**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ**

***Институт Принтмедиа и информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 18**

**Дисциплина:** Введение в программирование

**Выполнил(а): студент(ка) группы \_191-726\_\_**

\_\_\_\_\_Мухина Н.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия И.О.)

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Проверил: \_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_***

(Фамилия И.О., степень, звание) **(Оценка)**

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва**

**2019**

Оглавление

[Теория: 2](#_Toc28119746)

[Задания: 3](#_Toc28119747)

[Код программы: 4](#_Toc28119748)

[Результат программы: 10](#_Toc28119749)

# Теория:

Одномерный массив представляет собой список связанных переменных. Такие списки часто применяются в программировании. Например, в одномерном массиве можно хранить учетные номера активных пользователей сети или текущие средние уровни достижений бейсбольной команды. Для того чтобы воспользоваться массивом в программе, требуется двухэтапная процедура, поскольку в C# массивы реализованы в виде объектов. Во-первых, необходимо объявить переменную, которая может обращаться к массиву. И во-вторых, нужно создать экземпляр массива, используя оператор new. Так, для объявления одномерного массива обычно применяется следующая общая форма: тип[] имя\_массива = new тип[размер]; где тип объявляет конкретный тип элемента массива. Тип элемента определяет тип данных каждого элемента, составляющего массив. Обратите внимание на квадратные скобки, которые сопровождают тип. Они указывают на то, что объявляется одномерный массив. А размер определяет число элементов массива.

Обратимся к конкретному примеру. В приведенной ниже строке кода создается массив типа int, который составляется из десяти элементов и связывается с переменной ссылки на массив, именуемой sample. int[] sample = new int[10];

В переменной sample хранится ссылка на область памяти, выделяемой для массива оператором new. Эта область памяти должна быть достаточно большой, чтобы в ней могли храниться десять элементов массива типа int. Как и при создании экземпляра класса, приведенное выше объявление массива можно разделить на два отдельных оператора. Например: int[] sample; sample = new int[10];

Доступ к отдельному элементу массива осуществляется по индексу: Индекс обозначает положение элемента в массиве. В языке C# индекс первого элемента всех массивов оказывается нулевым.

# Задания:

1. Дана квадратная матрица A порядка M (M — нечетное число). Начиная с элемента A1,1 и перемещаясь против часовой стрелки, вывести все ее элементы по спирали: первый столбец, последняя строка, последний столбец в обратном порядке, первая строка в обратном порядке, оставшиеся элементы второго столбца и т. д.; последним выводится центральный элемент матрицы.

2. Дана матрица размера M × N и целое число K (1 ≤ K ≤ M). Найти сумму и произведение элементов K-й строки данной матрицы.

3. Дана матрица размера M × N. Найти номер ее столбца с наименьшим произведением элементов и вывести данный номер, а также значение наименьшего произведения.

4. Дана матрица размера M × N. В каждом ее столбце найти количество элементов, больших среднего арифметического всех элементов этого столбца

5. Дана целочисленная матрица размера M × N. Найти номер первого из ее столбцов, содержащих только нечетные числа. Если таких столбцов нет, то вывести 0.

# Код программы:

Листинг 1 – Задание 1

|  |
| --- |
| 1. using System; 2. using System.Collections.Generic; 3. using System.Linq; 4. using System.Text; 5. using System.Threading.Tasks; 6. namespace \_1\_ 7. { 8. class Program 9. { 10. static void Main(string[] args) 11. { 12. Console.Write("Введите размерность матрицы:"); 13. int n = int.Parse(Console.ReadLine()); 14. int[,] mas = new int[n, n]; 15. Console.WriteLine("Введите элементы матрицы : "); 16. int i, j; 17. for (i = 0; i < n; i++) 18. for (j = 0; j < n; j++) 19. { 20. Console.Write($"[{i}][{j}] = "); 21. mas[i, j] = int.Parse(Console.ReadLine()); 22. } 23. for (i = 0; i < n; i++) 24. { 25. for (j = 0; j < n; j++) 26. { 27. Console.Write($"{mas[i, j]}\t"); 28. } 29. Console.WriteLine(); 30. } 31. Console.Write("Матрица выведенная по спирали:\n"); 32. for (i = 0; i < n; i++) 33. Console.WriteLine(mas[i, 0]); 34. for (i = n - 1; i < n; i++) 35. for (j = 1; j < n; j++) 36. Console.WriteLine(mas[i, j]); 37. for (i = n - 2; i > 0; i--) 38. Console.WriteLine(mas[i, n - 1]); 39. for (j = n - 1; j > 0; j--) 40. Console.WriteLine(mas[0, j]); 41. for (i = 1; i < n - 1; i++) 42. Console.WriteLine(mas[i, 1]); 43. for (j = 2; j < n - 1; j++) 44. Console.WriteLine(mas[n - 2, j]); 45. for (i = n - 3; i > 0; i--) 46. Console.WriteLine(mas[i, n - 2]); 47. for (j = n - 3; j > 1; j--) 48. Console.WriteLine(mas[1, j]); 49. for (i = n / 2; i < n / 2; i++) 50. Console.WriteLine(mas[i, i]); 51. Console.ReadKey(); 52. }}} |

