МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Лабораторная работа:**

"Простые сортировки"

Выполнила: студентка гр.РИС-23-3Б

Мокроусова Ангелина Андреевна

Проверила: доцент кафедры ИТАС

Полякова Ольга Андреевна

Пермь 2024

**Постановка задачи:**

Реализовать в программе 4 вида простых сортировок: слиянием, быстрая, подсчетом, блочная.

**Анализ задачи:**

Блочная:

* Разбивает массив на блоки равного размера.
* Сортирует каждый блок по отдельности.
* Сливает отсортированные блоки.

Слиянием:

* Разделяет массив пополам.
* Рекурсивно сортирует обе половины.
* Сливает отсортированные половины.

Подсчётом:

* Определяет количество каждого элемента в массиве.
* Создаёт новый массив, вставляя каждый элемент в соответствии с его подсчитанной частотой.

Быстрая:

* Выбирает опорный элемент.
* Разделяет массив на два подмассива: элементы, меньшие или равные опорному, и элементы, большие опорного.
* Рекурсивно применяет быструю сортировку к подмассивам.

**Код на С++(подсчетом):**

#include <iostream>

#include <stdlib.h>

using namespace std;

int main()

{

int sizeA = 25;

int\* arr = new int[sizeA];

int mx = 0, mn = 0;

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

{

arr[i] = rand() % 10 - 5;

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

if (arr[i] < mn)

{

mn = arr[i];

}

if (arr[i] > mx)

{

mx = arr[i];

}

}

int sizeB = mx - mn + 1;

int\* barr = new int[sizeB];

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

{

barr[i] = 0;

}

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

{

barr[arr[i] - mn]++;

}

int index = 0;

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

{

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

{

arr[index++] = i + mn;

}

}

cout << endl;

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

{

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

}

}

**Код на С++(быстрая):**

#include <iostream>

using namespace std;

void QuickSort(int\* arr, int size)

{

int l = 0, r = size - 1;

int mid = arr[size / 2];

do

{

while (arr[l] < mid)

{

l++;

}

while (arr[r] > mid)

{

r--;

}

if (l <= r)

{

int tmp = arr[l];

arr[l] = arr[r];

arr[r] = tmp;

l++;

r--;

}

if (r > 0)

{

QuickSort(arr, r + 1);

}

if (l < size)

{

QuickSort(&arr[l], size - l);

}

} while (l <= r);

}

int main()

{

int arr[] = { 1, 101, 78, 12, 55, 92, 0, 2, 4, 12, -7, 16, 34, 5, 6, 4, 23, 60, 28, 52, 11, 3, 43, 6, 2 };

for (int i = 0; i < (sizeof(arr) / sizeof(int)); i++)

{

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

}

cout << endl;

QuickSort(arr, (sizeof(arr) / sizeof(int)));

for (int i = 0; i < (sizeof(arr) / sizeof(int)); i++)

{

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

}

}

**Код на С++(блочная):**

#include <iostream>

using namespace std;

void PrintArr(int\*\* buckets, int bucket\_count, int size, int mx)

{

for (int i = 0; i < bucket\_count; i++)

{

cout << endl;

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

{

cout << buckets[i][j] << " ";

}

}

}

void BucketSort(int arr[], int size)

{

int mx = arr[0];

int mn = arr[0];

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

{

if (arr[i] > mx)

{

mx = arr[i];

}

if (arr[i] < mn)

{

mn = arr[i];

}

}

int bucket\_count = ((mx - mn) / size) + 1;

int\*\* buckets = new int\* [bucket\_count];

for (int i = 0; i < bucket\_count; i++)

{

buckets[i] = new int[size];

}

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

{

int bucket\_index = ((arr[i] - mn) / size);

buckets[bucket\_index][i] = arr[i];

for (int j = 0; j < bucket\_count; j++)

{

if (j != bucket\_index)

{

buckets[j][i] = -1;

}

}

}

for (int i = 0; i < bucket\_count; i++)

{

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

{

if (buckets[i][j] != -1)

