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: 09.03.04 - «Программная инженерия»

**Лабораторная работа**

**«Сортировка подсчетом. Блочная сортировка. Сортировка слиянием.»**

Выполнил студент гр. РИС-24-1б

Иванов Семен Сергеевич

Проверил:

Доц. каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

**Постановка задачи**

Отсортировать массив при помощи трех видов сортировок: подсчетом, блочной, слиянием.

**Анализ задачи**

1. Создать массив.

2. Заполнить массив.

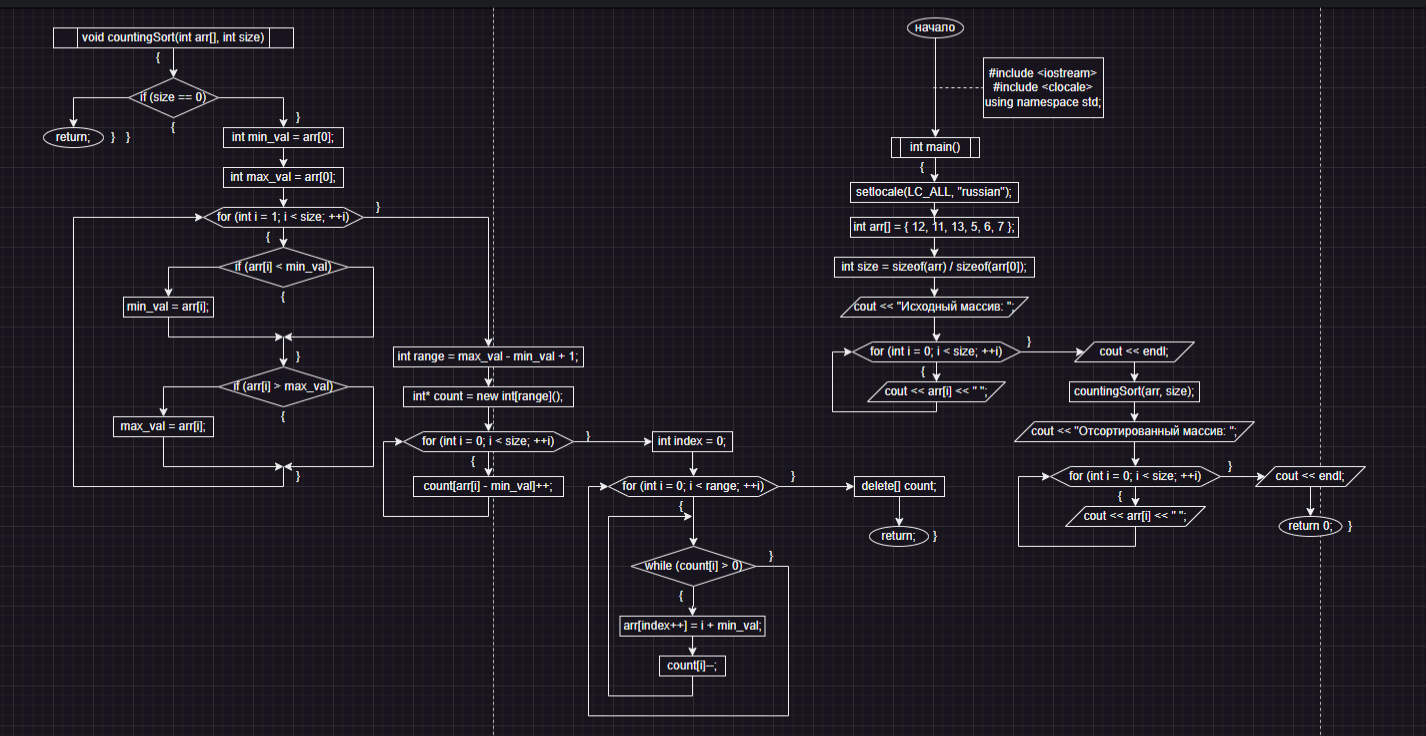
3. Написать функцию, сортирующую массив одним из способов.