Листинг 2 – Задание 2

|  |
| --- |
| 1. using System; 2. using System.Collections.Generic; 3. using System.Linq; 4. using System.Text; 5. using System.Threading.Tasks; 6. namespace \_2 7. { 8. class Program 9. { 10. static void Main(string[] args) 11. { 12. int M, N, K, a = 0, b = 1, i, j; 13. Console.WriteLine("Введите размер матрицы M и N"); 14. M = int.Parse(Console.ReadLine()); 15. N = int.Parse(Console.ReadLine()); 16. Console.WriteLine("Введите число K номер строки"); 17. K = int.Parse(Console.ReadLine()); 18. int[,] A = new int[M, N]; 19. Console.WriteLine("Введите элементы"); 20. for (i = 0; i < M; i++) 21. { 22. for (j = 0; j < N; j++) 23. { 24. A[i, j] = int.Parse(Console.ReadLine()); 25. } 26. } 27. for (i = 0; i < M; i++) 28. { 29. for (j = 0; j < N; j++) 30. { 31. Console.Write(" " + A[i, j]); 32. } 33. Console.WriteLine(""); 34. } 35. for (i = K - 1; i < K; i++) 36. { 37. for (j = 0; j < N; j++) 38. { 39. a += A[i, j]; 40. b \*= A[i, j]; 41. } 42. } 43. Console.WriteLine("Сумма элементов {0} строки = {1}, а произведение элементов = {2} ", K, a, b); 44. Console.ReadKey(); 45. } 46. } 47. } |

Листинг 3 – Задание 3

|  |
| --- |
| 1. using System; 2. using System.Collections.Generic; 3. using System.Linq; 4. using System.Text; 5. using System.Threading.Tasks; 6. namespace \_3 7. { 8. class Program 9. { 10. static void Main(string[] args) 11. { 12. int M, N, Min = 1, i, j, y = 0, p = 1; 13. Console.WriteLine("Введите размер матрицы M и N"); 14. M = int.Parse(Console.ReadLine()); 15. N = int.Parse(Console.ReadLine()); 16. int[,] A = new int[M, N]; 17. Console.WriteLine("Введите элементы"); 18. for (i = 0; i < M; i++) 19. { 20. for (j = 0; j < N; j++) 21. { 22. A[i, j] = int.Parse(Console.ReadLine()); 23. } 24. } 25. for (i = 0; i < M; i++) 26. { 27. for (j = 0; j < N; j++) 28. { 29. Console.Write(" " + A[i, j]); 30. } 31. Console.WriteLine(""); 32. } 33. for (i = 0; i < M; i++) 34. Min \*= A[i, 0]; 35. for (j = 1; j < N; j++) 36. { 37. for (i = 0; i < M; i++) 38. { 39. p \*= A[i, j]; 40. } 41. if (p < Min) 42. { 43. Min = p; 44. y = j; 45. } 46. } 47. Console.WriteLine("Номер столбца с минимальный произведение элементов- {0}, а также значение = {1} ", y + 1, Min); 48. Console.ReadKey(); 49. } 50. } 51. } |

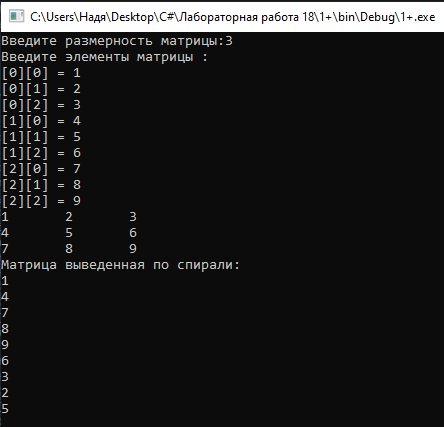
Листинг 4 – Задание 4

|  |
| --- |
| 1. using System; 2. using System.Collections.Generic; 3. using System.Linq; 4. using System.Text; 5. using System.Threading.Tasks; 6. namespace \_4 7. { 8. class Program 9. { 10. static void Main(string[] args) 11. { 12. int M, N, i, j, sr, k; 13. double Sr; 14. Console.WriteLine("Введите размер матрицы M и N"); 15. M = int.Parse(Console.ReadLine()); 16. N = int.Parse(Console.ReadLine()); 17. int[,] A = new int[M, N]; 18. Console.WriteLine("Введите элементы"); 19. for (i = 0; i < M; i++) 20. { 21. for (j = 0; j < N; j++) 22. { 23. A[i, j] = int.Parse(Console.ReadLine()); 24. } 25. } 26. for (i = 0; i < M; i++) 27. { 28. for (j = 0; j < N; j++) 29. { 30. Console.Write(" " + A[i, j]); 31. } 32. Console.WriteLine(""); 33. } 34. for (j = 0; j < N; j++) 35. { 36. k = 0; 37. sr = 0; 38. for (i = 0; i < M; i++) 39. { 40. sr += A[i, j]; 41. } 42. Sr = sr / M; 43. for (i = 0; i < M; i++) 44. { 45. if (A[i, j] > Sr) 46. k++; 47. } 48. Console.WriteLine("В {0} столбце количество элементов, большихсреднего арифметичкеского = {1} ", j + 1, k) 49. } 50. Console.ReadKey(); 51. } 52. } 53. } |

