**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа энергетики

Отделение электроэнергетики и электротехники

Направление: 09.04.03 Прикладная информатика

Дисциплина: Программирование и основы алгоритмизации

Отчет

по лабораторной работе №4

**Одномерные и многомерные массива**

Вариант 12

|  |  |  |
| --- | --- | --- |
| Выполнил:  студент гр. О-5КМ41 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | Рустамов Р.А. |
| Проверил: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата) | Гончаров А.С. |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (балл) |  |

Томск-2025

# ХОД РАБОТЫ

Задача 1: Дан массив из N элементов (натуральные числа). Определить количество элементов, имеющих четные порядковые номера и являющихся нечетными числами.

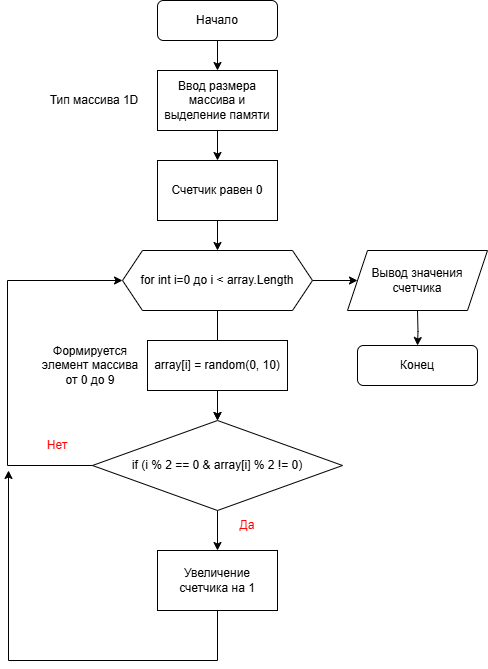


Рисунок – Блок схема поиска элемента с четным индексом и нечетным элементом

|  |  |
| --- | --- |
|  |  |

Рисунок – Программная реализация поиска элемента с четным индексом и нечетным элементом

1. Сформировался массив 1D;
2. Проверим на четность индекс и нечетность элемент:

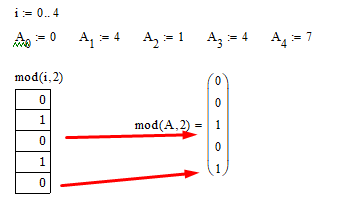


Рисунок – Пример работы в Mathcad 15

Задача 2: Результаты соревнований по пятиборью представлены в виде матрицы 5х5 (5 спортсменов и 5 видов соревнований), в которой указаны места, занятые каждым спортсменом в данном виде. Найти лучшего спортсмена (наименьшая сумма мест).

В последующих трех задачах использовать следующее:

Таблица футбольного чемпионата задана квадратной матрицей порядка n, в которой все элементы, принадлежащие главной диагонали, равны нулю, а каждый элемент, не принадлежащий главной диагонали, равен 2,1 или 0 (число очков, набранных в игре: 2 — выигрыш, 1 — ничья, 0 — проигрыш).

