Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования

Пермский национальный исследовательский политехнический университет

Факультет электротехнический

Кафедра ИТАС

ОТЧЁТ

о лабораторной работе №12

Выполнил:
Студент группы ИВТ-23-1Б
Пискунов Д. А.

_____

Проверил:
Доцент кафедры ИТАС
Яруллин Д.В.

_____

_____

Пермь 2024

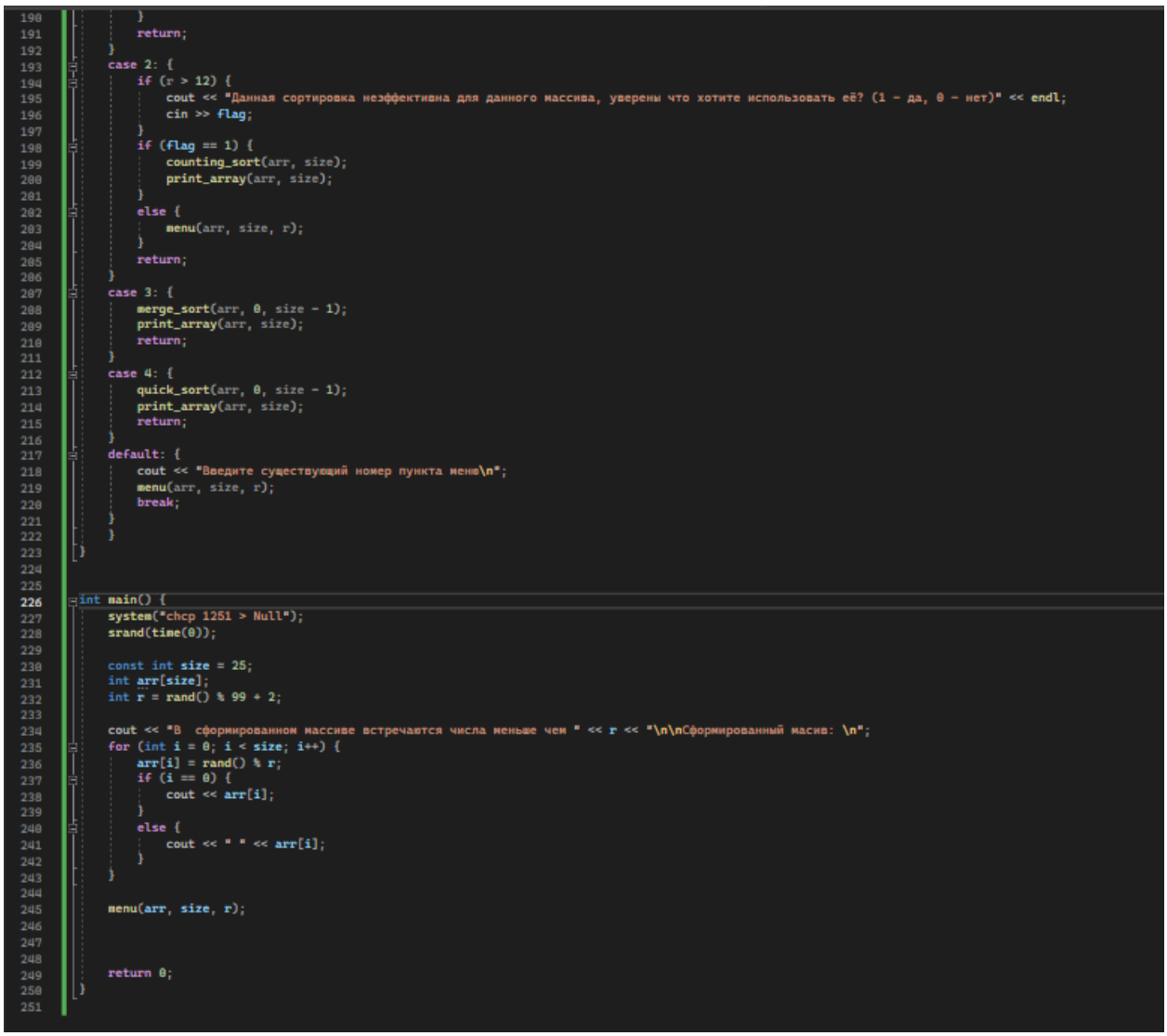
Задача:
Реализовать с помощью кода 4 сортировки:
1) блочная сортировка
2) сортировка подсчётом
3) сортировка слиянием
4) быстрая сортировка по Ломуто

