Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Институт информационных технологий

Факультет компьютерных технологий

Кафедра информационных систем и технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ЛАБОРАТОРНАЯ РАБОТА №2

по теме: «Разработка вычислительных процессов с использованием составных статических структур данных (массивы)»

Студент:

гр. 481571 Самойлов А.С.

Проверил:

ассистент кафедры ИСиТ ИИТ БГУИР Потоцкий Д.С.

Минск 2024

Вариант 22

Задание:

1. Согласовать вариант задания с преподавателем, внимательно изучить индивидуальное задание к задаче (Таблица 1) и пример.

2. Разработать алгоритм для решения следующей задачи : Объявить одномерный статический массив заданного типа и размера. Заполнить его одним из способов по выбору пользователя: пользовательскими данными с клавиатуры, либо случайными числами в диапазоне от A до B (значения A и B ввести с клавиатуры). Преобразовать массив в соответствии с индивидуальным заданием. Организовать вывод исходного массива и массива после обработки. Используйте по умолчанию алгоритм сортировки пузырьком.

3. Написать программный код реализации составленного алгоритма с учетом требований и ограничений по индивидуальному заданию.

4. Организовать текстовый пользовательский интерфейс в программе. Ввод данных пользователем реализовать с клавиатуры, а вывод результатов выполнения программы – в консоль (на экран).

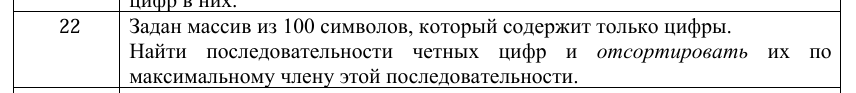
5. Проверить правильность вычислений на тестовых примерах, выполнив серию контрольных прогонов программы.

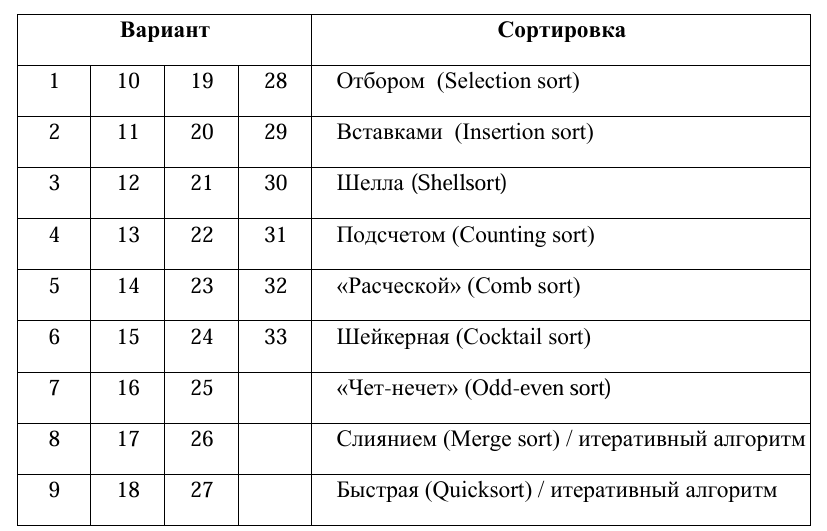
6. Построить укрупненную схему составленного алгоритма (блок-схему работы программы).

7. Дополнительно следует учесть, что правильность вводимых значений не гарантируется. Необходимо обеспечить проверку соответствия входных данных указанным в условии ограничениям и предусмотреть сообщение об ошибке в случае ее обнаружения.

8. Дополнительно реализовать сортировку массива с помощью алгоритма по варианту (Таблица 2).

9. Дополнительно для своего варианта реализации сортировки выполнить измерение времени работы программы и сделать анализ зависимости времени выполнения алгоритма сортировки от конфигурации входных данных (количества элементов и их исходного расположения).





