**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 № 17**

**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](#_Toc28117442)

[Задания: 3](#_Toc28117443)

[Код программы: 4](#_Toc28117444)

[Результат программы: 12](#_Toc28117445)

# 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. Дан целочисленный массив A размера N. Назовем серией группу подряд идущих одинаковых элементов, а длиной серии — количество этих элементов (длина серии может быть равна 1). Сформировать два новых целочисленных массива B и C одинакового размера, записав в массив B длины всех серий исходного массива, а в массив C — значения элементов, образующих эти серии

2. Дано целое число L (> 0) и целочисленный массив размера N. Заменить каждую серию массива, длина которой больше L, на один элемент с нулевым значением.

3. Дано целое число K (> 0) и целочисленный массив размера N. Поменять местами последнюю серию массива и его серию с номером K

4. Дано множество A из N точек (точки заданы своими координатами x, y). Среди всех точек этого множества, лежащих во второй четверти, найти точку, наиболее удаленную от начала координат. Если таких точек нет, то вывести точку с нулевыми координатами

5. Дано множество A из N точек (N > 2, точки заданы своими координатами x, y). Найти наибольший периметр треугольника, вершины которого принадлежат различным точкам множества A, и сами эти точки (точки выводятся в том же порядке, в котором они перечислены при задании множества A).

# 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. int N, x, y=0, z,b=0,c=0; 13. Console.WriteLine("Введите размер массива: "); 14. N = int.Parse(Console.ReadLine()); 15. int[] A = new int[N+1]; 16. int[] B = new int[N]; 17. int[] C = new int[N]; 18. Console.WriteLine("Введите элементы массива: "); 19. for(x=0;x<N;x++) 20. { 21. A[x] = int.Parse(Console.ReadLine()); 22. } 23. for(x=0;x<N;x++) 24. { 25. b = 1; 26. c=A[x] ; 27. while(A[x]==A[x+1]) 28. { 29. b++; 30. c = A[x]; 31. x++; 32. } 33. B[y] = b; 34. C[y] = c; 35. y++; 36. } 37. for(x=0;x<y;x++) 38. { 39. Console.WriteLine("Массив: {0},{1}", B[x], C[x]); 40. } 41. Console.ReadKey(); 42. } 43. } 44. } |

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 N, L, x, y , z, a ,b=0,c,k,n=0; 13. Console.WriteLine("Введите размер массива и число L: "); 14. N = int.Parse(Console.ReadLine()); 15. L = int.Parse(Console.ReadLine()); 16. int[] A = new int[N+2]; 17. A[N] = 0; 18. Console.WriteLine("Введите элементы массива: "); 19. for(x=0;x<N;x++) 20. { 21. A[x] = int.Parse(Console.ReadLine()); 22. } 23. for(x=0;x<N;x++) 24. { 25. y = x+1; 26. a = 0; 27. k = 1; 28. while ((A[x] == A[y])&& (A[x]!=0)) 29. { 30. y++; 31. k++; 32. a++; 33. } 34. if(k>L) 35. { 36. b += a; 37. for(c=y-1;c>=x;c--) 38. { 39. A[c] = 0; 40. } 41. while(n<k-1) 42. { 43. for(z=x+1;z<N+1;z++) 44. { 45. A[z] = A[z + 1]; 46. } 47. n++; 48. } 49. } 50. } 51. for(x=0;x<N-b;x++) 52. { 53. Console.WriteLine("Массив: " + A[x]); 54. } 55. Console.ReadKey(); 56. }}} |