4. Вывести исходный массив.

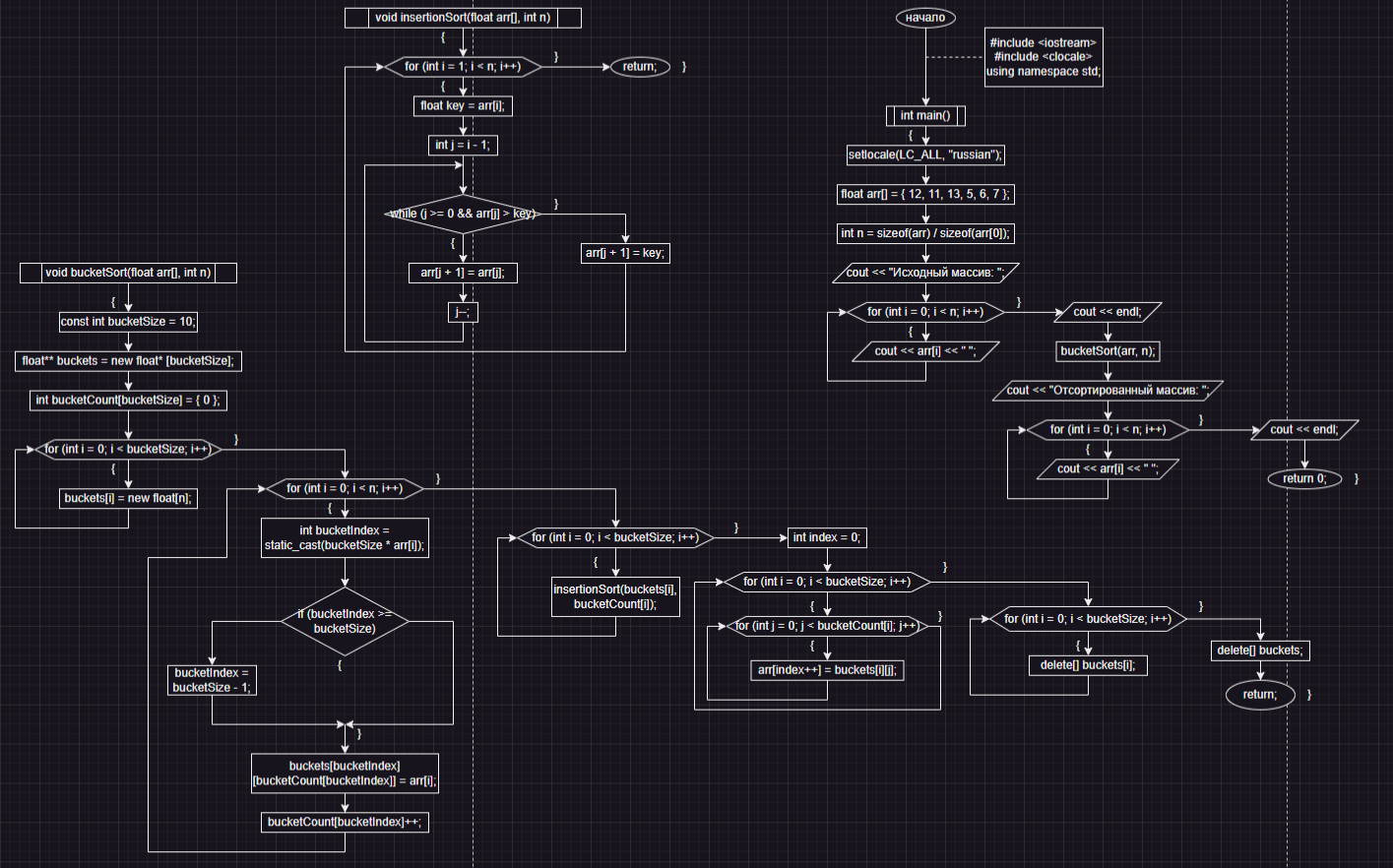
5. Отсортировать массив при помощи этой функции.

