***ПУСТОЙ ЛИСТ ДЛЯ ЗАГОТОВОК***

Содержание

[*Задача А 1/1* 3](#__RefHeading___Toc7428_1772474102)

[*Задача А 2/1* 5](#__RefHeading___Toc7428_1772474102_Copy_1)

[*Задача А 3/2 ИСПРАВЛЕНО* 7](#__RefHeading___Toc7428_1772474102_Copy_2)

[*Задача А 4/3* 9](#__RefHeading___Toc7428_1772474102_Copy_3)

[*Задача А 5/3* 11](#__RefHeading___Toc7428_1772474102_Copy_4)

[*Задача А 6/3* 13](#__RefHeading___Toc7428_1772474102_Copy_5)

[*Задача А 7/3* 15](#__RefHeading___Toc7428_1772474102_Copy_6)

[*Задача А 8/4* 17](#__RefHeading___Toc7428_1772474102_Copy_7)

# ***Задача А 1/1***

***Главная государственная налоговая инспекция страны Х определила следующую шкалу ставок месячного подоходного налога граждан: до 1000 у.е. – не облагается, более 1000-5000 у.е. - 10% суммы дохода, превышающего размер не облагаемого минимума, более 5000 до 10000 – 500 у.е. плюс 20% от суммы, превышающей 5000 у.е., свыше 10000 у.е. – 1000 у.е    плюс 30% от суммы, превышающей 10000 у.е. Требуется определить величину месячного подоходного налога по заданной сумме месячного дохода.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static double CalculateTax(double income)  {  if (income <= 1000) return 0;  if (income <= 5000) return (income - 1000) \* 0.1;  if (income <= 10000) return 500 + (income - 5000) \* 0.2;  return 1000 + (income - 10000) \* 0.3;  }  static void Main()  {  Console.Write("Введите доход: ");  string input = Console.ReadLine() ?? "";  try  {  double income = double.Parse(input);  if (income < 0) throw new ArgumentException("Отрицательный доход");  double tax = CalculateTax(income);  Console.WriteLine($"Налог: {tax} у.е.");  }  catch (FormatException) { Console.WriteLine("Ошибка: Некорректное число"); }  catch (ArgumentException ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |
|  | |

# ***Задача А 2/1***

***Дан массив вещественных чисел. Выяснить:***

***а) верно ли, что максимальный элемент превышает минимальный не более чем на 25;***

***б) верно ли, что минимальный элемент меньше мак­симального более чем в два раза?***

Решение

|  |  |
| --- | --- |
|  | using System;  class Program  {  static bool CheckDifference(double max, double min) => (max - min) <= 25;  static bool CheckRatio(double max, double min) => min != 0 && max / min > 2;  static void Main()  {  Console.Write("Количество чисел: ");  int n = int.Parse(Console.ReadLine() ?? "");  if (n <= 0) { Console.WriteLine("Положительное число!"); return; }  double min = 0, max = 0;  try  {  for (int i = 0; i < n; i++)  {  Console.Write($"Число {i + 1}: ");  double num = double.Parse(Console.ReadLine() ?? "");  if (i == 0) min = max = num;  else { if (num < min) min = num; if (num > max) max = num; }  }  Console.WriteLine($"Мин: {min}\nМакс: {max}");  Console.WriteLine($"Условие а): {CheckDifference(max, min)}");  Console.WriteLine($"Условие б): {CheckRatio(max, min)}");  }  catch (FormatException) { Console.WriteLine("Ошибка: Некорректное число"); }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |
|  | |

# ***Задача А 3/2 ИСПРАВЛЕНО***

***Найти все простые несократимые дроби, зак­люченные между 0 и 1, знаменатели которых не превышают 7 (дробь задается двумя натуральными числами - числителем и знаменателем)***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static int Gcd(int a, int b)  {  while (b != 0) { int temp = b; b = a % b; a = temp; }  return a;  }  static void Main()  {  Console.WriteLine("Несократимые дроби (0 < x < 1, знаменатель ≤ 7):");  for (int d = 2; d <= 7; d++)  for (int n = 1; n < d; n++)  if (Gcd(n, d) == 1)  Console.WriteLine($"{n}/{d}");  }  } |
|  | |

# ***Задача А 4/3***

***Дан массив x [1]..x[n] целых чисел. Не используя других массивов, переставить элементы массива в обратном порядке.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static void ReverseArray(int[] arr)  {  for (int i = 0; i < arr.Length / 2; i++)  {  int temp = arr[i];  arr[i] = arr[arr.Length - 1 - i];  arr[arr.Length - 1 - i] = temp;  }  }  static void Main()  {  Console.Write("Введите размер массива: ");  try  {  int n = int.Parse(Console.ReadLine() ?? "");  if (n <= 0) { Console.WriteLine("Размер должен быть положительным"); return; }  int[] arr = new int[n];  for (int i = 0; i < n; i++)  {  Console.Write($"Элемент {i + 1}: ");  arr[i] = int.Parse(Console.ReadLine() ?? "");  }  Console.WriteLine("Массив: [" + string.Join(", ", arr) + "]");  ReverseArray(arr);  Console.WriteLine("Обратный массив: [" + string.Join(", ", arr) + "]");  }  catch (FormatException) { Console.WriteLine("Ошибка: Некорректное число"); }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |
|  | |

