***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* 7](#__RefHeading___Toc7428_1772474102_Copy_2)

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

# ***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***

***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 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 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}"); }  }  } |
|  | |