{

for (int k = j; k < size; k++)

{

if (buckets[i][j] > buckets[i][k])

{

int tmp = buckets[i][j];

buckets[i][j] = buckets[i][k];

buckets[i][k] = tmp;

}

}

}

}

}

int index = 0;

for (int i = 0; i < bucket\_count; i++)

{

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

{

if (buckets[i][j] != -1)

{

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

}

}

}

for (int i = 0; i < bucket\_count; i++)

{

delete[]buckets[i];

}

delete[]buckets;

}

int main()

{

int arr[] = { 1, 101, 78, 12, 55, 92, 0, 2, 4, 12, 7, 16, 34, 5, 6, 4, 23, 60, 28, 52, 11, 3, 43, 6, 2 };

BucketSort(arr, sizeof(arr) / sizeof(int));

cout << endl;

for (int i = 0; i < sizeof(arr) / sizeof(int); i++)

{

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

}

}

**Код на С++(слиянием):**

#include <iostream>

using namespace std;

void Merge(int\* arr, int first, int last)

{

int st, md, fl;

st = first;

md = (first + last) / 2;

fl = md + 1;

int\* barr = new int[256];

for (int i = first; i <= last; i++)

{

if ((st <= md && ((fl > last) || arr[st] < arr[fl])))

{

barr[i] = arr[st];

st++;

}

else

{

barr[i] = arr[fl];

fl++;

}

}

for (int i = first; i <= last; i++)

{

arr[i] = barr[i];

}

delete[]barr;

}

void Sort(int\* arr, int first, int last)

{

if (first < last)

{

Sort(arr, first, (first + last) / 2);

Sort(arr, (first + last) / 2 + 1, last);

Merge(arr, first, last);

}

}

int main()

{

int arr[] = { 1, 3, 78, 12, 55, 92, 0, 2, 4, 12, 7, 16, 34, 5, 6, 4, 23, 60, 28, 52, 11, 101, 43, 6, 2 };

for (int i = 0; i < (sizeof(arr) / sizeof(int)) - 1; i++)

{

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

}

Sort(arr, 0, (sizeof(arr) / sizeof(int)) - 1);

cout << endl;

for (int i = 0; i < (sizeof(arr) / sizeof(int)); i++)

