**5 Массивы. Класс Array**

Задание 1. В массиве вещественных чисел найти количество отрицательных

элементов.

Листинг программы для выражения A:

double[] arr = { -1, 4, 5, -7, 9, -5 }; int count = 0;

foreach (double num in arr)

{if (num < 0) {count++;}}

Console.WriteLine("Количество отрицательных элементов: " + count);

Таблица 5.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 4 | Количество отрицательных элементов: 3 |

Источник: собственная разработка

Анализ результатов:



Рисунок 5.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Во всех задачах сформировать и вывести на экран одномерный массив согласно варианту. Произвести его сортировку и бинарный поиск числа k (k - вводится с клавиатуры). Пусть даны целые числа а 1, ..., а 100. Получите новую последовательность из 100 целых чисел, заменяя а i - нулями, если значение не равно максимальному из а 1, ..., а 100, и заменяя а i, единицей - в противном случае (i = 1, ..., 100).

Листинг программы:

int[] arr = {5, 2, 9, 15, 50, 96, 94, 100}; Array.Sort(arr);

int max = arr[0];

for (int i = 1; i < arr.Length; i++)

{if (arr[i] > max){max = arr[i];}}

int[] newArr = new int[arr.Length];

for (int i = 0; i < arr.Length; i++)

{if (arr[i] != max){newArr[i] = 0;} else{newArr[i] = 1;}}

Console.WriteLine("Исходный массив:"); foreach (int num in arr)

{Console.Write(num + " ");} Console.WriteLine("\n");

Console.WriteLine("Новый массив:");foreach (int num in newArr)

{Console.Write(num + " ");} Console.WriteLine("\n");

Console.Write("Какое число найти в исх. массиве? k: ");

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

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

if (index >= 0){

Console.WriteLine("Число " + k + " найдено в массиве на позиции " + index);} else{Console.WriteLine("Число " + k + " не найдено в массиве");}

Таблица 5.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 4 | Исходный массив:  2 5 9 15 50 94 96 100  Новый массив:  0 0 0 0 0 0 0 1  Число 50 найдено в массиве на позиции 4 |

Источник: собственная разработка

Анализ результатов:

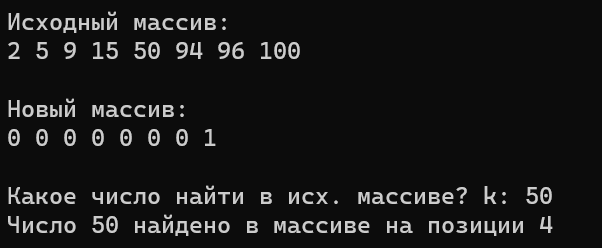


Рисунок 5.2 – Результат работы программы

Источник: собственная разработка

Задание 3. В каждой задаче задается квадратная целочисленная матрица NxN (значение N вводится с клавиатуры, N<10). Программа должна заполнять матрицу случайными числами из диапазона [a, b] (a, b вводятся с клавиатуры) и осуществлять вывод на экран исходной матрицы. Затем необходимо произвести необходимые действия и напечатать результаты. Вычислить сумму чётных элементов. Вычислить количество положительных элементов каждого столбца.

Листинг программы:

Console.Write("Введите значение N: ");

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

if (n > 10){Console.WriteLine("Ошибка: N должно быть меньше 10");

return;}

Console.Write("Диапозон от, a: "); int a = int.Parse(Console.ReadLine());

Console.Write("Диапозон до, b: "); int b = int.Parse(Console.ReadLine());

int[,] matrix = new int[n, n]; Random random = new Random();

for (int i = 0; i < n; i++){for (int j = 0; j < n; j++){

matrix[i, j] = random.Next(a, b + 1);}}

Console.WriteLine("Исходная матрица:");

for (int i = 0; i < n; i++){for (int j = 0; j < n; j++){

Console.Write("{0,4}", matrix[i, j]);} Console.WriteLine();}

int sum = 0; for (int i = 0; i < n; i++){for (int j = 0; j < n; j++){

if (matrix[i, j] % 2 == 0){sum += matrix[i, j];}}}

Console.WriteLine("Сумма четных элементов матрицы: {0}", sum);

for (int j = 0; j < n; j++){int positiveCount = 0; for (int i = 0; i < n; i++){

if (matrix[i, j] > 0) {positiveCount++;}}

Console.WriteLine("Количество положительных элементов в столбце {0}: {1}", j, positiveCount);}

Таблица 5.3 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 3; -5; 5 | Исходная матрица:  -3 1 -5  5 5 1  0 3 1  Сумма четных элементов матрицы: 0  Количество положительных элементов в столбце 0: 1  Количество положительных элементов в столбце 1: 3  Количество положительных элементов в столбце 2: 2 |