Текст программы

```cpp
#include <iostream>
#include <string>
#include <time.h>

using namespace std;

void print_array(int* arr, int size) {
    for (int i = 0; i < size; i++) {
        if (i == 0) {
            cout << arr[i];
        }
        else {
            cout << " " << arr[i];
        }
    }
}

void bucket_sort(int* arr, int size) {
    const int bucket_num = 10;
    int max = arr[0];
    for (int i = 0; i < size; i++) {
        if (arr[i] > max) {
            max = arr[i];
        }
    }
    max++;

    int** buckets = new int* [bucket_num];
    for (int i = 0; i < bucket_num; i++) {
        buckets[i] = new int[size];
    }

    int bucket_size[bucket_num] = { 0 };
    for (int i = 0; i < size; i++) {
        int bucket_index = arr[i] * bucket_num / max;
        buckets[bucket_index][bucket_size[bucket_index]++] = arr[i];
    }

    for (int i = 0; i < bucket_num; i++) {
        for (int j = 0; j < bucket_size[i]; j++) {
            int tmp = buckets[i][j];
            int k = j - 1;
            while (k >= 0 && buckets[i][k] > tmp)
            {
                buckets[i][k + 1] = buckets[i][k];
                k--;
            }
            buckets[i][k + 1] = tmp;
        }
    }
    int index = 0;
    for (int i = 0; i < bucket_num; i++) {
        for (int j = 0; j < bucket_size[i]; j++) {
            arr[index++] = buckets[i][j];
        }
    }
    delete[] buckets;
}

void counting_sort(int* arr, int size) {
    int max = arr[0];
    for (int i = 1; i < size; i++) {
        if (max < arr[i]) {
            max = arr[i];
```

```
            max = arr[i];
        }
    }
    max++;

    int* count = new int[max];
    int* output = new int[size];

    for (int i = 0; i < max; i++) {
        count[i] = 0;
    }
    for (int i = 0; i < size; i++) {
        count[arr[i]]++;
    }
    for (int i = 1; i < max; ++i) {
        count[i] += count[i - 1];
    }
    for (int i = size - 1; i >= 0; i--) {
        output[count[arr[i]] - 1] = arr[i];
        count[arr[i]]--;
    }
    for (int i = 0; i < size; i++) {
        arr[i] = output[i];
    }
    delete[] count;
    delete[] output;

}

void merge(int* arr, int left, int mid, int right) {
    int left_size = mid - left + 1;
    int right_size = right - mid;
    int* left_arr = new int[left_size];
    int* right_arr = new int[right_size];

    for (int i = 0; i < left_size; i++) {
        left_arr[i] = arr[left + i];

    }
    for (int i = 0; i < right_size; i++) {
        right_arr[i] = arr[mid + 1 + i];

    }

    int left_index = 0, right_index = 0, merged_index = left;
    while (left_index < left_size && right_index < right_size) {
        if (left_arr[left_index] <= right_arr[right_index]) {
            arr[merged_index] = left_arr[left_index];
            left_index += 1;
        }
        else {
            arr[merged_index] = right_arr[right_index];
            right_index += 1;
        }
        merged_index += 1;
    }

    while (left_index < left_size) {
        arr[merged_index] = left_arr[left_index];
        merged_index += 1;
        left_index += 1;
    }

    while (right_index < right_size) {
        arr[merged_index] = right_arr[right_index];
```

