Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт информационных технологий и анализа данных

Факультет Информационные системы и технологии

**Задача на For** – вариант 14

Отчёт по лабораторной работе № 1

по дисциплине Технология программирования

Выполнил студент ИСТб-20-3 М. Д. Ярков 15.09.2021

номер группы подпись И. О. Фамилия Дата

Принял З. А. Бахвалова . .2021

должность подпись И. О. Фамилия Дата

Иркутск 2021

### Постановка задачи

**Задача №14.** Дана последовательность натуральных чисел. Определить, есть ли в последовательности хотя бы одна n-ка одинаковых “соседних” чисел (n и элементы последовательности вводятся с клавиатуры). В случае положительного ответа определить порядковые номера чисел первой из таких пар.

### Словесное описание алгоритма

1. Создаём числовую переменную n для вводимого количества чисел, добавляем в неё полученное от пользователя значение. Число получаем с помощью метода Custom.ReadInt("количество натуральных чисел", 0):
   1. Методу были переданы значения переменным: string name = "количество натуральных чисел", int min = 0, int max = maxInt
   2. Создаём переменную для вывода ошибки при необходимости isInputGot, даём ей значение false – оно обозначает, что пользователь ещё ничего не вводил
   3. Создаём переменную для хранения введённого числа output, даём ей значение unused (константа, введённая для удобства в классе Custom) – оно обозначает, что пользователь ещё ничего не вводил
   4. Запускаем цикл do..while:
      1. Если в isInputGot уже true, выводим сообщение об ошибке
      2. Проверяем значения переменных min и max, которые были переданы методу на шаге 1.1. В зависимости от них выводим соответствующее сообщение, предлагающее пользователю ввести число
      3. Получаем введённую строку, помещаем её в новую переменную input
      4. Задаём переменной isInputGot значение true
      5. Если введённая строка содержит только число, добавляем его в output
   5. Выходим из цикла, если output изменён и входит в заданный промежуток значений
   6. Выводим output
2. Создаём целочисленный массив nums размерности n, заполняем в него с помощью цикла for и Custom.ReadInt("число " + (i + 1), 1) натуральные числа
3. Пишем в консоль значение, получаемое из GetTriplet(nums, n):
   1. Методу были переданы массив чисел nums, импортированный как unsorted и их количество n, импортированное как len
   2. Инициализируем массив sorted длины len, копируем в него всё из массива unsorted и сортируем по возрастанию
   3. Инициализируем числовые переменные v1, v2 и v3 для хранения последних трёх различных чисел
   4. Запускаем цикл, который сработает len раз для перебора чисел в отсортированном массиве sorted:
      1. Проверяем, является ли новое число отличающимся от предыдущего. Если не является, переходим к следующей итерации цикла, а если является, то продолжаем:
      2. Добавляем в переменную v1 значение из v2, в переменную v2 значение из v3, а в v3 – новое число из массива
      3. Если все переменные, хранящие последние числа v1, v2 и v3 заполнены, то проверяется их соседство с помощью метода IsBetween(v1, v2, v3):
         1. В IsBetween передаются переменные v1, v2, v3 как first, second, third
         2. Возвращается истина, если first равен (second – 1) и second равен (third - 1), иначе ложь
      4. При возврате истины продолжается алгоритм: в переменную f1 добавляется порядковый номер (из несортированного списка) первого числа в тройке, аналогично в f2 добавляется порядковый номер второго числа, а в f3 – третьего
      5. Выводим номера чисел в сформированной строке
   5. Если вывода не произошло до этого шага (тройка не найдена), выводим строку о том, что тройка не найдена
4. Вызываем Custom.Exit() для ожидания закрытия программы пользователем:
   1. Пишем в консоль «Для выхода из программы нажмите любую клавишу…»
   2. Ждём нажатия клавиши

### Таблица спецификаций

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| Входные величины | | |
| n | Количество чисел | Натуральное число |
| nums[n] | Вводимые числа | Массив натуральных чисел |

