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»

**О Т Ч Е Т**

**по лабораторной работе №4**

**по дисциплине**

**«Основы алгоритмизации и программирования» семестр 2**

Выполнил студент гр. ИВТ-21-1б

Ипатов Дмитрий Сергеевич

Проверил:

Ст. Преподаватель кафедры ИТАС

Яруллин Д.В.

(оценка) (подпись)

г. Пермь-2021

**Постановка задачи:**

Получение навыков обработки одномерных массивов.

1) Сформировать одномерный массив целых чисел, используя датчик случайных чисел.

2) Распечатать полученный массив.

3) Удалить из массива все элементы, совпадающие с его минимальным значением.

4) Добавить в начало массива 3 элемента с значением равным среднему арифметическому массива.

5) Распечатать полученный массив.

Анализ задачи:

1. Создаем динамический массив, заполняем его 10ю случайными числами;
2. Находим минимальный элемент и все элементы ему равные (Используя счётчик минимальных элементов и новый массив с размером, равным предыдущему минус количество минимальных элементов);
3. Заполнив массив, расширяем его на три элемента;
4. Ищем среднее арифметическое три раза и ставим в начало массива;
5. Выводим получившийся массив.

**Описание переменных:**

int\* mas – массив, который хранит 10 случайных чисел;

int\* mas1 – массив, в который мы перепишем все элементы не равные минимальному;

int min – переменная, в которую запишем найденный минмальный элемент;

int counter – счётчик минимальных элементов;

int const size – размер первого массива;

int new\_size – размер нового массива;

int sum – сумма всех элементов в массиве;

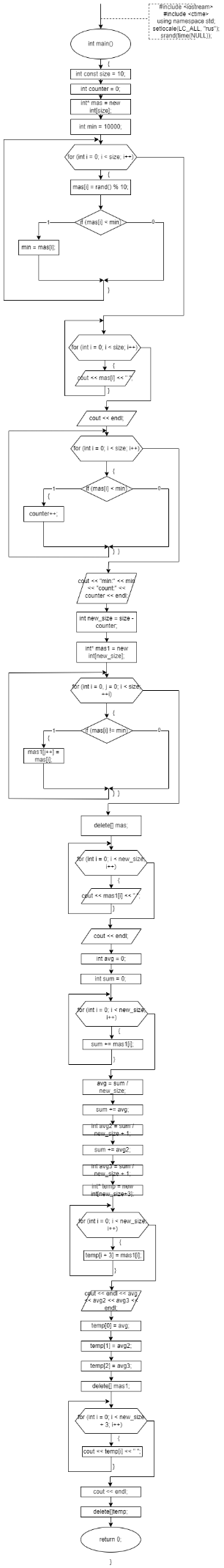
int avg – среднее арифметическое массива;

int avg2 – среднее арифметическое после добавление в массив элемента avg;

int avg3 – среднее арифметическое после добавление в массив элемента avg2;

int\* temp – массив, в котором в начало мы впишем ср. арифметические значения, а потом элементы не равные минимальному.

**Блок схема:**

****

**Код программы:**

#include <iostream>

#include <ctime>

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

srand(time(NULL)); //"точка" от которой будет браться рандомное число (привязано к времени)

int const size = 10; // Количество элементов в массиве

int counter = 0; //счетчик, который будет считать количество минимальных элементов в массиве

int\* mas = new int[size];

int min = 10000;

for (int i = 0; i < size; i++) // Ввод рандомных элементов в массив

{

mas[i] = rand() % 10;

if (mas[i] < min)

{

min = mas[i];

}

}

for (int i = 0; i < size; i++) // вывод массива

{

cout << mas[i] << " ";

}

cout << endl;

for (int i = 0; i < size; i++) // Считаем количество элементов, совпадающих с минимальным

{

if (mas[i] == min)

{

counter++;

}

}

cout << "min:" << min << "count:" << counter << endl;

int new\_size = size - counter; // Задаем размер нового массива

int\* mas1 = new int[new\_size]; // Создаем новый массив без элементов, равных минимальному

for (int i = 0, j = 0; i < size; ++i) // Заполняем новый массив

{

if (mas[i] != min)

{

mas1[j++] = mas[i];

}

}

delete[] mas; // Чистим память (удаление массива)

for (int i = 0; i < new\_size; i++) // Вывод нового массива

{

cout << mas1[i] << " ";

}

cout << endl;

int avg = 0; // Начинаем искать среднее арифметическое в массиве

int sum = 0;

for (int i = 0; i < new\_size; i++) // Суммируем все элементы массива

{

sum += mas1[i];

}

avg = sum / new\_size; // Среднее арифметическое

sum += avg;

int avg2 = sum / new\_size + 1;

sum += avg2;

int avg3 = sum / new\_size + 1;

int\* temp = new int[new\_size+3];

for (int i = 0; i < new\_size; i++)

{

temp[i + 3] = mas1[i];

}

cout << endl << avg << avg2 << avg3 << endl;

temp[0] = avg;

temp[1] = avg2;

temp[2] = avg3;

delete[] mas1;

for (int i = 0; i < new\_size + 3; i++) // Вывод нового массива

{

cout << temp[i] << " ";

}

cout << endl;

delete[]temp; // Снова чистим память

return 0;

}