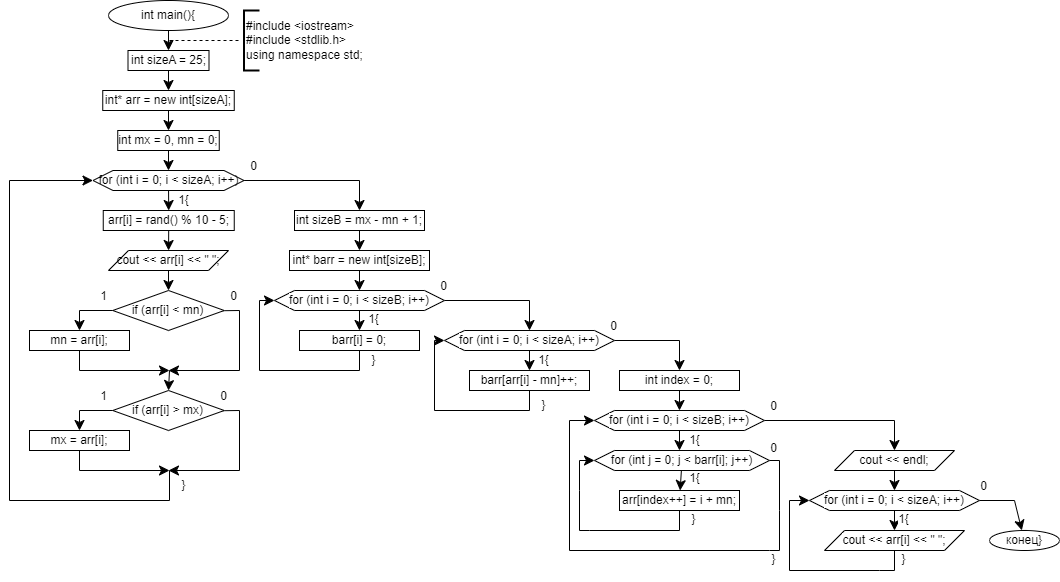
{

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