Листинг 1.1 – Программная реализация решения задачи по лабораторной работе № 2 (вариант 22)

#include <iostream>

#include <vector>

void showArray(const std::vector<int>& array)

{

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

{

if (i % 20 == 0 && i != 0)

{

std::cout << "\n";

}

std::cout << array[i] << " ";

}

}

void showEvenSequences(std::vector<std::vector<int>>& allEvenSequences)

{

std::vector<int> sequence;

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

std::cout << "[";

sequence = allEvenSequences[i];

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

std::cout << sequence[i];

if (i < sequence.size() - 1) {

std::cout << ", ";

}

}

std::cout << "]" << std::endl;

}

}

void countingSort(std::vector<int>& array, int minValue, int maxValue)

{

int range = maxValue - minValue + 1;

std::vector<int> count(range, 0);

for (int num : array)

{

count[num - minValue]++;

}

int index = 0;

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

{

while (count[i] > 0)

{

array[index++] = i + minValue;

count[i]--;

}

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

int flag;

bool value = true;

int minValue = 0, maxValue = 0;

int arraySize;

std::cout << "Введите размерность массива (от 1 до 100) \n";

while (!((std::cin >> arraySize) && (arraySize <= 100 && arraySize > 0))) {

std::cin.clear();

while (std::cin.get() != '\n');

std::cout << "Размерность введена неправильно!\n";

}

std::vector<int> array(arraySize);

do

{

std::cout << "Выберите ввод:\n1 - Ручной ввод\n2 - Автоввод" << std::endl;

while (!(std::cin >> flag)) {

std::cin.clear();

while (std::cin.get() != '\n');

std::cout << "Некорректный ввод!\n";

}

switch (flag)

{

case 1:

std::cout << "Заполните массив значениями:\n";

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

{

std::cout << "Введите значение для arr[" << i << "]: ";

while (!(std::cin >> array[i])) {

std::cin.clear();

while (std::cin.get() != '\n');

std::cout << "Некорректный ввод!\n";

}

if (i == 0 || array[i] < minValue)

minValue = array[i];

if (i == 0 || array[i] > maxValue)

maxValue = array[i];

}

value = false;

break;

case 2:

{

int lowerBound, upperBound;

std::cout << "Введите минимальное значение для ввода: ";

std::cin >> lowerBound;

std::cout << "Введите максимальное значение для ввода: ";

std::cin >> upperBound;

if (lowerBound > upperBound)

{

std::cout << "Минимальное значение больше максимального.\n";

break;

}

minValue = lowerBound;

maxValue = upperBound;

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

{

array[i] = rand() % (upperBound - lowerBound + 1) + lowerBound;

}

value = false;

break;

}

default:

std::cout << "Введите корректное значение" << std::endl;

break;

}

} while (value);

for (int num : array)

{

if (maxValue < num)

{

maxValue = num;

}

}

std::cout << "Массив до сортировки:" << std::endl;

showArray(array);

std::vector<std::vector<int>> allEvenSequences;

std::vector<int> currentSequence;

for (int number : array) {

if (number % 2 == 0) {

currentSequence.push\_back(number);

}

else if (!currentSequence.empty()) {

allEvenSequences.push\_back(currentSequence);

currentSequence.clear();

}

}

if (!currentSequence.empty()) {

allEvenSequences.push\_back(currentSequence);

}

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

for (int j = 0; j < allEvenSequences.size() - i - 1; ++j) {

int maxA = \*std::max\_element(allEvenSequences[j].begin(), allEvenSequences[j].end());

int maxB = \*std::max\_element(allEvenSequences[j + 1].begin(), allEvenSequences[j + 1].end());

if (maxA < maxB) {

std::swap(allEvenSequences[j], allEvenSequences[j + 1]);

}

}

}

std::cout << "\nОтсортированные последовательности четных чисел:" << std::endl;

showEvenSequences(allEvenSequences);

bool switchValue = true;

int flagSort;

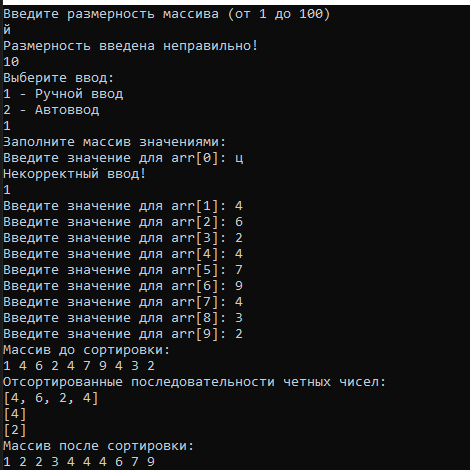
countingSort(array, minValue, maxValue);

std::cout << "Массив после сортировки:" << std::endl;

showArray(array);

return 0;