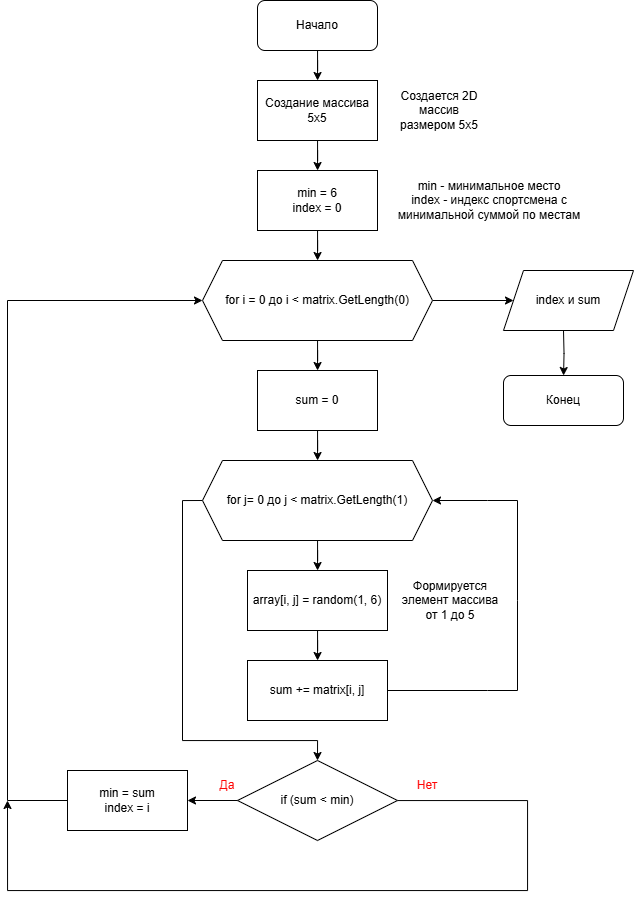


Рисунок – Блок схема поиска лучшего спортсмена по пятиборью

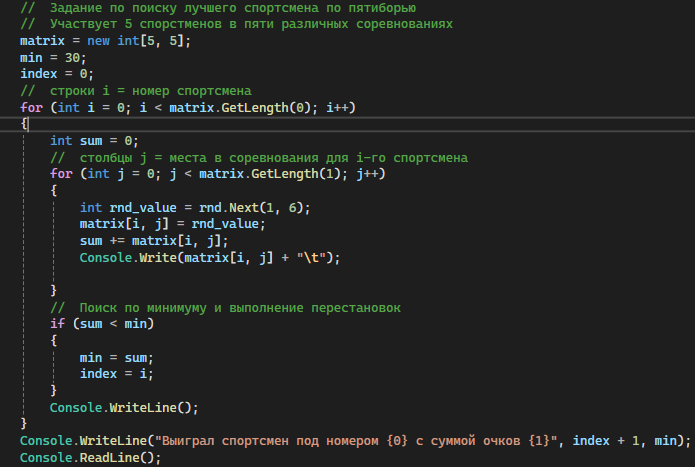


Рисунок – Программный код по поиску лучшего спортсмена

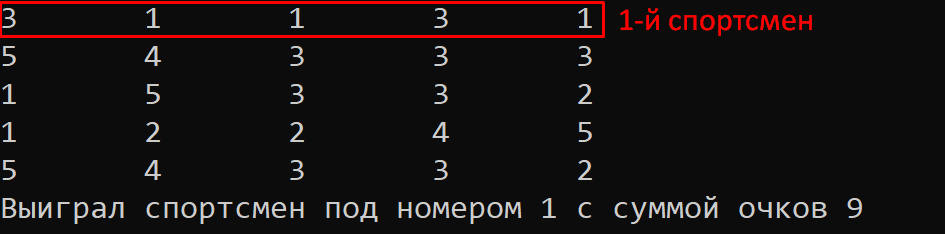


Рисунок – Пример работы кода

Дополнительное задание №1

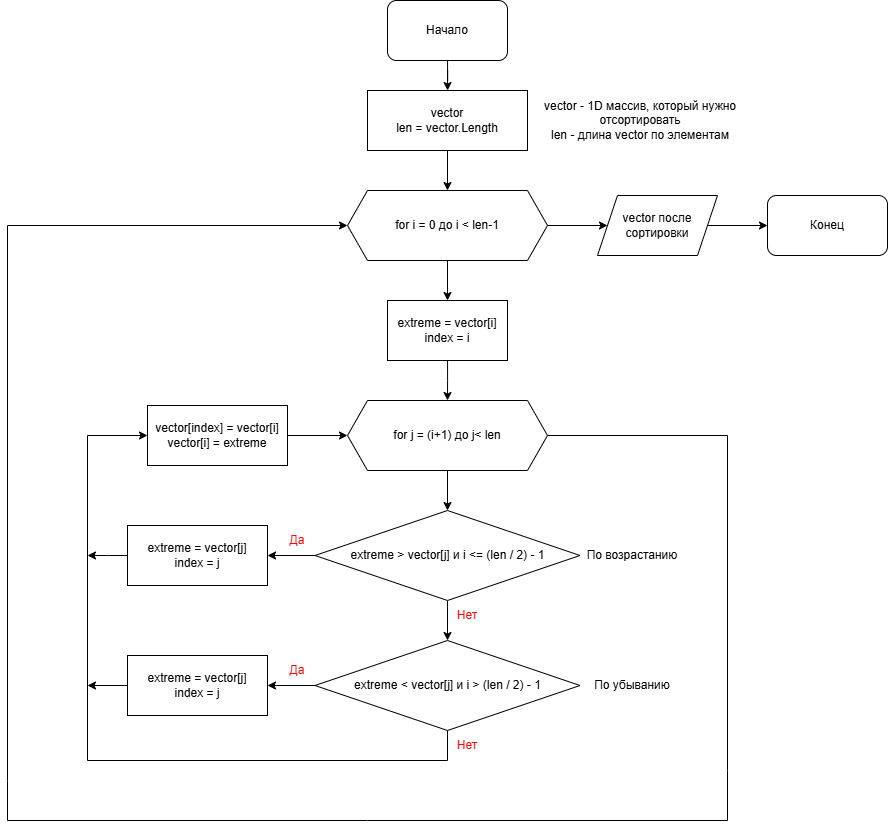