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 \_16.\_3 7. { 8. class Program 9. { 10. static void Main(string[] args) 11. { 12. int K, N, C, B = 0, x, y = 0, a, k, l = 0, b = 0, c = 0, m =0, n = 0, r, S, M, R, f; 13. Console.WriteLine("Введите размер массива"); 14. N = int.Parse(Console.ReadLine()); 15. Console.WriteLine("Введите номер последовательности"); 16. K = int.Parse(Console.ReadLine()); 17. int[] Mas = new int[N + 2]; 18. int[] Mass = new int[N]; 19. int[] Masss = new int[N]; 20. Mas[N] = 0; 21. Console.WriteLine("Введите элементы массива"); 22. for (x = 0; x < N; x++) 23. { 24. Mas[x] = int.Parse(Console.ReadLine()); 25. } 26. for (x = 0; x < N; x++) 27. { 28. y = x + 1; 29. a = 0; 30. k = 1; 31. while (Mas[x] == Mas[y]) 32. { 33. y++; 34. a++; 35. } 36. if (a != 0) 37. l++; 38. if (l == K) 39. { 40. b = x; 41. c = y - 1; 42. break; 43. } 44. } 45. for (x = 0; x < N && y < N; x++) 46. { 47. y = x + 1; 48. while (Mas[x] == Mas[y]) 49. { 50. m = x; 51. n = y; 52. y++; 53. } 54. } 55. B = b; 56. C = c; | | | |
| 1. f = n - m; 2. { 3. if ((n - m) == (c - b) && (c - b) != 0) 4. { 5. for (; b <= c && m <= n; m++, b++) 6. { 7. k = Mas[b]; 8. Mas[b] = Mas[m]; 9. Mas[m] = k; 10. } 11. } 12. else if ((n - m) > (c - b) && (c - b) != 0) 13. { 14. r = (n - m) - (c - b); 15. M = m; 16. R = r; 17. for (S = 0; S <= (n - m); S++) 18. { 19. Mass[S] = Mas[M];Masss[S] = Mas[b]; 20. b++; 21. M++; 22. } 23. while (r != 0) 24. { 25. for (C = (m + R) + 1; C > c; C--) 26. { 27. Mas[C] = Mas[C - 1]; 28. } 29. r--; 30. } 31. for (S = 0; S <= (C - B); S++) 32. { 33. Mas[m + R] = Masss[S]; 34. m++; 35. } 36. for (S = 0; S <= C + 1; B++, S++) 37. { 38. Mas[B] = Mass[S]; 39. } 40. } 41. else if ((n - m) < (c - b) && (c - b) != 0) 42. { 43. r = (c - b) - (n - m); 44. M = m; 45. R = r; 46. for (S = 0; S <= (c - b); S++) 47. { 48. Mass[S] = Mas[M]; 49. Masss[S] = Mas[B]; 50. B++; 51. M++; 52. } 53. while (r != 0) 54. { 55. for (C = (b + R); C < m; C++) 56. { 57. Mas[C] = Mas[C + 1]; | | | |
| 1. Console.WriteLine("Mas C" + Mas[C]); 2. } 3. r--; 4. } 5. for (S = 0; S <= (c - b); S++) 6. { 7. Mas[m - R] = Masss[S]; 8. m++; 9. } 10. for (S = 0, B = b; S <= f; B++, S++) 11. { 12. Mas[B] = Mass[S]; 13. } 14. } 15. Console.WriteLine("Новый массив"); 16. for (x = 0; x < N; x++) 17. { 18. Console.WriteLine("" + Mas[x]); 19. } 20. Console.ReadKey(); 21. } 22. } 23. } 24. } |

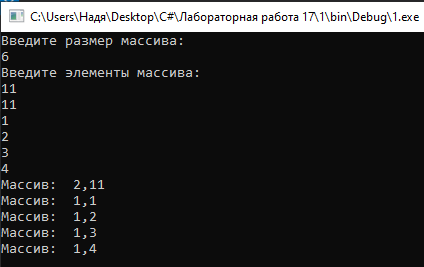
Листинг 4 – Задание 4

|  |
| --- |
| 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 N, x, y, z=0,max; 13. Console.WriteLine("Введите количество точек: "); 14. N = int.Parse(Console.ReadLine()); 15. int[,] A = new int[N, 2]; 16. Console.WriteLine("Введите координаты x и y: "); 17. for(x=0;x<N;x++) 18. { 19. for(y=0;y<2;y++) 20. { 21. A[x, y] = int.Parse(Console.ReadLine()); 22. } 23. } 24. max = A[0, 0] + A[0, 1]; 25. for(x=0;x<N;x++) 26. { 27. if(A[x,0]+A[x,1]>max) 28. { 29. max = A[x, 0] + A[x, 1]; 30. z = x; 31. } 32. } 33. Console.WriteLine("Координаты точки: {0} {1}", A[z, 0], A[z, 1]); 34. Console.ReadKey(); 35. } 36. } 37. } |