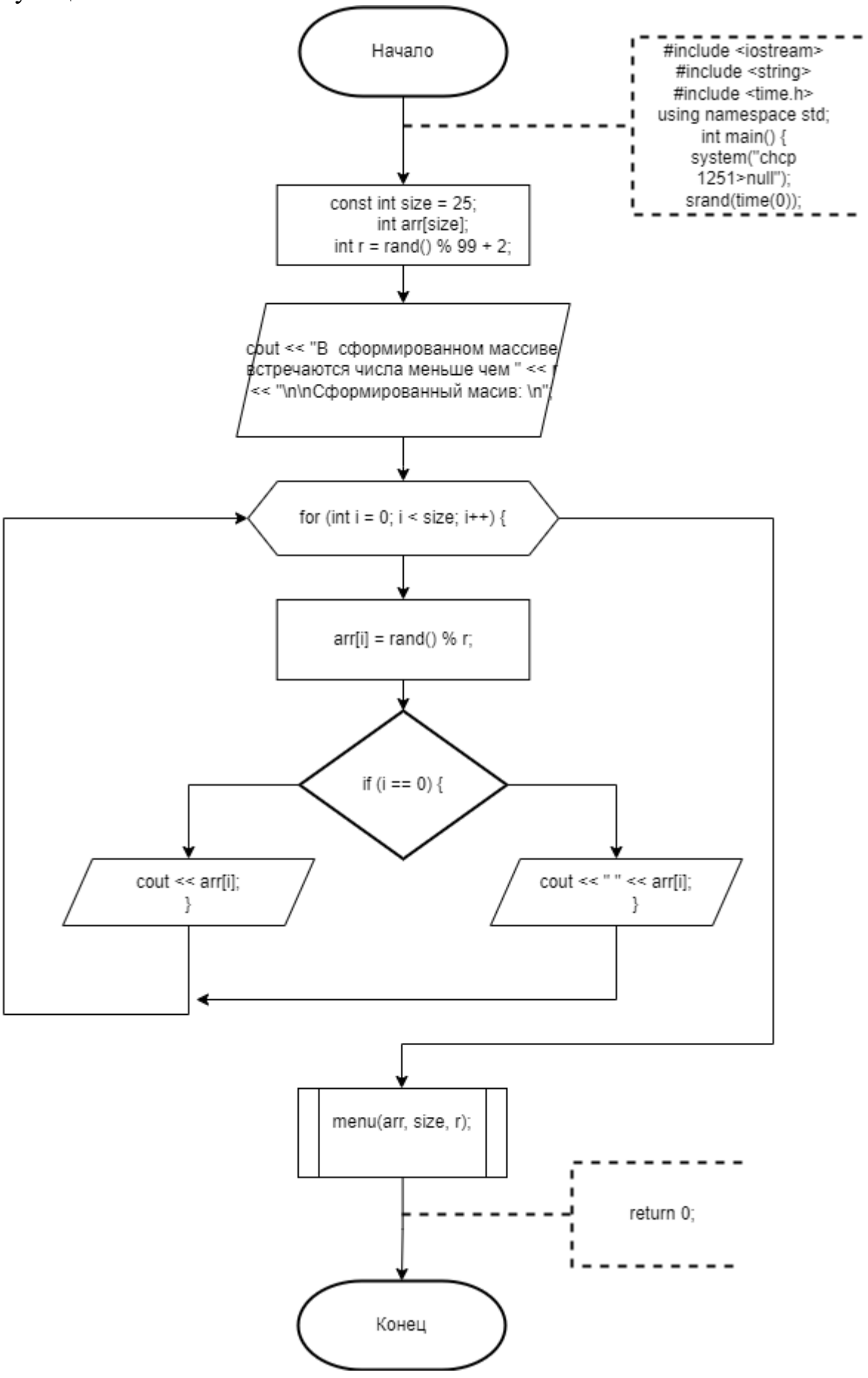
```
    while (right_index < right_size) {
        arr[merged_index] = right_arr[right_index];
        merged_index += 1;
        right_index += 1;
    }

    delete[] left_arr;
    delete[] right_arr;
}
void merge_sort(int* arr, int left, int right) {
    if (left < right) {
        int mid = left + (right - left) / 2;
        merge_sort(arr, left, mid);
        merge_sort(arr, mid + 1, right);
        merge(arr, left, mid, right);
    }
}

int partition(int* arr, int low, int high, int pivot) {
    int pind = low;
    for (int i = low; i <= high; i++) {
        if (arr[i] <= pivot) {
            swap(arr[pind], arr[i]);
            pind++;
        }
    }
    pind--;
    return pind;
}
void quick_sort(int* arr, int low, int high) {
    if (low < high) {
        int pivot = arr[high];
        int pi = partition(arr, low, high, pivot);
        quick_sort(arr, low, pi - 1);
        quick_sort(arr, pi + 1, high);
    }
}


void menu(int* arr, int size, int r) {
    int method_of_sort;
    int flag = 1;

    cout << "\n\nВыберите метод сортировки: \n";
    cout << "1. Блочная сортировка\n";
    cout << "2. Сортировка подсчетом\n";
    cout << "3. Сортировка слиянием\n";
    cout << "4. Быстрая по Ломуто\n";
    cin >> method_of_sort;
    cout << "=================================================================\n\n";

    switch (method_of_sort) {
    case 1: {
        if (r < 30) {
            cout << "Данная сортировка неэффективна для данного массива, уверены что хотите использовать её? (1 - да, 0 - нет)" << endl;
            cin >> flag;
        }
        if (flag == 1) {
            bucket_sort(arr, size);
            print_array(arr, size);
        }
        else {
            menu(arr, size, r);
        }
        return;
```