Рисунок – Блок кода метода простых вставок до середины по возрастанию, с середины по убыванию

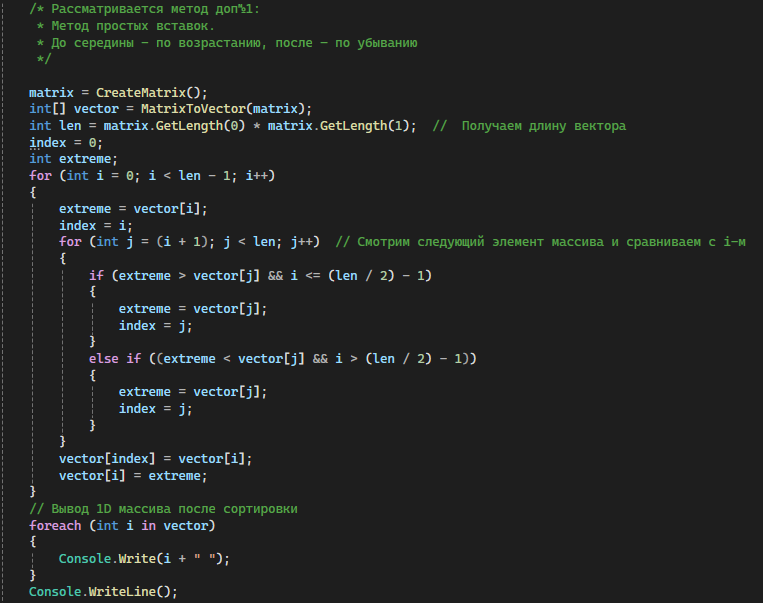


Рисунок –Метод простых вставок

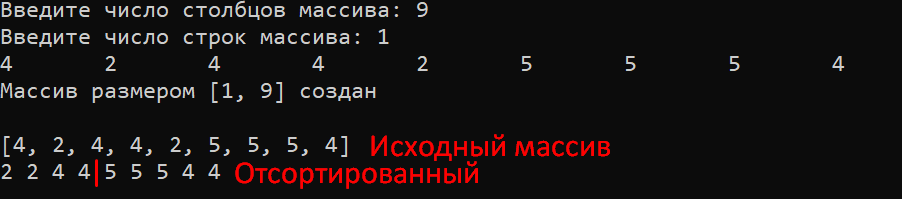


Рисунок –Результат сортировки

Дополнительная задача 2: Реализация алгоритма быстрой сортировки.