}



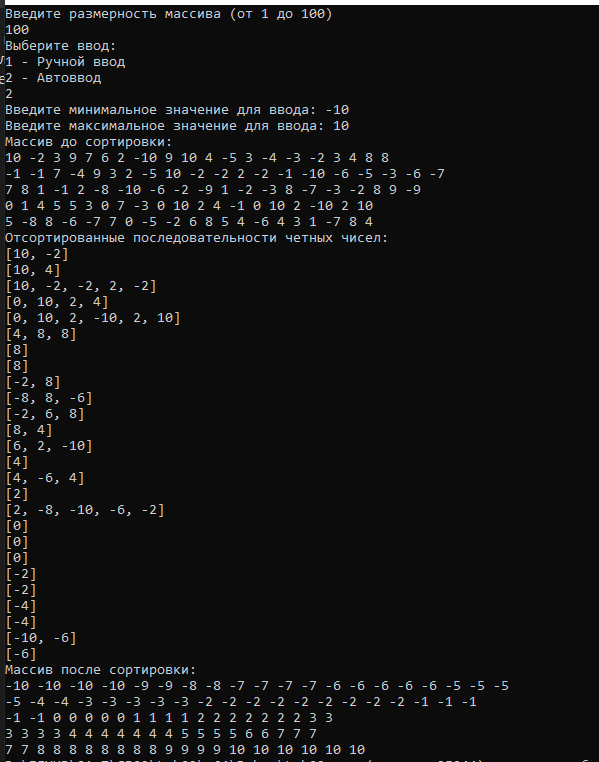


Рисунок 1.1 – Экранные формы результатов работы программы по лабораторной работе №2