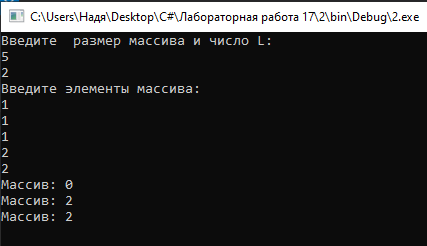
Листинг 5 – Задание 5

|  |
| --- |
| 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. Console.Write("Введите размер массива: "); 13. int n = Convert.ToInt32(Console.ReadLine()); 14. double[,] a = new double[n, 2]; 15. Console.WriteLine("Введите координаты точек: "); 16. for (int i = 0; i < n; i++) 17. { 18. Console.Write($"x{i}: "); 19. a[i, 0] = Convert.ToInt32(Console.ReadLine()); 20. Console.Write($"y{i}: "); 21. a[i, 1] = Convert.ToInt32(Console.ReadLine()); 22. } 23. int p1 = 0, p2 = 0, p3 = 0; 24. double p, pmax = 0; 25. for (int i = 0; i < n; i++) 26. for (int i2 = i + 1; i2 < n; i2++) 27. for (int i3 = i2 + 1; i3 < n; i3++) 28. { 29. p = 0; 30. p += Math.Sqrt(Math.Pow(a[i, 0] - a[i2, 0], 2) + Math.Pow(a[i, 1] -a[i2, 1], 2)); 31. p += Math.Sqrt(Math.Pow(a[i, 0] - a[i3, 0], 2) + Math.Pow(a[i, 1] - a[i3, 1], 2)); 32. p += Math.Sqrt(Math.Pow(a[i2, 0] - a[i3, 0], 2) + Math.Pow(a[i2, 1] -a[i3, 1], 2)); 33. if (p > pmax) 34. { 35. p1 = i; 36. p2 = i2; 37. p3 = i3; 38. pmax = p; 39. } 40. } 41. Console.WriteLine("Наибольший периметр треугольника: " + pmax); 42. Console.WriteLine("Координаты вершин:"); 43. Console.WriteLine(a[p1, 0] + ", " + a[p1, 1]); 44. Console.WriteLine(a[p2, 0] + ", " + a[p2, 1]); 45. Console.WriteLine(a[p3, 0] + ", " + a[p3, 1]); 46. Console.ReadLine(); 47. } 48. } 49. } |

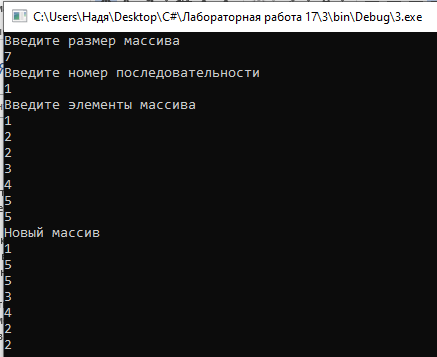
# Результат программы:



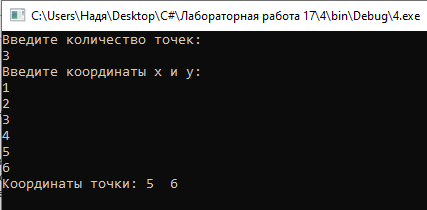
**Рис.1-Скриншот работы программы Задание №1**



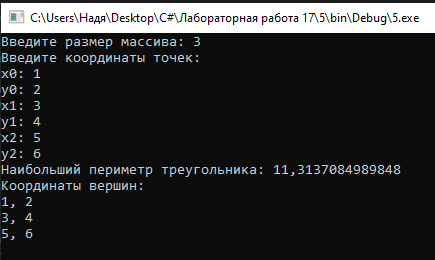
**Рис.2-Скриншот работы программы Задание №2**



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



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



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