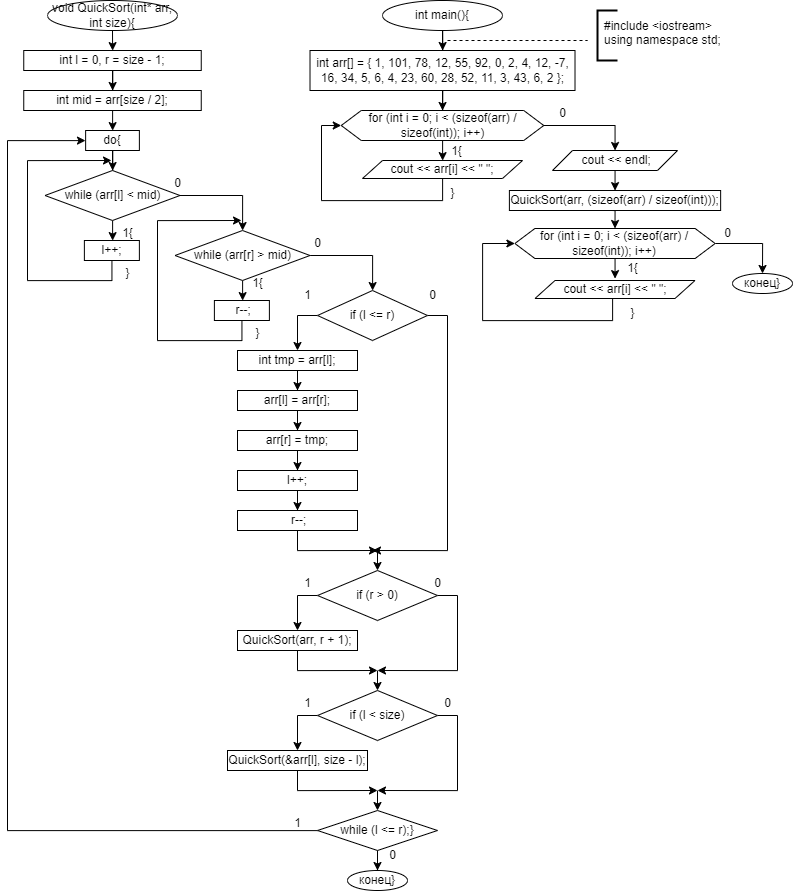
}

}

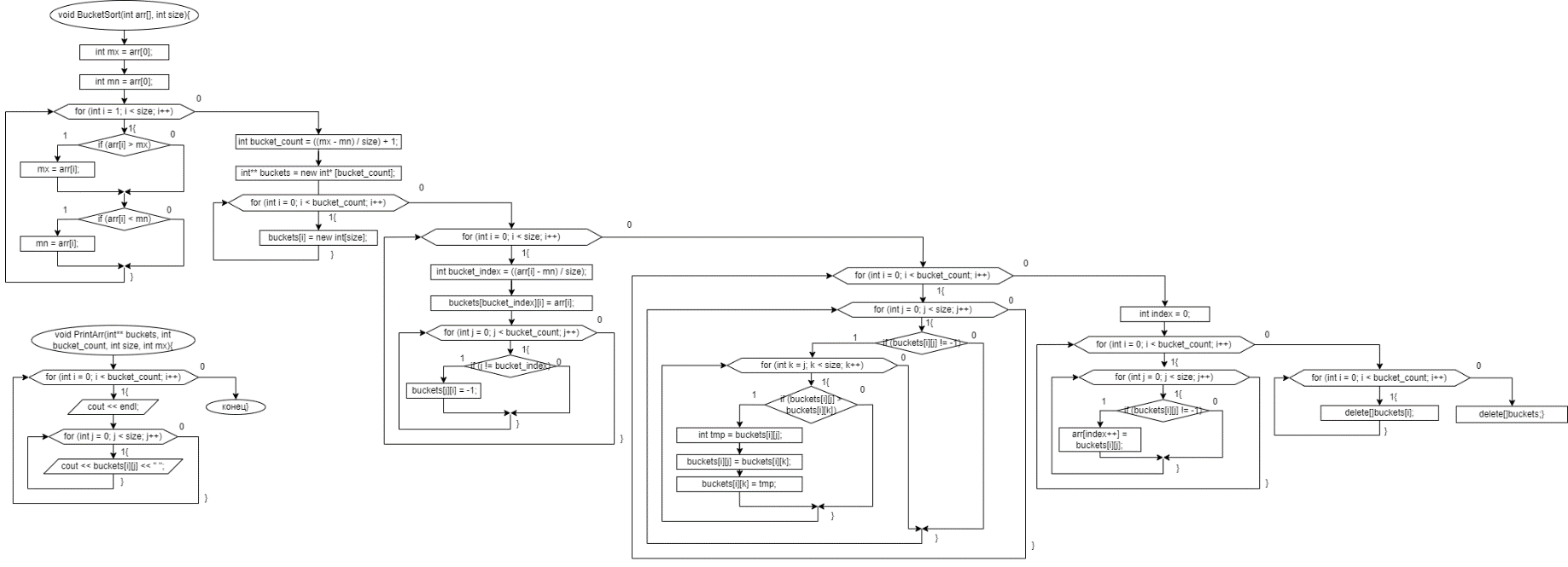
**Блок схема(Подсчетом):**



