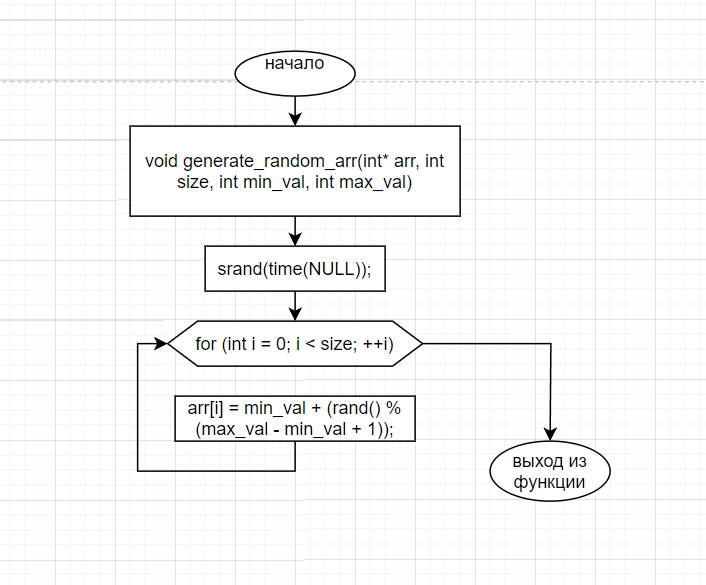


ОСНОВНАЯ ЧАСТЬ

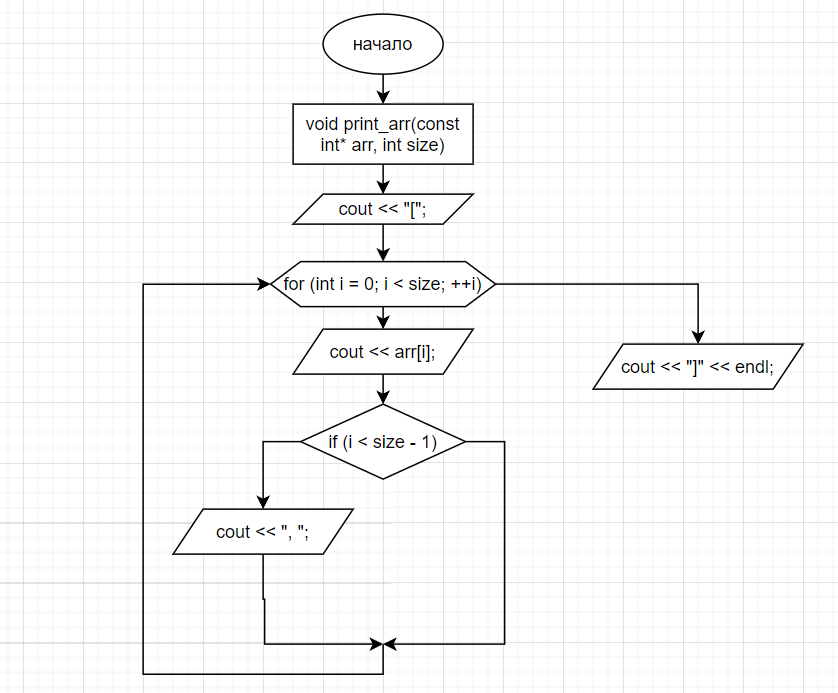
**Блок-схема:**



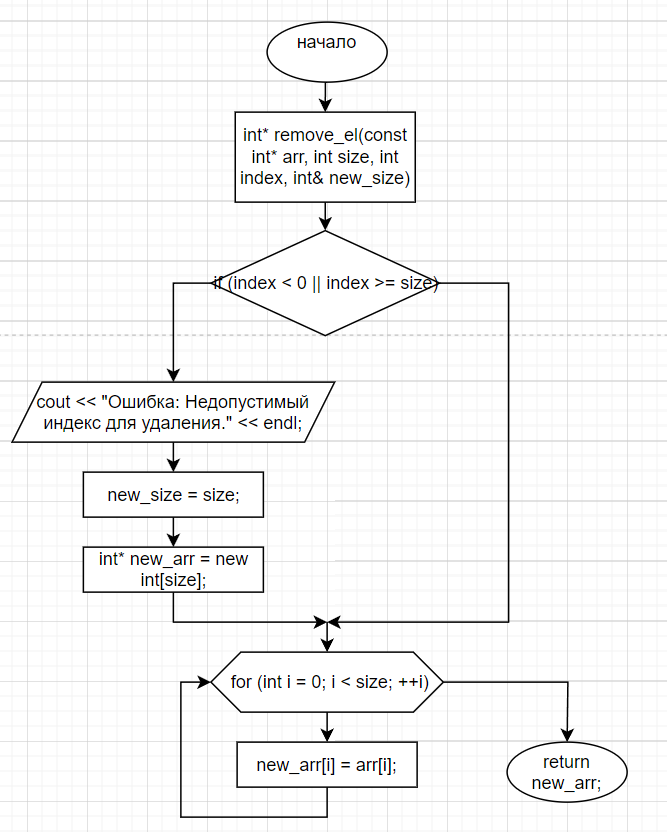
**Функция для формирования случайных чисел:**