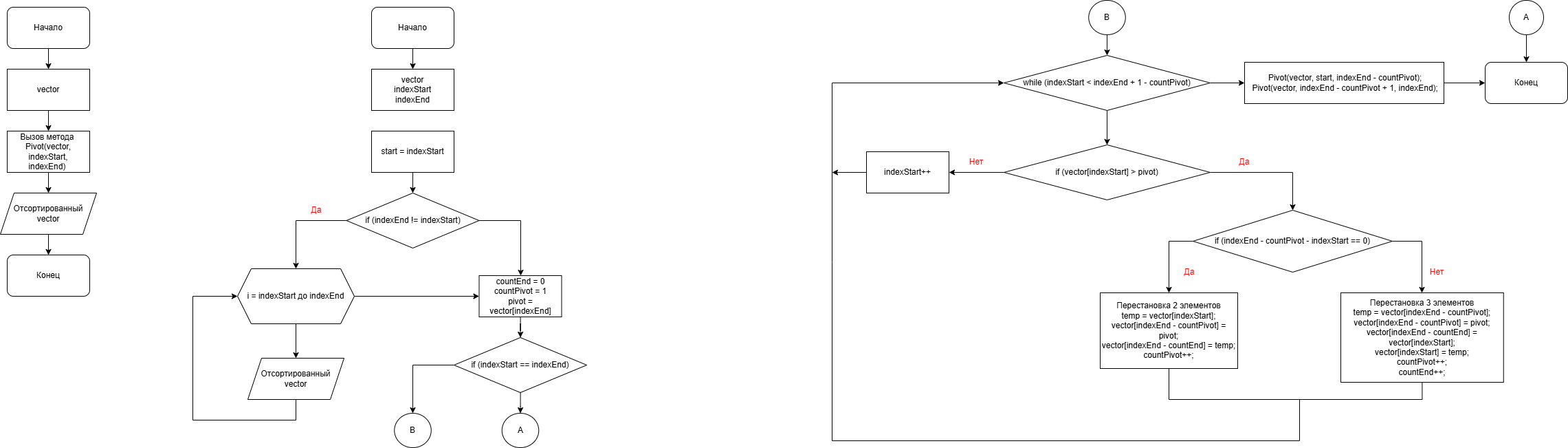


Рисунок – Блок схема вызова метода быстрой сортировки в case

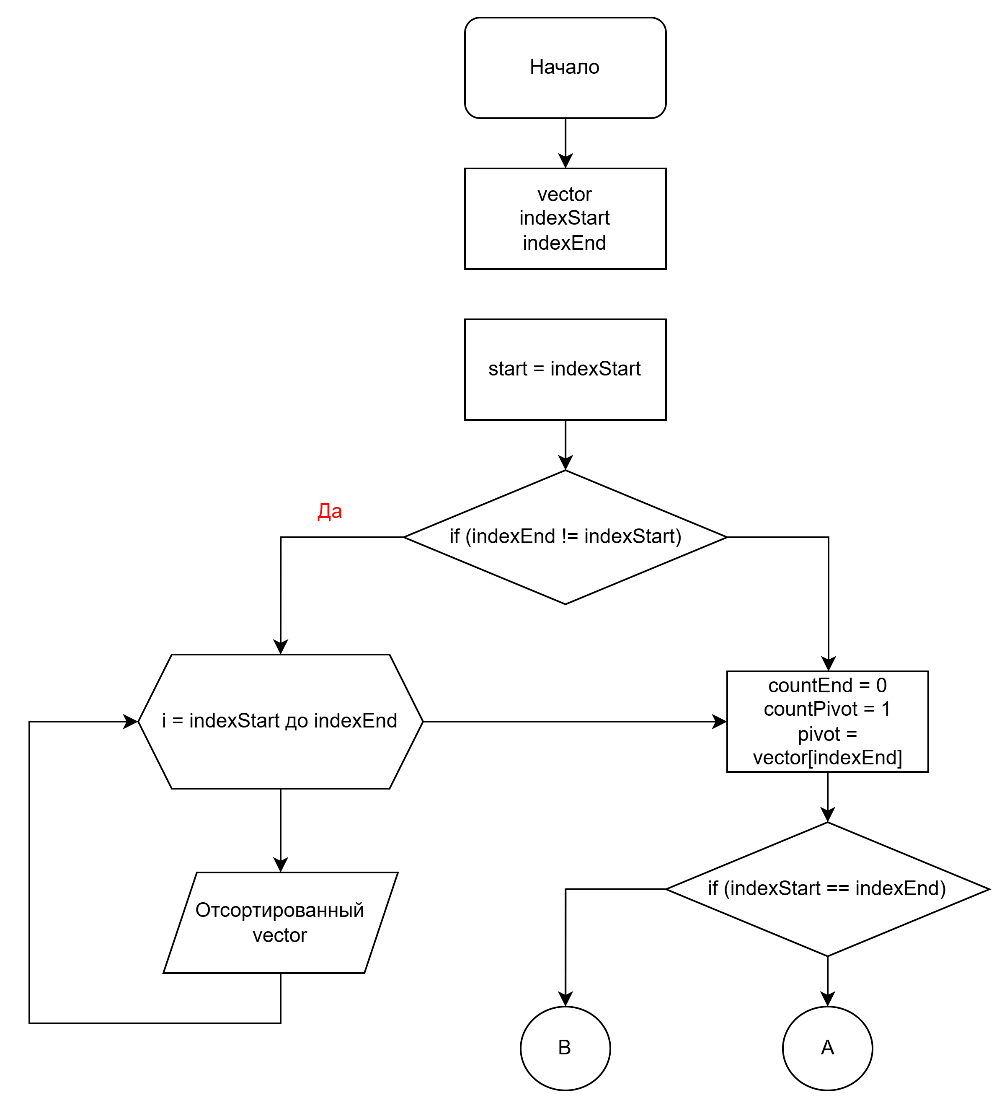


Рисунок – Блок схема метода быстрой сортировки

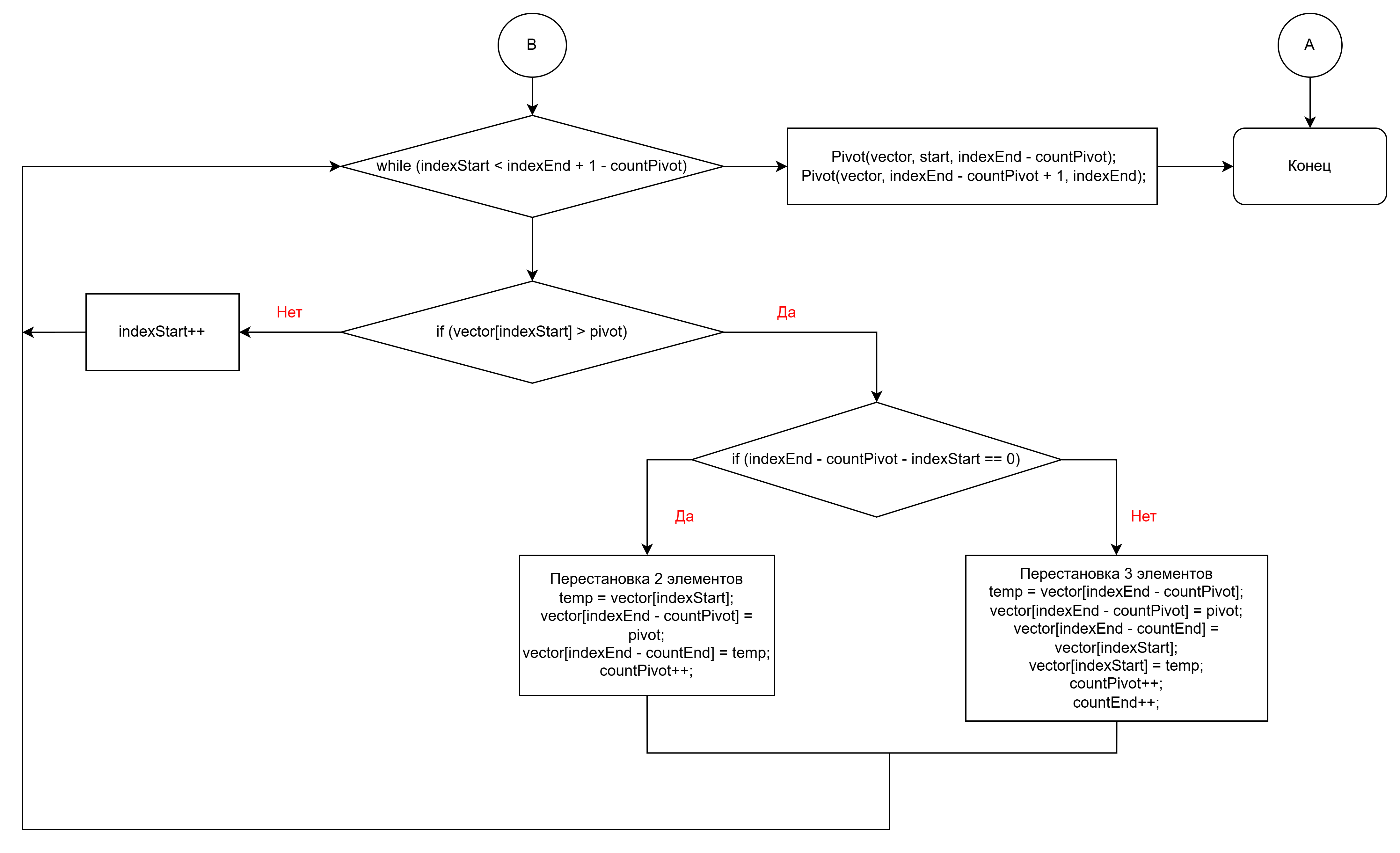


Рисунок 11 – Блок схема метода быстрой сортировки конец

|  |
| --- |
| static void Pivot(int[] vector, int indexStart, int indexEnd)  {  int start = indexStart;  if (indexEnd != indexStart)  {  for (int i = indexStart; i <= indexEnd; i++)  {  Console.Write(vector[i].ToString() + "\t");  }  Console.WriteLine();  }  