# ***Задача А 5/3***

***Дан массив x [1]..x[n] целых чисел. Не используя других массивов, переставить элементы массива в обратном порядке.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static void FillArrayWithRandomHeights(int[] arr, int min, int max)  {  Random rand = new Random();  for (int i = 0; i < arr.Length; i++)  arr[i] = rand.Next(min, max + 1);  }  static void Main()  {  const int n = 12, minHeight = 163, maxHeight = 190;  int[] heights = new int[n];  FillArrayWithRandomHeights(heights, minHeight, maxHeight);  Console.WriteLine("Рост: [" + string.Join(", ", heights) + "]");  }  } |

|  |
| --- |
|  |

# ***Задача А 6/3***

***В массиве хранятся сведения о количестве осадков, выпавших за каждый день июня. Определить общее количество осадков, выпавших за каждую дека­ду этого месяца.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static int SumDecade(int[] arr, int start, int count)  {  int sum = 0;  for (int i = start; i < start + count; i++)  {  if (arr[i] < 0) throw new ArgumentException($"Отрицательное значение в день {i + 1}");  sum += arr[i];  }  return sum;  }  static void Main()  {  int[] rainfall = new int[30];  try  {  for (int i = 0; i < 30; i++)  {  Console.Write($"День {i + 1}: ");  string input = Console.ReadLine() ?? "";  if (!int.TryParse(input, out int value)) throw new FormatException("Введено не число. Пожалуйста, введите целое число.");  rainfall[i] = value;  }  Console.WriteLine($"1-я декада: {SumDecade(rainfall, 0, 10)} мм");  Console.WriteLine($"2-я декада: {SumDecade(rainfall, 10, 10)} мм");  Console.WriteLine($"3-я декада: {SumDecade(rainfall, 20, 10)} мм");  }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |

|  |
| --- |
|  |

# ***Задача А 7/3***

***Из элементов массива P сформировать мас­сив Q того же размера по правилу: элементы с номе­ром i с 3-го по 10-й находятся по формуле Qi= —Рi,, все остальные — по формуле Qi = Рi\*i.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static void FormArrayQ(int[] P, int[] Q)  {  if (P.Length != Q.Length) throw new ArgumentException("Одинаковый размер массивов");  for (int i = 0; i < P.Length; i++)  Q[i] = (i >= 2 && i <= 9) ? -P[i] : P[i] \* i;  }  static void Main()  {  Console.Write("Размер массива: ");  try  {  int n = int.Parse(Console.ReadLine() ?? "");  if (n <= 0) throw new ArgumentException("Положительный размер");  int[] P = new int[n], Q = new int[n];  for (int i = 0; i < n; i++)  {  Console.Write($"Элемент {i + 1}: ");  P[i] = int.Parse(Console.ReadLine() ?? "");  }  FormArrayQ(P, Q);  Console.WriteLine("P: [" + string.Join(", ", P) + "]");  Console.WriteLine("Q: [" + string.Join(", ", Q) + "]");  }  catch (FormatException) { Console.WriteLine("Ошибка: Некорректное число"); }  catch (ArgumentException ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |

|  |
| --- |
|  |

# ***Задача А 8/4***

***Дана квадратная таблица a[1:n,1:n]. Переставить местами строки 1 и 2, 3 и 4 и т.д. Если n нечетно, то последняя строка остается на месте.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static void SwapRows(int[,] a, int n)  {  if (n <= 0 || a.GetLength(0) != n || a.GetLength(1) != n)  throw new ArgumentException("Квадратная таблица с n > 0");  for (int i = 0; i < n - 1; i += 2)  for (int j = 0; j < n; j++)  {  int temp = a[i, j];  a[i, j] = a[i + 1, j];  a[i + 1, j] = temp;  }  }  static void Main()  {  Console.Write("Размер таблицы: ");  try  {  int n = int.Parse(Console.ReadLine() ?? "");  if (n <= 0) throw new ArgumentException("Положительный размер");  int[,] a = new int[n, n];  for (int i = 0; i < n; i++)  for (int j = 0; j < n; j++)  {  Console.Write($"Элемент [{i + 1},{j + 1}]: ");  a[i, j] = int.Parse(Console.ReadLine() ?? "");  }  Console.WriteLine("Исходная:");  for (int i = 0; i < n; i++)  {  int[] row = new int[n];  for (int j = 0; j < n; j++) row[j] = a[i, j];  Console.WriteLine("[" + string.Join(", ", row) + "]");  }  SwapRows(a, n);  Console.WriteLine("После обмена:");  for (int i = 0; i < n; i++)  {  int[] row = new int[n];  for (int j = 0; j < n; j++) row[j] = a[i, j];  Console.WriteLine("[" + string.Join(", ", row) + "]");  }  }  catch (FormatException) { Console.WriteLine("Ошибка: Некорректное число"); }  catch (ArgumentException ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |

|  |
| --- |
|  |