6. Вывести отсортированный массив.

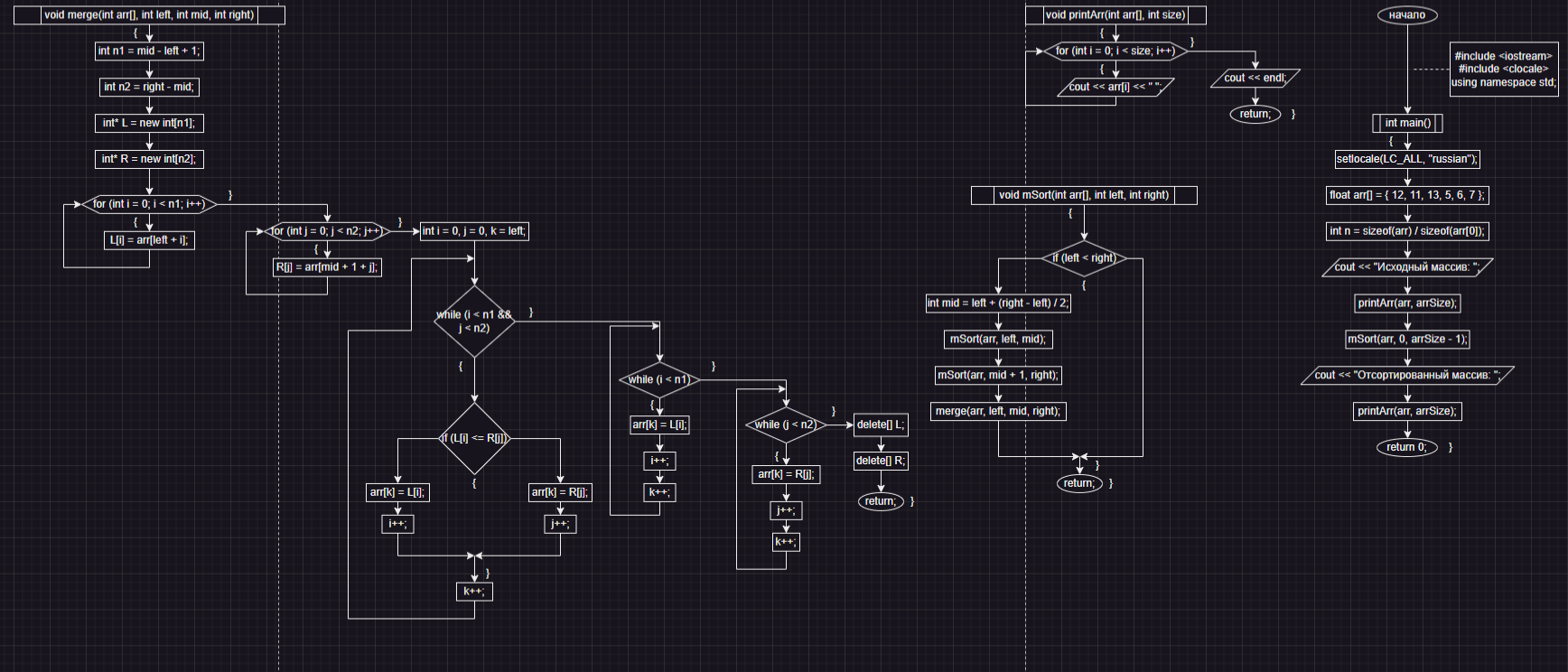
**Блок-схема(Сортировка подсчетом)**

****

**Блок-схема(Блочная сортировка)**

****

**Блок-схема(Сортировка слиянием)**

****

**Код(Сортировка подсчетом)**

#include <iostream>

#include <clocale>

using namespace std;

void countingSort(int arr[], int size) {

if (size == 0) return;

int min\_val = arr[0];

int max\_val = arr[0];