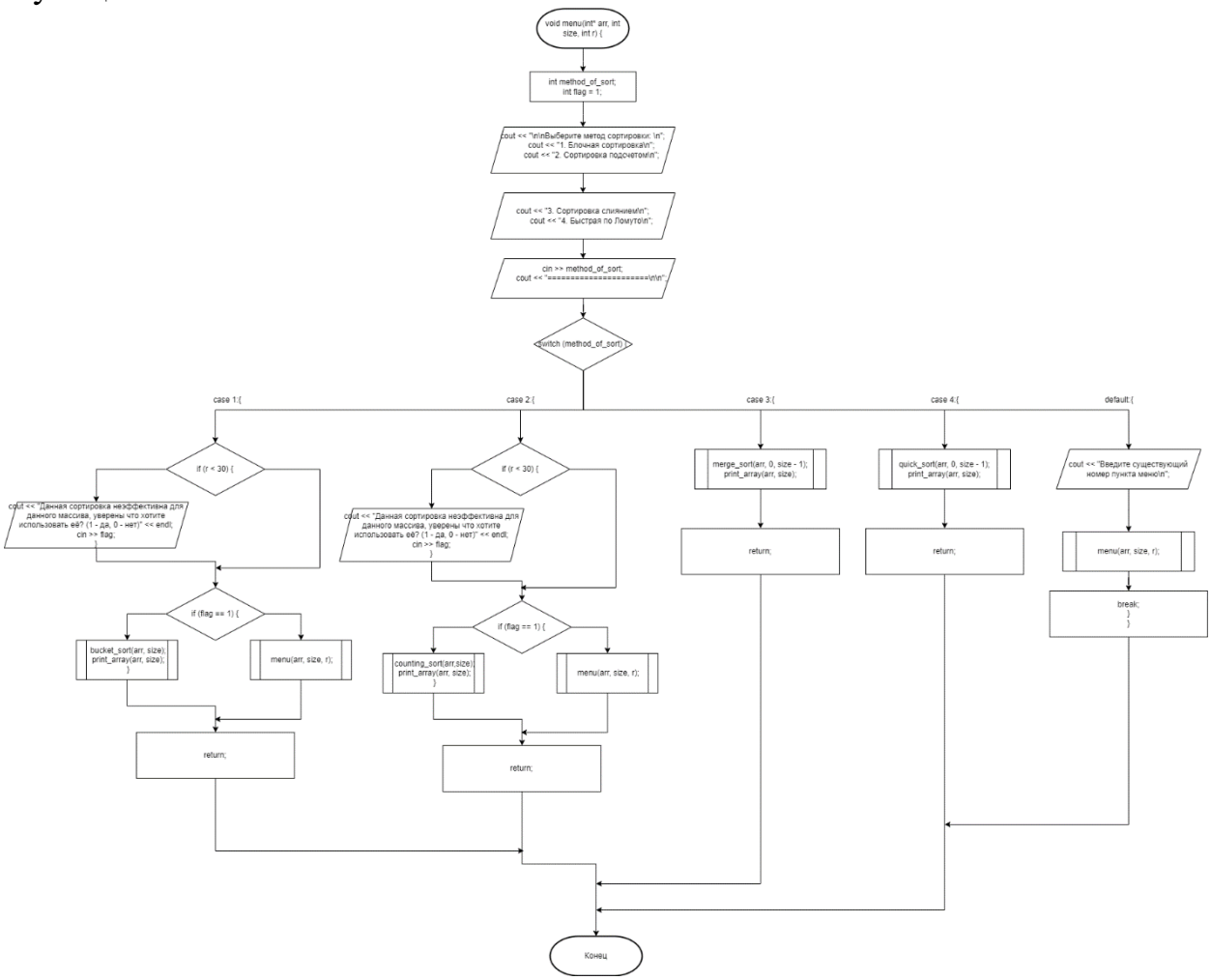
```
        }
        return;
    }
    case 2: {
        if (r > 12) {
            cout << "Данная сортировка неэффективна для данного массива, уверены что хотите использовать её? (1 - да, 0 - нет)" << endl;
            cin >> flag;
        }
        if (flag == 1) {
            counting_sort(arr, size);
            print_array(arr, size);
        }
        else {
            menu(arr, size, r);
        }
        return;
    }
    case 3: {
        merge_sort(arr, 0, size - 1);
        print_array(arr, size);
        return;
    }
    case 4: {
        quick_sort(arr, 0, size - 1);
        print_array(arr, size);
        return;
    }
    default: {
        cout << "Введите существующий номер пункта меню\n";
        menu(arr, size, r);
        break;
    }
    }
}


int main() {
    system("chcp 1251 > Null");
    srand(time(0));

    const int size = 25;
    int arr[size];
    int r = rand() % 99 + 2;

    cout << "В  сформированном массиве встречаются числа меньше чем " << r << "\n\nСформированный масив: \n";
    for (int i = 0; i < size; i++) {
        arr[i] = rand() % r;
        if (i == 0) {
            cout << arr[i];
        }
        else {
            cout << " " << arr[i];
        }
    }

    menu(arr, size, r);



    return 0;
}
```