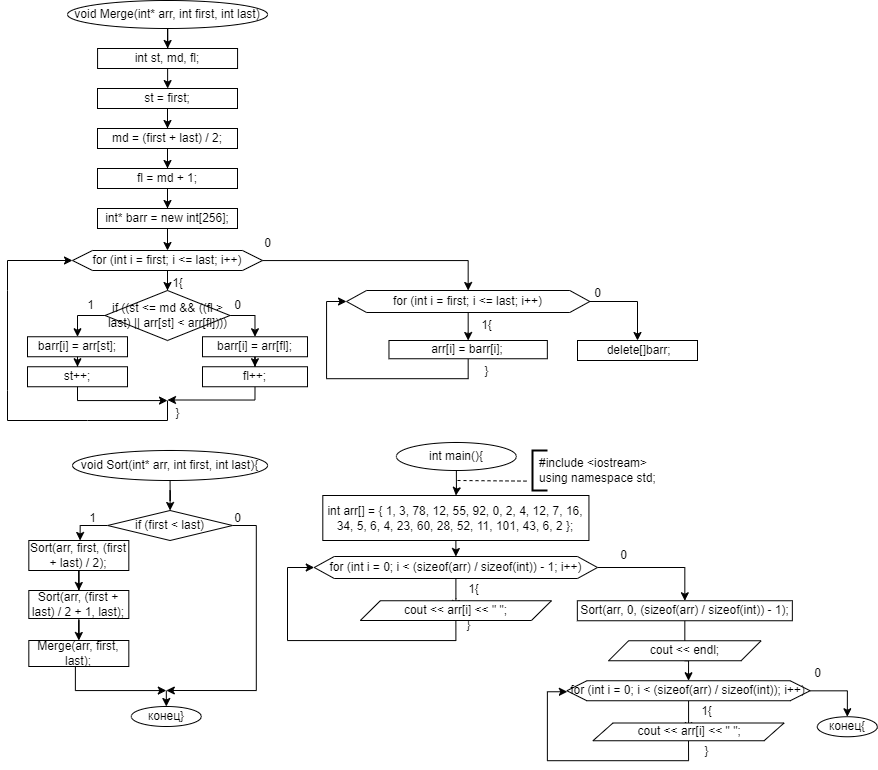
**Блок схема(Быстрая):**



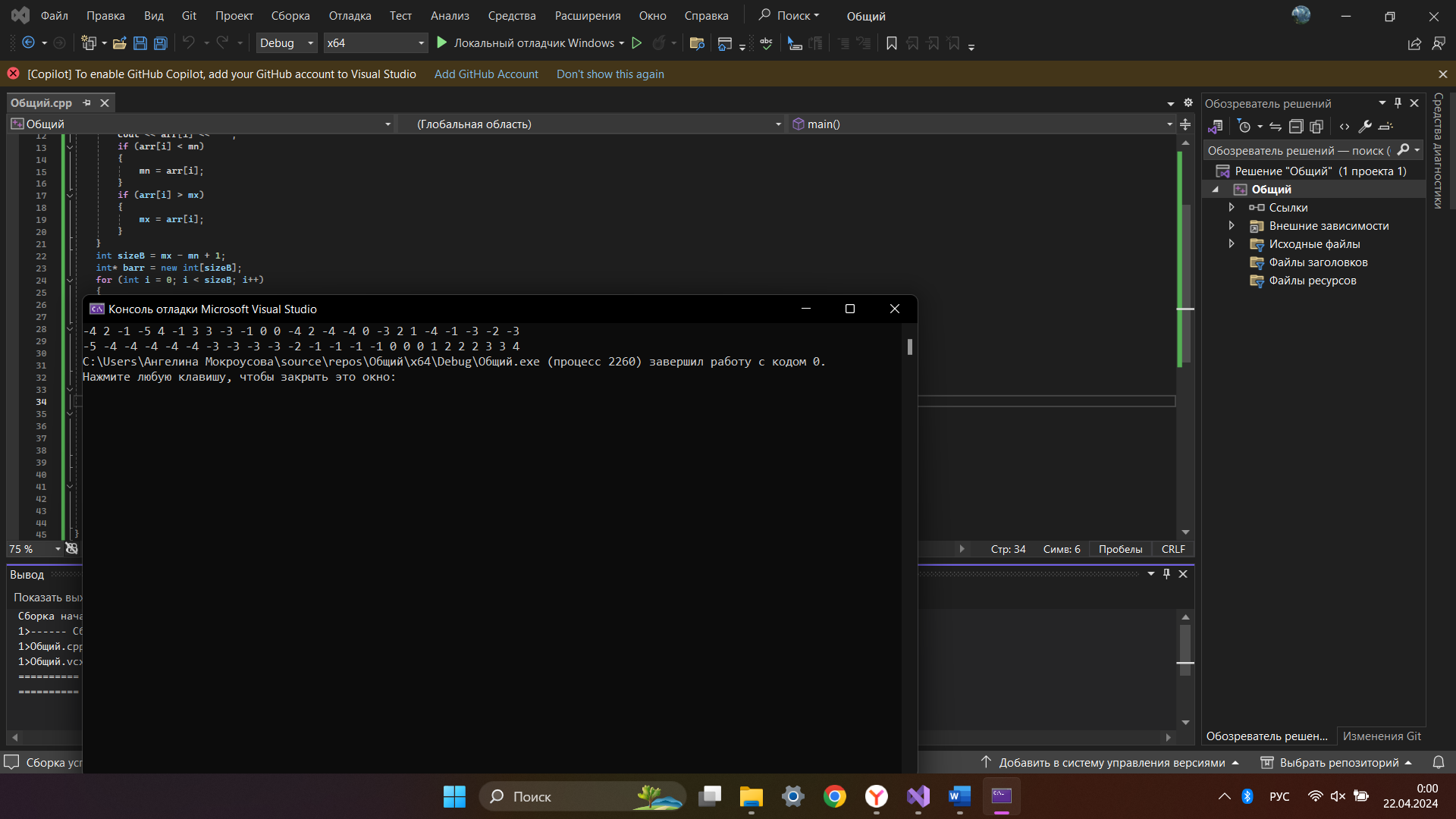
