**Лабораторна робота №1**

**Розробка та реалізація програми пошуку та сортування**

**Мета роботи:** оволодіти навичками складання програм пошуку та сортування елементів масиву.

**Завдання**

**Завдання 1.1.** Сформувати математичний запис фрагмента програми та обчистили значення змінної х після його виконання, згідно варіанту. N – Номер варіанту (5), кількість елементів масив (size) – 15.

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

for (int j = size - 1; j > i; j--) {

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

int tmp = a[j];

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

a[j - 1] = tmp;

}

}

}

*Розв’язання*

Данный фрагмент програми реалізує сортування массиву методом bubble sort (за зростанням), при цьому алгоритм «опускає» наймешні елементи в початок массиву. Після виконання змінна х = -45.

**Завдання 1.2.** Дана послідовність дійсних чисел . Розташувати елементи послідовності, менші за середньоарифметичне значення, за зростанням (інші елементи залишаються на своїх місцях). Метод сортування – вставка.

*Розв’язання*

1. Постановка задачі

Дана послідовність дійсних чисел . Розташувати елементи послідовності, менші за середньоарифметичне значення, за зростанням.

1. Алгоритм розв’язання задачі:

Алгоритм розв’язання задачі можна представити у вигляді такої послідовності дій:

Дія 1. Ввести елементи одновимірного масиву аrr.

Дія 2. Знайти середнє арифметичне.

Дія 2. Відсортувати елементи, які менші середнього арифметичного

Дія 3. Вивести упорядкований масив на екран.

1. **Текст програми**

#include <iostream>

#include <fstream>

#include <cmath>

#include <ctime>

using namespace std;

void writeArr(int \*arr, int len);

void printArr(int \*arr, int len, ofstream &fout, char sep = ' ');

void insertionSort(int \*arr, int n, int max);

int getCer(int \*arr, int len);

int main() {

ifstream fin("input.txt");

ofstream fout("output.txt");

srand(time(0));

int len;

fin >> len;

fout << "Длина массива: " << len << endl;

int \*arr = new int[len];

writeArr(arr, len);

fout << endl << "Был сгенерирован массив: " << endl;

printArr(arr, len, fout, ' ');

int cer = getCer(arr, len);

fout << endl << "Элементы не меньше среднего арифметического (" << cer << "):" << endl;

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

if (arr[i] >= cer) fout << arr[i];

else fout << " ";

fout << ' ';

}

fout << endl;

insertionSort(arr, len, cer);

fout << endl << "Отсортированный массив: " << endl;

printArr(arr, len, fout, ' ');

fin.close();

fout.close();

delete[] arr;

return 0;

}

void insertionSort(int \*arr, int n, int max) {

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

if (arr[i] >= max) continue;

int right = i, left = NULL;

while (true) {

for (int j = right - 1; j >= 0; j--) {

if (arr[j] < max) {

left = j;

break;

}

}

if (left == NULL) break;

if (arr[right] >= arr[left]) break;

int tmp = arr[right];

arr[right] = arr[left];

arr[left] = tmp;

right = left;

left = NULL;

}

}

}

int getCer(int \*arr, int len) {

long long sum = 0;

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

sum += arr[i];

}

return sum / len;

}

void writeArr(int \*arr, int len) {

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

arr[i] = 10 + rand() % 90;

}

}

void printArr(int \*arr, int len, ofstream &fout, char sep) {

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

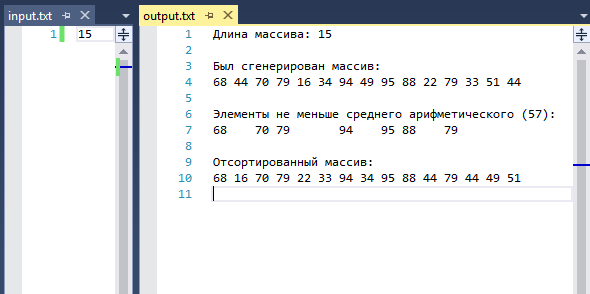
fout << arr[i] << sep;

}

fout << endl;

}

1. Результат виконання програми:



**Висновок:** на цій лабораторній роботі я оволодів навичками скаладання програм пошуку та сортування елементів массиву.