### Таблица тестов

| Номер теста | Что проверяем | Входные данные | Выходные данные |
| --- | --- | --- | --- |
| 1 | Проверка несоседства | 6, 3, 4 | false |
| 2 | Проверка соседства | 4, 5, 6 | true |
| 3 | Нахождение тройки чисел | { 6, 3, 5, 6, 7 }, 5 | "Тройка чисел найдена, их порядковые номера 3, 1 и 5" |
| 4 | Ненахождение тройки чисел | { 1, 2 }, 2 | "Тройка чисел не найдена" |

### Код программы

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Lab1Num2Var14 {  public class Program {  public static bool IsBetween(int first, int second, int third) { // вывод, стоят ли числа рядом  return (first == second - 1 && second == third - 1);  }  public static string GetTriplet(int[] unsorted, int len) { // вывод строки с инф. о  // сортируем массив  int[] sorted = new int[len];  Array.Copy(unsorted, 0, sorted, 0, len);  Array.Sort(sorted);  // инициализируем переменные  int v1 = Custom.unused, v2 = Custom.unused, v3 = Custom.unused;  // добавляем в переменные данные и обрабатываем их  for (int i = 0; i < len; i++) {  if (v3 != sorted[i]) { // проверка на разницу последних чисел  // при разнице смещаем значения влево и добавляем в v3 новое число  v1 = v2; v2 = v3; v3 = sorted[i];  if (v1 != Custom.unused) { // в случае, если все переменные заполнены  if (IsBetween(v1, v2, v3)) { // проверяем их на соседство  // получаем порядковые номера найденых соседей  int f1 = Array.IndexOf(unsorted, v1) + 1;  int f2 = Array.IndexOf(unsorted, v2) + 1;  int f3 = Array.IndexOf(unsorted, v3) + 1;  // вывод первого вхождения при нахождении  return ("Тройка чисел найдена, их порядковые номера " + f1 + ", " + f2 + " и " + f3);  }  }  }  }  // вывод исключения при ненаходе  return "Тройка чисел не найдена";  }  static void Main(string[] args) {  int n = Custom.ReadInt("количество натуральных чисел", 0);  int[] nums = new int[n];  for (int i = 0; i < n; i++) {  nums[i] = Custom.ReadInt("число " + (i + 1), 1);  }  Console.WriteLine(GetTriplet(nums, n));  Custom.Exit();  }  }  public class Custom {  public const int unused = Int32.MinValue; // константа для неиспользованных переменных  public const int minInt = Int32.MinValue; // константа минимального числа для типа Int  public const int maxInt = Int32.MaxValue; // константа максмального числа для типа Int  public static int ReadInt(string name, int min = minInt, int max = maxInt) {  bool isInputGot = false; // переменная для вывода ошибки при необходимости  int output = unused; // тут будет храниться введённое число  do {  if (isInputGot) Console.Write("Ошибка ввода! "); // вывод ошибки при повторном вводе  if (min == minInt && max == maxInt) // ввод без ОДЗ  Console.Write("Введите " + name + ": ");  else if (min != minInt && max != maxInt) // ввод с ОДЗ  Console.Write("Введите " + name + " (в промежутке от " + min + " до " + max + "): ");  else if (min == minInt) // ввод числа до заданного  Console.Write("Введите " + name + " (до " + max + "): ");  else if (max == maxInt) // ввод числа от заданного  Console.Write("Введите " + name + " (от " + min + "): ");  string input = Console.ReadLine(); // ввод некоторой строки  isInputGot = true; // изменение переменной для вывода ошибки при необходимости  // перевод строки input в числовой output, если строка является числом  if (int.TryParse(input, out int trash)) output = int.Parse(input);  }  while (output == unused || output < min || output > max);  return output;  }  public static void Exit() { // ожидание закрытия консоли пользователем  Console.Write("\nДля выхода из программы нажмите любую клавишу...");  Console.ReadKey();  }  }  } |

## Группа B

…