**Блок схема(Блочная):**

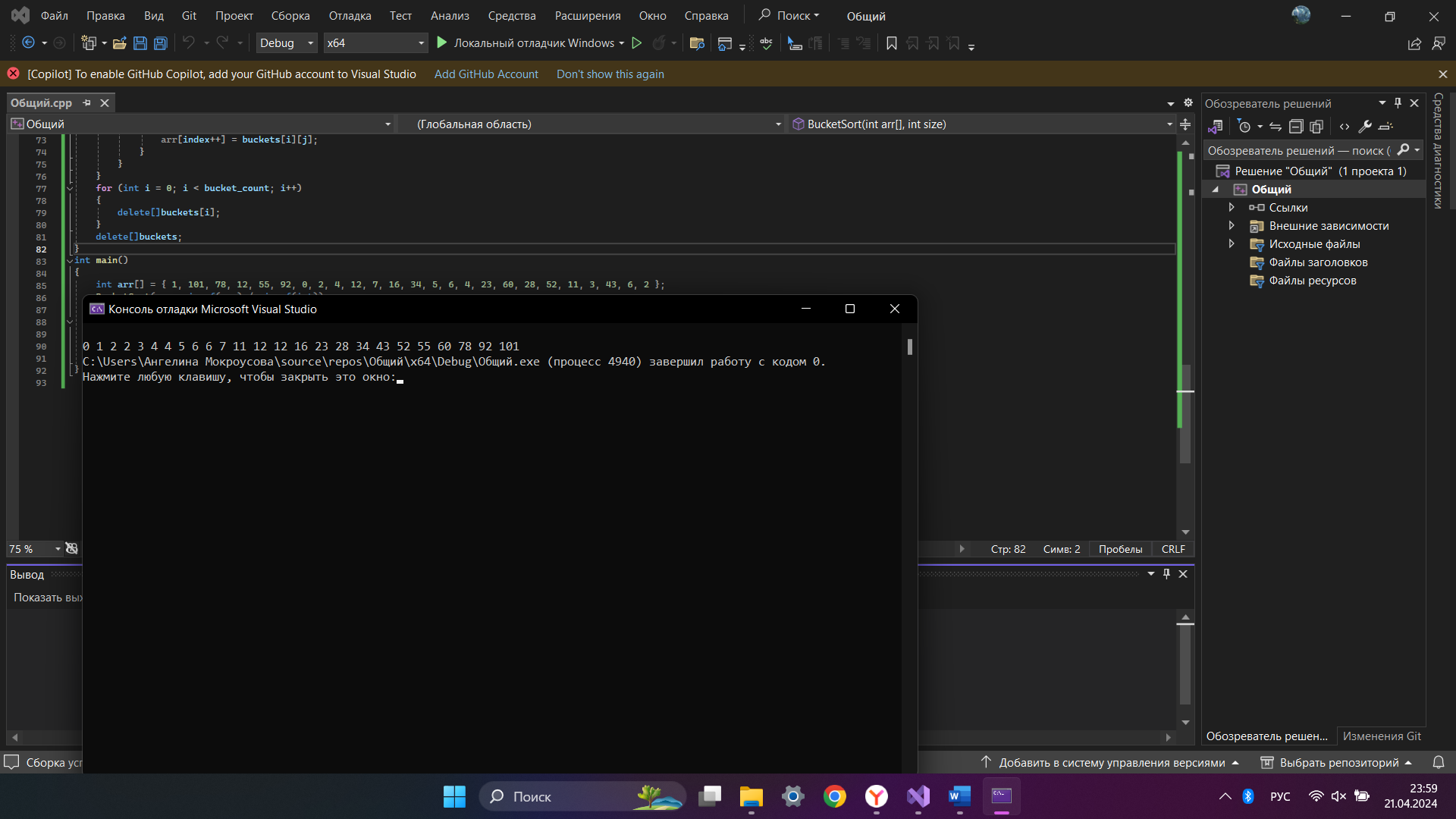


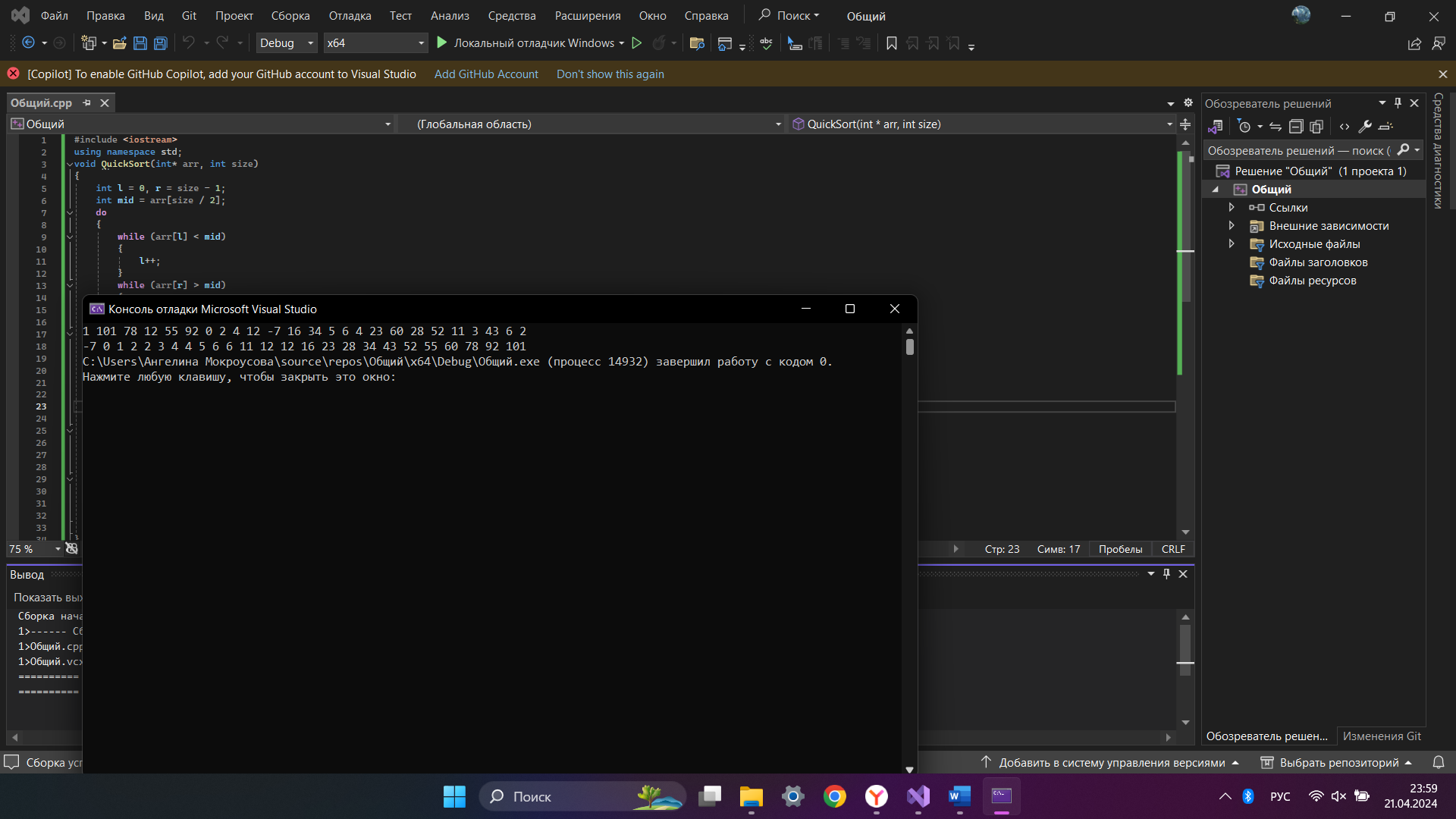