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

if (arr[i] < min\_val) {

min\_val = arr[i];

}

if (arr[i] > max\_val) {

max\_val = arr[i];

}

}

int range = max\_val - min\_val + 1;

int\* count = new int[range]();

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

count[arr[i] - min\_val]++;

}

int index = 0;

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

while (count[i] > 0) {

arr[index++] = i + min\_val;

count[i]--;

}

}

delete[] count;

}

int main() {

setlocale(LC\_ALL, “russian”);

int arr[] = { 12, 11, 13, 5, 6, 7 };

int size = sizeof(arr) / sizeof(arr[0]);

cout << "Исходный массив: ";

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

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

}

cout << endl;

countingSort(arr, size);

cout << "Отсортированный массив: ";

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

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

}

cout << endl;

return 0;

}

**Код(Блочная сортировка)**

#include <iostream>

#include <clocale>

using namespace std;

void insertionSort(float arr[], int n) {

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

float key = arr[i];

int j = i - 1;

while (j >= 0 && arr[j] > key) {

arr[j + 1] = arr[j];

j--;

}

arr[j + 1] = key;

}

}

void bucketSort(float arr[], int n) {

const int bucketSize = 10;

float\*\* buckets = new float\* [bucketSize];

int bucketCount[bucketSize] = { 0 };

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

buckets[i] = new float[n];

}

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

int bucketIndex = static\_cast<int>(bucketSize \* arr[i]);

if (bucketIndex >= bucketSize) {

bucketIndex = bucketSize - 1;

}

buckets[bucketIndex][bucketCount[bucketIndex]] = arr[i];

bucketCount[bucketIndex]++;

}

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

insertionSort(buckets[i], bucketCount[i]);

}

int index = 0;

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

for (int j = 0; j < bucketCount[i]; j++) {

arr[index++] = buckets[i][j];

}

}

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

delete[] buckets[i];

}

delete[] buckets;

}

int main() {

setlocale(LC\_ALL, "russian");

float arr[] = { 12, 11, 13, 5, 6, 7 };

int n = sizeof(arr) / sizeof(arr[0]);

cout << "Исходный массив: ";

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

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

}

cout << endl;

bucketSort(arr, n);

cout << "Отсортированный массив: ";

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

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

}

cout << endl;

return 0;

}

**Код(Сортировка слиянием)**

#include <iostream>

#include <clocale>

using namespace std;

void merge(int arr[], int left, int mid, int right) {

int n1 = mid - left + 1;

int n2 = right - mid;

int\* L = new int[n1];

int\* R = new int[n2];

for (int i = 0; i < n1; i++)

L[i] = arr[left + i];

for (int j = 0; j < n2; j++)

R[j] = arr[mid + 1 + j];

int i = 0, j = 0, k = left;

while (i < n1 && j < n2) {

if (L[i] <= R[j]) {

arr[k] = L[i];

i++;

}

else {

arr[k] = R[j];

j++;

}

k++;

}

while (i < n1) {

arr[k] = L[i];

i++;

k++;

}

while (j < n2) {

arr[k] = R[j];

j++;

k++;

}

delete[] L;

delete[] R;

}

void mSort(int arr[], int left, int right) {

if (left < right) {

int mid = left + (right - left) / 2;

mSort(arr, left, mid);

mSort(arr, mid + 1, right);

merge(arr, left, mid, right);

}

}

void printArr(int arr[], int size) {

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

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

cout << std::endl;

}

int main() {

setlocale(LC\_ALL, “russian”);

int arr[] = { 12, 11, 13, 5, 6, 7 };

int arrSize = sizeof(arr) / sizeof(arr[0]);

cout << "Исходный массив: ";

printArr(arr, arrSize);

mSort(arr, 0, arrSize - 1);

cout << "Отсортированный массив: ";

printArr(arr, arrSize);

return 0;

}

**Скриншот решения(Все)**

****

**github**

https://github.com/Lagus645/sortPr