int countEnd = 0;  int countPivot = 1;  int pivot = vector[indexEnd];  if (indexStart == indexEnd)  {  return;  }  while (indexStart < indexEnd + 1 - countPivot)  {  // Перестановка элементов массива  if (vector[indexStart] > pivot)  {  // Перестановка двух элементов, когда  // pivot и сравниваемый элемент соседи  if (indexEnd - countPivot - indexStart == 0)  {  var temp = vector[indexStart];  vector[indexEnd - countPivot] = pivot;  vector[indexEnd - countEnd] = temp;  countPivot++;  }  else  {  var temp = vector[indexEnd - countPivot];  vector[indexEnd - countPivot] = pivot;  vector[indexEnd - countEnd] = vector[indexStart];  vector[indexStart] = temp;  countPivot++;  countEnd++;  }  }  else  {  indexStart++;  }  }  Console.WriteLine();  Pivot(vector, start, indexEnd - countPivot);  Pivot(vector, indexEnd - countPivot + 1, indexEnd);  } |

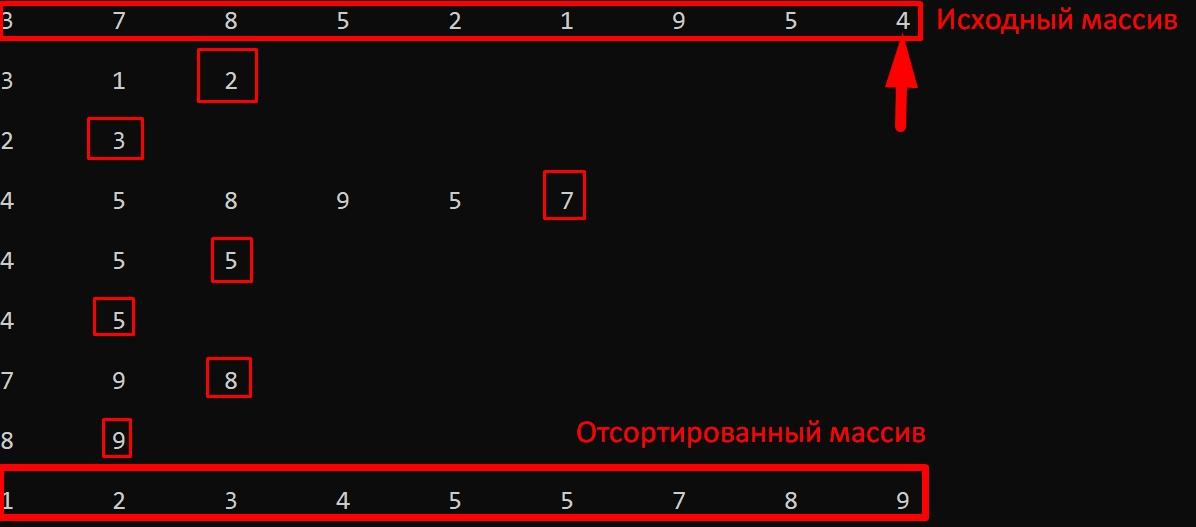


Рисунок – Результат работы быстрой сортировки с опорным элементом в конце

# Вывод

В ходе выполнения лабораторной работы № 4 были изучены:

1. Одномерные и многомерные массивы.
2. Алгоритмы обработки данных в массиве.