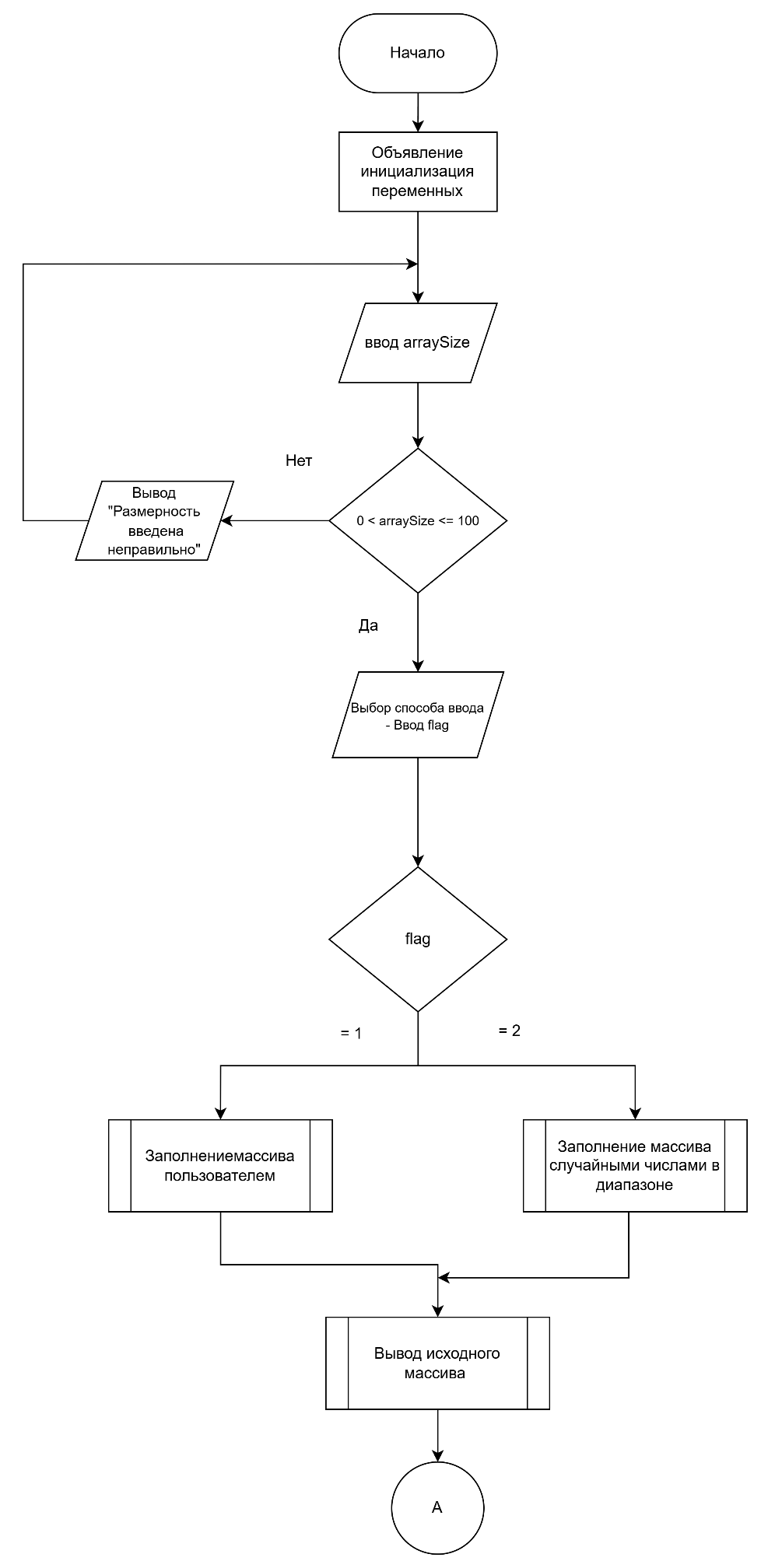


Рисунок 1.2 – Блок-схема алгоритма решения лабораторной работы №2

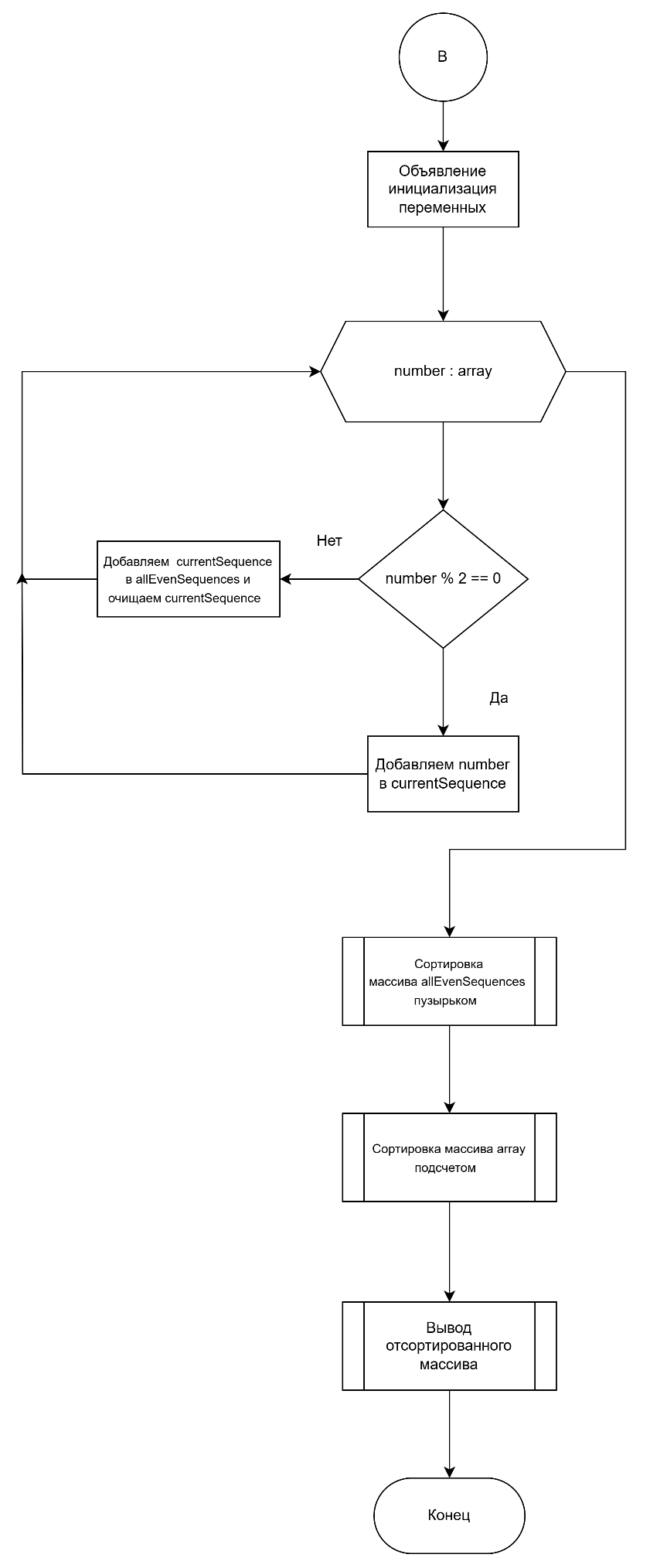


Рисунок 1.3 – Блок-схема алгоритма решения лабораторной работы №2