Блок схема

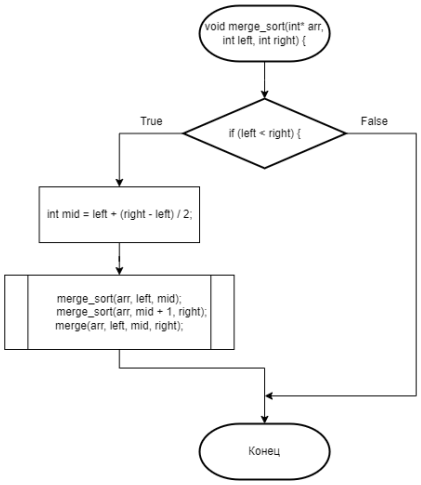
Функция main

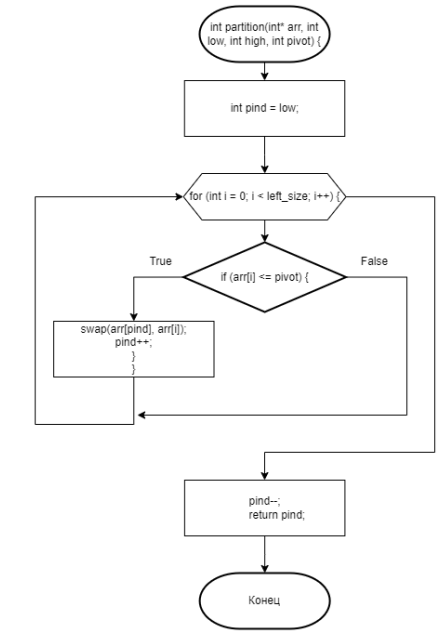


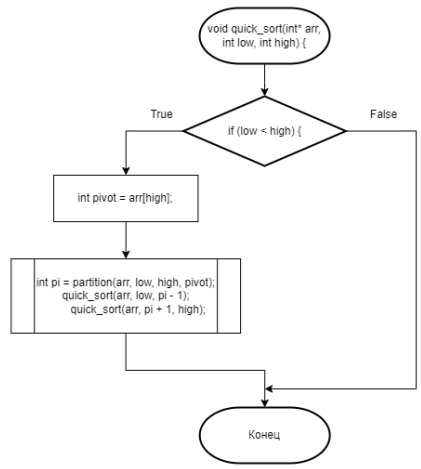
Функция menu

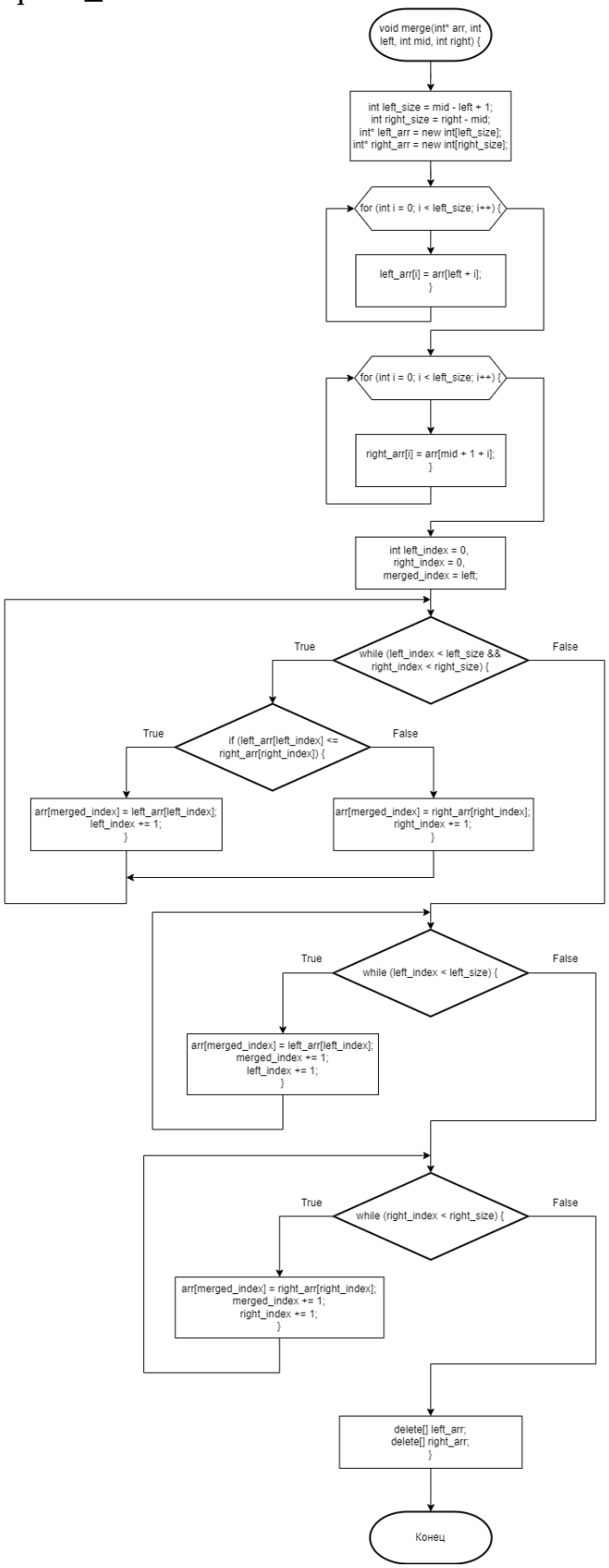


Функции merge, merge_sort, partition, quick_sort

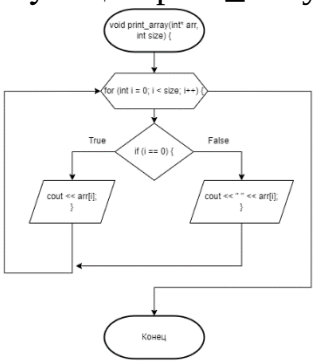


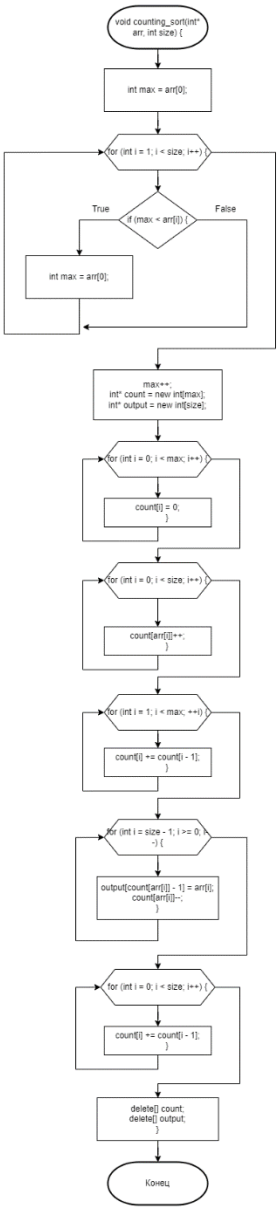


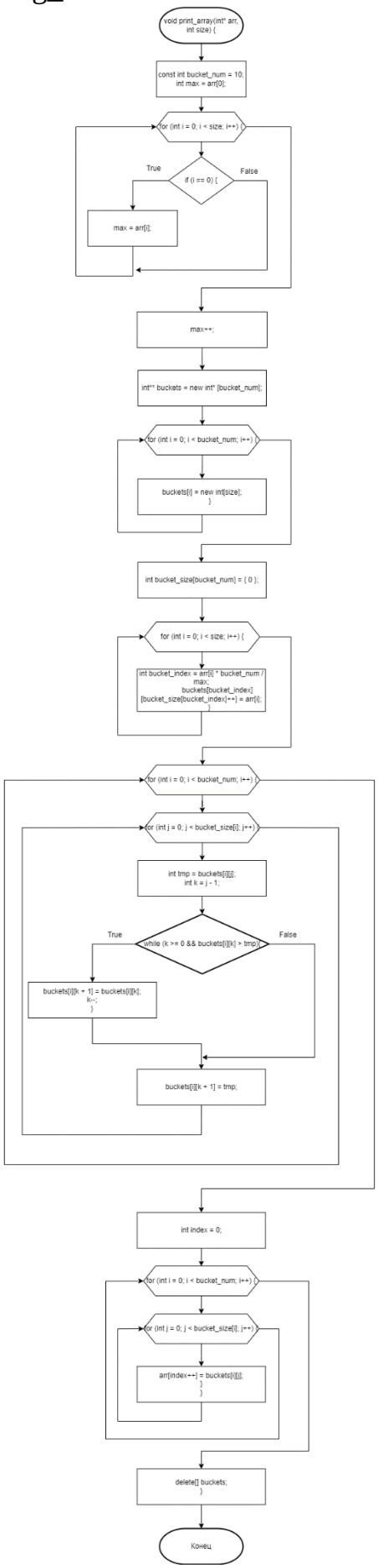




Функции print_array, bucket_sort, counting_sort







Тесты

```
В  сформированном массиве встречаются числа меньше чем 46

Сформированный масив:
40 43 23 45 35 20 23 7 41 4 12 5 18 24 37 35 15 9 31 11 44 45 4 2 14

Выберите метод сортировки:
1. Блочная сортировка
2. Сортировка подсчетом
3. Сортировка слиянием
4. Быстрая по Ломуто
2
========================================================================

Данная сортировка неэффективна для данного массива, уверены что хотите использовать её? (1 - да, 0 - нет)
1
2 4 4 5 7 9 11 12 14 15 18 20 23 23 24 31 35 35 37 40 41 43 44 45 45
C:\Users\MOkASiH\Desktop\Test\x64\Debug\Test.exe (процесс 19616) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно…
```

```
В  сформированном массиве встречаются числа меньше чем 50

Сформированный масив:
9 28 48 3 20 16 38 41 45 40 7 36 10 0 30 24 15 10 33 0 6 45 37 22 2

Выберите метод сортировки:
1. Блочная сортировка
2. Сортировка подсчетом
3. Сортировка слиянием
4. Быстрая по Ломуто
3