**Блок схема(Слиянием):**

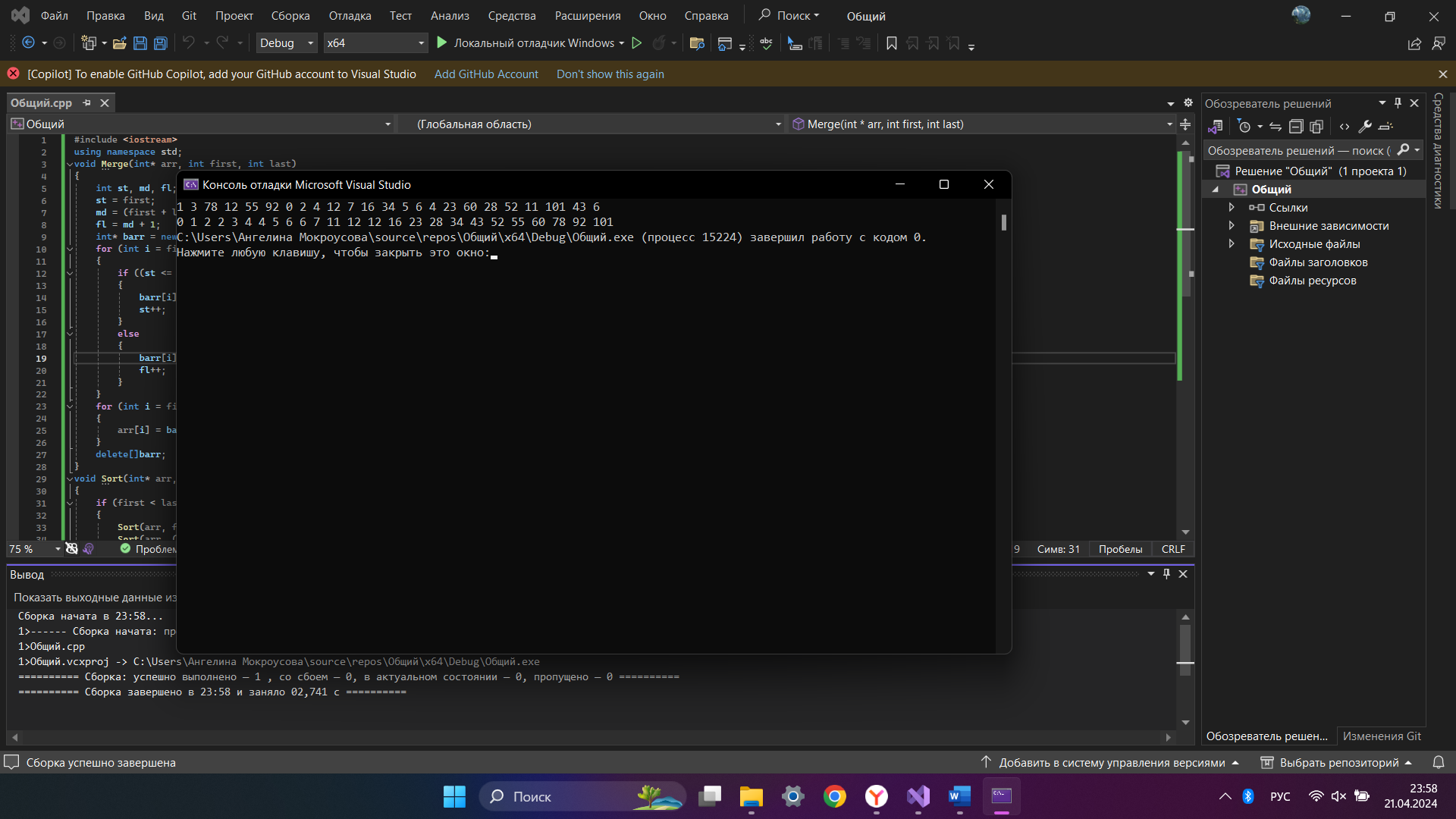


**Скриншот работы программы:**









**Вывод:**

Все программы успешно выполняют свою задачу.

**Скриншот с GitHab:**