# Задания по исключениям

## Число с плавающей точкой

Напишите консольное приложение которое предлагает пользователю ввести число с плавающей точкой. Если введено не число или слишком большое/маленькое число, то программа выводит соответствующее сообщение об ошибке и предлагает повторить ввод. Если введено число, то выводим его значение на экран с округлением до двух знаков после запятой и завершаем программу.

Реализуйте программу используя цикл while (или do while) и цикл for (;;).

* В зависимости от языка ОС в качестве разделителя нужно использовать или запятую (русская ОС) или точку (английская ОС)
* Для проверки ввода слишком маленького или слишком большого числа введите строки (без кавычек) – «-1e600» и «1e600».

## Проверка номера банковской карты

Напишите static класс с функцией для проверки номера банковской карты. Функция должна принимать номер банковской карты через параметр строкового типа и возвращать булевское значение. Для проверки номера реализуйте [алгоритм Луна](http://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%9B%D1%83%D0%BD%D0%B0).

Если аргумент функции равен null, то должно генерироваться исключение ArgumentNullException.

Если в аргументе функции содержаться символы отличные от цифр, то должно генерироваться исключение FormatException.

Примеры правильных номеров:

* 4444444444444448
* 4055011111111111
* 5555555555555557
* 5454545454545454
* 5555515555555551

Примеры неправильных номеров:

* 79927398711
* 79927398712
* 79927398715
* 79927398716

# Задания по System.IO

## Больше-меньше

В двух строчках входного файла INPUT.TXT записаны числа A и B, не превосходящие по абсолютной величине 2\*109. Запишите в выходной файл OUTPUT.TXT один символ “<”, если A < B, “>”, если A > B и “=”, если A=B.

### Примеры

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 5  7 | < |
| 2 | -7  -12 | > |
| 3 | 13  13 | = |

## Определение формата

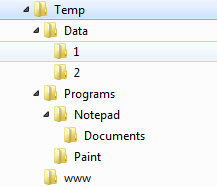
В заданной папке находятся файлы с расширением bin. Вам требуется «угадать» формат файла и дать ему правильное расширение. Формат файла можно узнать проанализировав начальные байты. То есть если в начале файла найдена одна из последовательностей байтов в таблице ниже, то файл имеет соответствующий формат, и его требуется переименовать. Если формат файла определить не удалось, то переименовывать его не нужно. Постарайтесь написать программу таким образом чтобы можно было легко добавить новые форматы.

|  |  |
| --- | --- |
| Формат | «Магическая» последовательность |
| PNG | 0x89, 0x50, 0x4E, 0x47, 0x0D, 0x0A, 0x1A, 0x0A, 0x00, 0x00, 0x00, 0x0D, 0x49, 0x48, 0x44, 0x52 |
| JPG | 0xFF, 0xD8, 0xFF |
| GIF | 0x47, 0x49, 0x46, 0x38 |
| 0x47, 0x49, 0x46, 0x38, 0x37, 0x61 |
| 0x47, 0x49, 0x46, 0x38, 0x39, 0x61 |

Тестовые файлы находятся в архиве guess-extension.zip рядом с заданием.

## Дерево каталогов

Напишите программу которая просит пользователя ввести путь к каталогу и выводит в текстовый файл имена всех каталогов включая вложенные в виде «дерева». Глубину вложенности каталога отмечаем с помощью пробелов или табуляции (\t). Например, у нас есть каталог Temp с такой иерархией:



Тогда выходной файл должен выглядеть так:

Data  
 1  
 2  
Programs  
 Notepad  
 Documents  
 Paint  
www

Реализуйте эту программу двумя способами:

1. С помощью рекурсии
2. Без рекурсии используя перегруженный вариант функции [Directory.GetDirectories()](http://msdn.microsoft.com/en-us/library/ms143314%28v=vs.110%29.aspx) принимающий аргумент c типом SearchOption.

Проверьте как ваша программа справляется с каталогом Windows. Там некоторые каталоги недоступны без административных прав и функция GetDirectories() будет генерировать исключение.