========================================================================

0 0 2 3 6 7 9 10 10 15 16 20 22 24 28 30 33 36 37 38 40 41 45 45 48
C:\Users\MOkASiH\Desktop\Test\x64\Debug\Test.exe (процесс 9980) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно…
```

```
В  сформированном массиве встречаются числа меньше чем 35

Сформированный масив:
6 19 0 6 11 18 9 22 9 17 15 19 28 18 9 8 17 20 24 16 26 27 17 26 16

Выберите метод сортировки:
1. Блочная сортировка
2. Сортировка подсчетом
3. Сортировка слиянием
4. Быстрая по Ломуто
1
========================================================================

0 6 6 8 9 9 9 11 15 16 16 17 17 17 18 18 19 19 20 22 24 26 26 27 28
C:\Users\MOkASiH\Desktop\Test\x64\Debug\Test.exe (процесс 3732) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно…
```

```
В  сформированном массиве встречаются числа меньше чем 99

Сформированный масив:
63 63 9 96 38 28 74 64 15 37 94 62 38 31 83 94 8 10 82 56 39 6 38 36 17

Выберите метод сортировки:
1. Блочная сортировка
2. Сортировка подсчетом
3. Сортировка слиянием
4. Быстрая по Ломуто
4
========================================================================

6 8 9 10 15 17 28 31 36 37 38 38 38 39 56 62 63 63 64 74 82 83 94 94 96
C:\Users\MOkASiH\Desktop\Test\x64\Debug\Test.exe (процесс 14344) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно…
```

```
В  сформированном массиве встречаются числа меньше чем 46

