**2 Структуры данных. Алгоритмы обработки структур данных.**

Задание 1. Во всех задачах сформировать и вывести на экран одномерный массив согласно варианту. Произвести его сортировку и бинарный поиск числа k (k -вводится с клавиатуры). Напишите программу для решения задачи: пусть даны натуральное число п, целые числа а1, ..., аn. Получите сумму положительных, число отрицательных и число нулевых членов последовательности а1, ..., аn.

Листинг программы:

public class Program

{

public static void Main(string[] args)

{

Console.WriteLine("Введите натуральное число n: ");

int arrLenght = int.Parse(Console.ReadLine());

int[] arrayNumbers = new int[arrLenght];

Random random = new Random();

Console.WriteLine("Сформированный массив:");

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

{

arrayNumbers[i] = random.Next(-10, 11);

}

Console.WriteLine(string.Join(" ,", arrayNumbers));

int positiveSum = 0;

int negativeCount = 0;

int zeroCount = 0;

foreach(int number in arrayNumbers)

{

if (number > 0)

{

positiveSum += number;

}

else if (number < 0)

{

negativeCount++;

}

else

{

zeroCount++;

}

}

Console.WriteLine($"Сумма положительных элементов: {positiveSum}");

Console.WriteLine($"Количество отрицательных элементов: {negativeCount}");

Console.WriteLine($"Количество нулевых элементов: {zeroCount}");

Array.Sort(arrayNumbers);

Console.WriteLine("Отсортированный массив:");

Console.WriteLine(string.Join(" ,", arrayNumbers));

Console.WriteLine("Введите K для поиска: ");

int k = int.Parse(Console.ReadLine());

int index = Array.BinarySearch(arrayNumbers, k);

if (index >= 0)

{

Console.WriteLine($"Число {k} находится под индексом {index}.");

}

else

{

Console.WriteLine($"Число {k} не найдено.");

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| n: 20, k: 5 | positiveSum = 44;  negativeCount = 11;  zeroCount = 0; |

Анализ результатов:

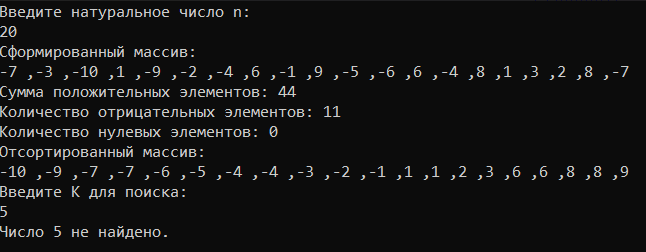
****

Рисунок 1.1 – Результат работы программы