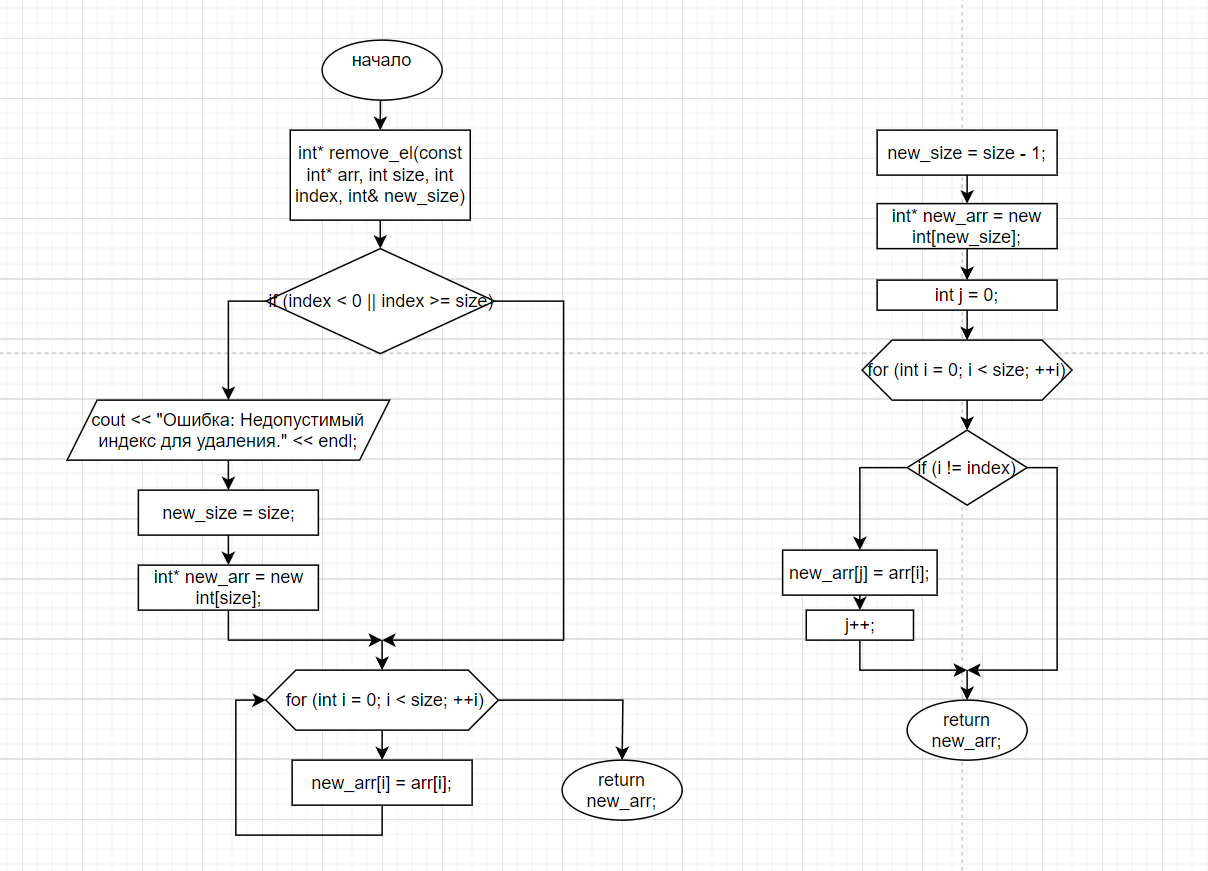
**Функция для печати массива:**

****

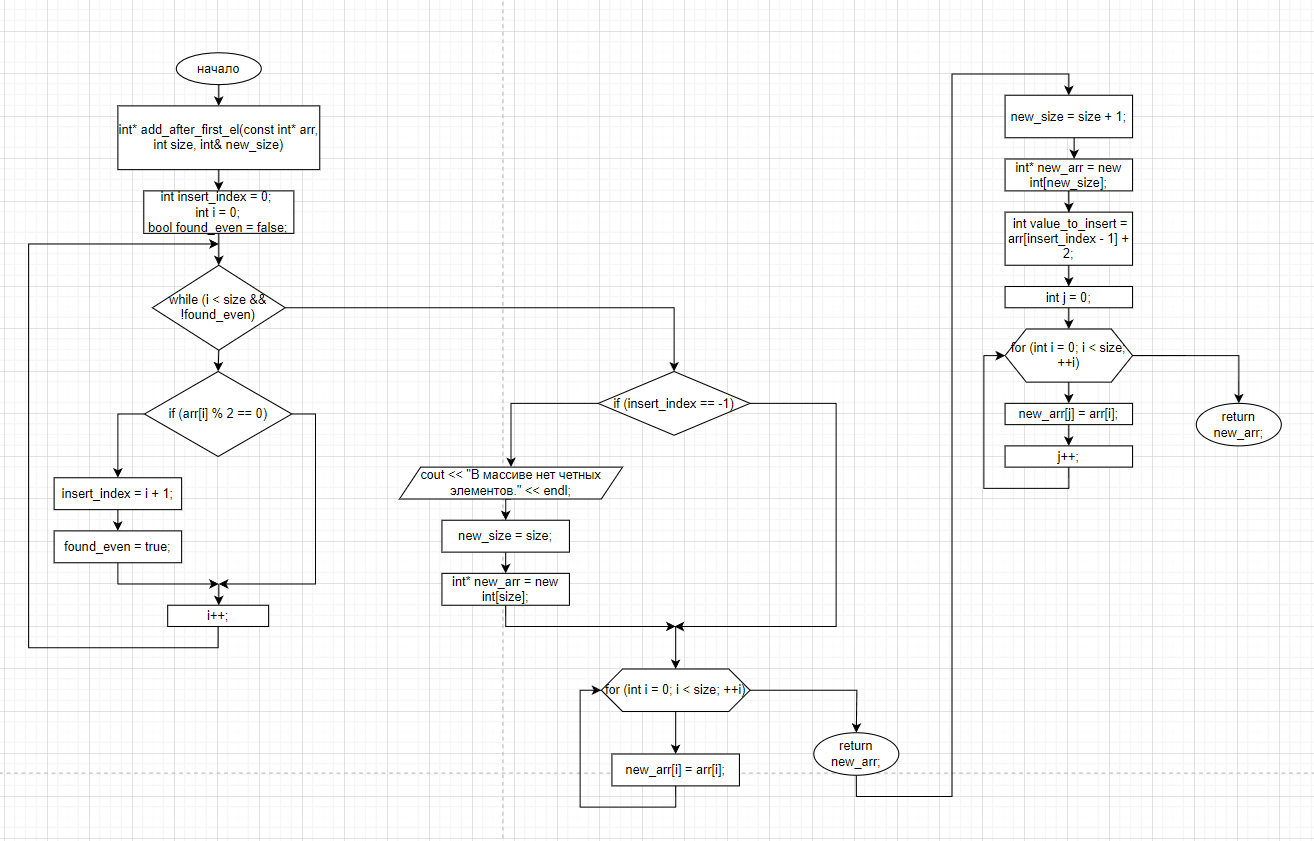
**Функция для удаления по индексу:**

****

**Функция для удаления по индексу:**

****

**Функция для добавления элемента после первого четного:**

****

**Код:**

#include <iostream>

#include <ctime>

#include <cstdlib>

using namespace std;

// Функция для формирования массива случайных чисел

void generate\_random\_arr(int\* arr, int size, int min\_val, int max\_val) {

srand(time(NULL));