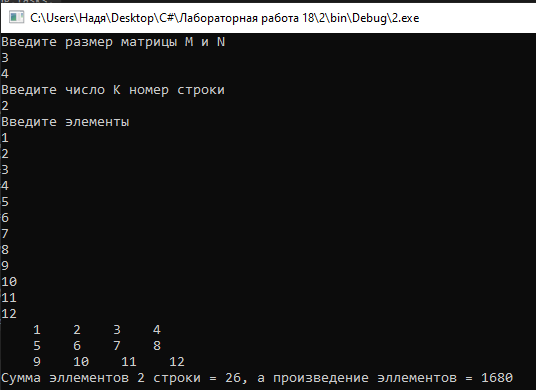
Листинг 5 – Задание 5

|  |
| --- |
| 1. using System; 2. using System.Collections.Generic; 3. using System.Linq; 4. using System.Text; 5. using System.Threading.Tasks; 6. namespace \_5 7. { 8. class Program 9. { 10. static void Main(string[] args) 11. { 12. int M, N, i, j, k = 0; 13. Console.WriteLine("Введите размер матрицы M и N"); 14. M = int.Parse(Console.ReadLine()); 15. N = int.Parse(Console.ReadLine()); 16. int[,] A = new int[M, N]; 17. Console.WriteLine("Введите элементы"); 18. for (i = 0; i < M; i++) 19. { 20. for (j = 0; j < N; j++) 21. { 22. A[i, j] = int.Parse(Console.ReadLine()); 23. } 24. } 25. for (i = 0; i < M; i++) 26. { 27. for (j = 0; j < N; j++) 28. { 29. Console.Write(" " + A[i, j]); 30. } 31. Console.WriteLine(""); 32. } 33. for (j = 0; j < N; j++) 34. { 35. k = 0; 36. for (i = 0; i < M; i++) 37. { 38. if (A[i, j] % 2 == 0) 39. k++; 40. } 41. if (k == 0) 42. { 43. Console.WriteLine("Номер столбца, содержащий только нечетныечисла -{0} ", j + 1); 44. break; 45. } 46. } 47. if (k != 0) 48. { 49. Console.WriteLine("0"); 50. } 51. Console.ReadKey(); 52. } 53. } 54. } |

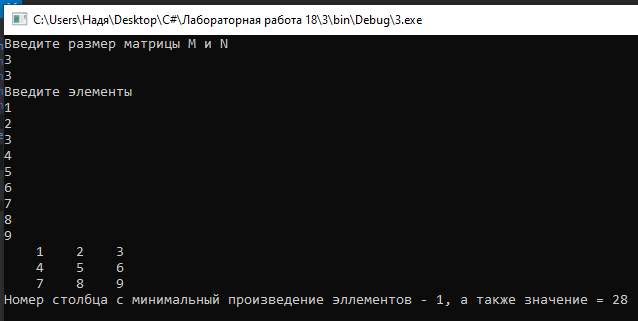
# Результат программы:



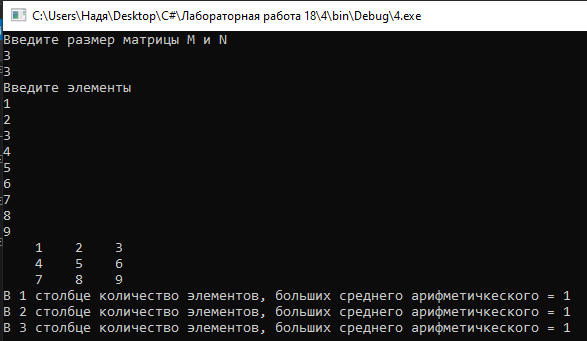
**Рис.1-Скриншот работы программы Задание №1**



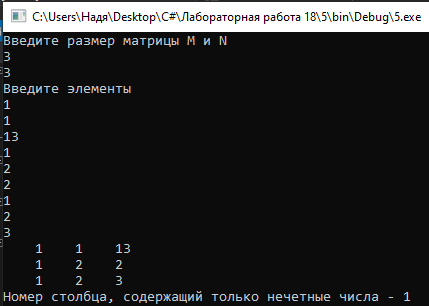
**Рис.2-Скриншот работы программы Задание №2**



**Рис.3-Скриншот работы программы Задание №3**



**Рис.4-Скриншот работы программы Задание №4**



**Рис.5-Скриншот работы программы Задание №5**