Источник: собственная разработка

Анализ результатов:

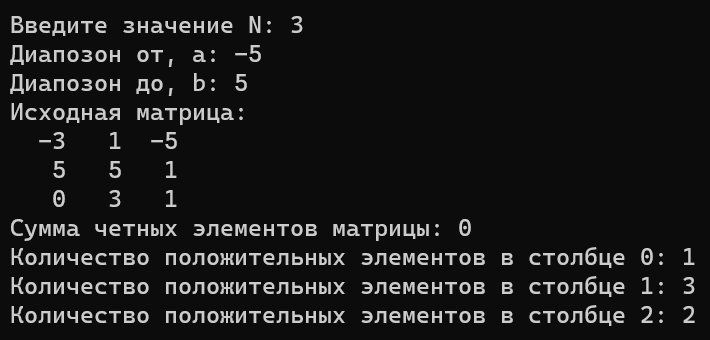


Рисунок 5.3 – Результат работы программы

Источник: собственная разработка

Задание 4. В поезде 18 вагонов, в каждом по 36 мест. Информация о проданных на поезд билетах хранится в двумерном массиве, номера строк которого соответствуют номерам вагонов, а номера столбцов – номерам мест. Если билет на то или иное место продан, то соответствующий элемент массива имеет значение 1, в противном случае – 0. Составить программу, определяющую, имеются ли свободные места в том или ином вагоне поезда.

Листинг программы:

int[,] tickets = new int[18, 36]; Random rnd = new Random();

for (int i = 0; i < 18; i++)

{ for (int j = 0; j < 36; j++) {

tickets[i, j] = rnd.Next(0, 2); }}

for (int i = 0; i < 18; i++) // выводим массивчик

{

for (int j = 0; j < 36; j++) { Console.Write(tickets[i, j] + " "); }

Console.WriteLine();

}

for (int i = 0; i < 18; i++)

{

bool freeSeats = false; // Флаг, указывающий на наличие свободных мест в вагоне

for (int j = 0; j < 36; j++)

{

if (tickets[i, j] == 0) // Если нашли свободное место в вагоне, устанавливаем флаг и выходим из цикла

{

freeSeats = true;

break;

}

}

if (freeSeats) // Если в вагоне есть свободные места, выводим сообщение на экран

{ Console.WriteLine("В вагоне " + (i + 1) + " есть свободные места.");}

}

Таблица 5.4 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 6 | В вагоне 1 есть свободные места.  В вагоне 2 есть свободные места.  В вагоне 3 есть свободные места.  В вагоне 4 есть свободные места.  В вагоне 5 есть свободные места.  В вагоне 6 есть свободные места.  В вагоне 7 есть свободные места.  В вагоне 8 есть свободные места.  В вагоне 9 есть свободные места.  В вагоне 10 есть свободные места.  В вагоне 11 есть свободные места.  В вагоне 12 есть свободные места.  В вагоне 13 есть свободные места.  В вагоне 14 есть свободные места.  В вагоне 15 есть свободные места.  В вагоне 16 есть свободные места.  В вагоне 17 есть свободные места.  В вагоне 18 есть свободные места. |

Источник: собственная разработка

Анализ результатов:

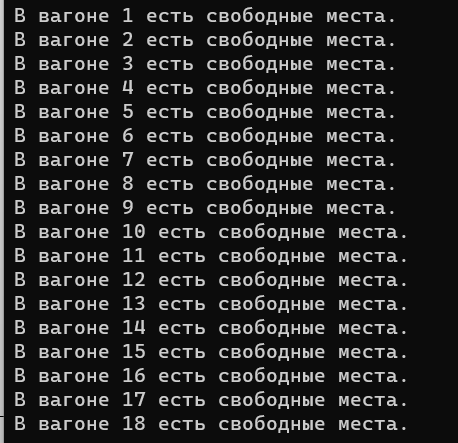


Рисунок 5.4 – Результат работы программы

Источник: собственная разработка

Задание 5. Разработать рекурсивный алгоритм и программу решения задачи, в которой вычислить: f(n)=1/(n+5)!. Исходные данные вводятся с клавиатуры.

Листинг программы:

static double CalculateFunction(int n){if (n == 0){return 1;}else{

double factorial = 1; int fac = (n + 5);

for (int i = 2; i <= fac; i++) {factorial = factorial \* i;}double res = 1/factorial;

return res;}} static void Main(string[] args){

Console.Write("Введите значение n: ");int n = int.Parse(Console.ReadLine());

double result = CalculateFunction(n);

Console.WriteLine("f({0}) = {1}", n, result);}

Таблица 5.5 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 6 | f(1) = 0,001388888888888889 |

Источник: собственная разработка

Анализ результатов:



Рисунок 5.5 – Результат работы программы

Источник: собственная разработка