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

// Генерируем случайное число

arr[i] = min\_val + (rand() % (max\_val - min\_val + 1));

}

}

// Функция для печати массива

void print\_arr(const int\* arr, int size) {

cout << "[";

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

cout << arr[i];

if (i < size - 1) {

cout << ", ";

}

}

cout << "]" << endl;

}

// Функция для удаления элемента по индексу

int\* remove\_el(const int\* arr, int size, int index, int& new\_size) {

if (index < 0 || index >= size) {

cout << "Ошибка: Недопустимый индекс для удаления." << endl;

new\_size = size;

int\* new\_arr = new int[size];

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

new\_arr[i] = arr[i];

}

return new\_arr;

}

new\_size = size - 1;

int\* new\_arr = new int[new\_size];

int j = 0;

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

if (i != index) {

new\_arr[j] = arr[i];

j++;

}

}

return new\_arr;

}

// Функция для добавления элемента после первого четного

int\* add\_after\_first\_el(const int\* arr, int size, int& new\_size) {

int insert\_index = 0;

int i = 0;

bool found\_even = false; // Флаг

while (i < size && !found\_even) {

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

insert\_index = i + 1; // Вставляем после этого индекса

found\_even = true; // Устанавливаем флаг, чтобы прекратить поиск

}

i++;

}

if (insert\_index == -1) {

cout << "В массиве нет четных элементов." << endl;

new\_size = size;

int\* new\_arr = new int[size];

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

new\_arr[i] = arr[i];

}

return new\_arr;

}

new\_size = size + 1;

int\* new\_arr = new int[new\_size];

int value\_to\_insert = arr[insert\_index - 1] + 2;

int j = 0;

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

new\_arr[j] = arr[i];

j++;

if (i == insert\_index - 1) {

new\_arr[j] = value\_to\_insert;

j++;

}

}

return new\_arr;

}

int main() {

setlocale(LC\_ALL, "RU");

int array\_size, index\_to\_remove;

cout << "Введите размер массива: ";

cin >> array\_size;

int\* my\_array = new int[array\_size];

// Генерируем массив случайных чисел

generate\_random\_arr(my\_array, array\_size, 1, 100);

cout << "Исходный массив: ";

print\_arr(my\_array, array\_size);

// Получаем индекс для удаления от пользователя

cout << "Введите индекс элемента для удаления: ";

cin >> index\_to\_remove;

// Удаляем элемент по индексу

int new\_size\_remove;

int\* array\_after\_remove = remove\_el(my\_array, array\_size, index\_to\_remove, new\_size\_remove);

delete[] my\_array; // освобождаем динамическую память

my\_array = nullptr;

// Печатаем массив после удаления

cout << "Массив после удаления элемента с индексом " << index\_to\_remove << ": ";

print\_arr(array\_after\_remove, new\_size\_remove);

// Добавляем элемент после первого четного

int new\_size\_add;

int\* array\_after\_add = add\_after\_first\_el(array\_after\_remove, new\_size\_remove, new\_size\_add);

delete[] array\_after\_remove; // освобождаем динамическую память

array\_after\_remove = nullptr;

cout << "Массив после добавления элемента после первого четного: ";

print\_arr(array\_after\_add, new\_size\_add);

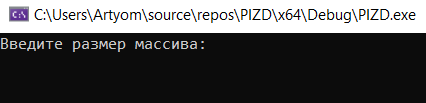
delete[] array\_after\_add; // освобождаем динамическую память

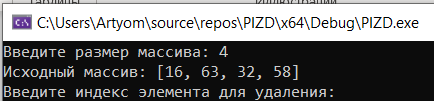
array\_after\_add = nullptr;

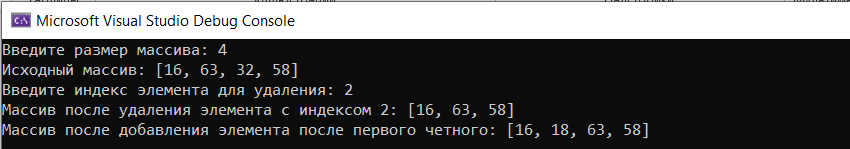
return 0;

}

**Результат работы кода:**

****

****

****

**Ссылка на GitHub:**

<https://github.com/Drazda1488/LAB_4_ARR>