Сформированный масив:
27 25 32 37 1 41 17 17 31 23 18 17 16 38 17 0 23 4 0 44 15 29 9 28 31

Выберите метод сортировки:
1. Блочная сортировка
2. Сортировка подсчетом
3. Сортировка слиянием
4. Быстрая по Ломуто
2
========================================================================

Данная сортировка неэффективна для данного массива, уверены что хотите использовать её? (1 - да, 0 - нет)
0


Выберите метод сортировки:
1. Блочная сортировка
2. Сортировка подсчетом
3. Сортировка слиянием
4. Быстрая по Ломуто
1
========================================================================

0 0 1 4 9 15 16 17 17 17 17 18 23 23 25 27 28 29 31 31 32 37 38 41 44
C:\Users\MOkASiH\Desktop\Test\x64\Debug\Test.exe (процесс 11840) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно…
```

```
В  сформированном массиве встречаются числа меньше чем 5

Сформированный масив:
2 3 0 3 1 4 4 2 3 1 3 0 1 2 0 1 3 3 3 1 3 1 0 4 0

Выберите метод сортировки:
1. Блочная сортировка
2. Сортировка подсчетом
3. Сортировка слиянием
4. Быстрая по Ломуто
1
========================================================================

Данная сортировка неэффективна для данного массива, уверены что хотите использовать её? (1 - да, 0 - нет)
1
0 0 0 0 0 1 1 1 1 1 1 2 2 2 3 3 3 3 3 3 3 3 4 4 4
C:\Users\MOkASiH\Desktop\Test\x64\Debug\Test.exe (процесс 6460) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно…
```