**Programming Fundamentals   
and Unit Testing Regular Exam - 19 October 2023**

## **English Name of Each Digit**

Write a program that:

* Reads an **integer positive number** from the console
* Print **English name** of the each digit of a given number on a separate line, **starting from the last to the first digit**
  + 1 -> "one"
  + 2 -> "two"
  + 3 -> "three"
  + 4 -> "four"
  + 5 -> "five"
  + 6 -> "six"
  + 7 -> "seven"
  + 8 -> "eight"
  + 9 -> "nine"

### **Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 512 | two  one  five |
| 121 | one  two  one |
| 1643 | three  four  six  one |

## **Average Last Elements**

Write a program that:

* Reads an **array of integer numbers** from the first line of the console, separated by single space
* Read an **integer number N** from the second line of the console
* Find **average value** of the **last N elements** in the array
* Print the **average value** formatted to the **second decimal digit**

### **Example**

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 3 42 61 7 8 9 10 23  4 | 12.50 | Last 4 numbers in the array are: 8 9 10 23  Average value is: (8 + 9 + 10 + 23) / 4 = 12.5 |
| 12 34 98 42 65 12  3 | 39.67 | Last 3 numbers in the array are: 42 65 12  Average value is: (42 + 65 + 12) / 3 = 39.67 |

## **Unit Test Method: Extract File**

Test a given method which takes in **a path in the form of a string** and gets the **filename** and **file extension** (if it exists).

### **Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| C:\Users\John\Documents\example.txt | File name: example\nFile extension: txt |
| C:\Users\John\Documents\example | File name: example |
| C:\Users\John\Documents?\example.txt | File name: example\nFile extension: txt |
| null | Throws ArgumentNullException |
| (empty string) | Throws ArgumentNullException |

The method is found in the ExtractFile.cs file:

A computer screen shot of text

Description automatically generated

You are given a **test** **file** ExtractFileTests.cs containing **5 empty tests**. Implement all the unit tests:

A screenshot of a computer

Description automatically generated

When you are ready make sure your **tests run:**

A screenshot of a computer

Description automatically generated

**IMPORTANT:** **DO NOT REMOVE OR CHANGE ANY NAMESPACES AND USINGS.**

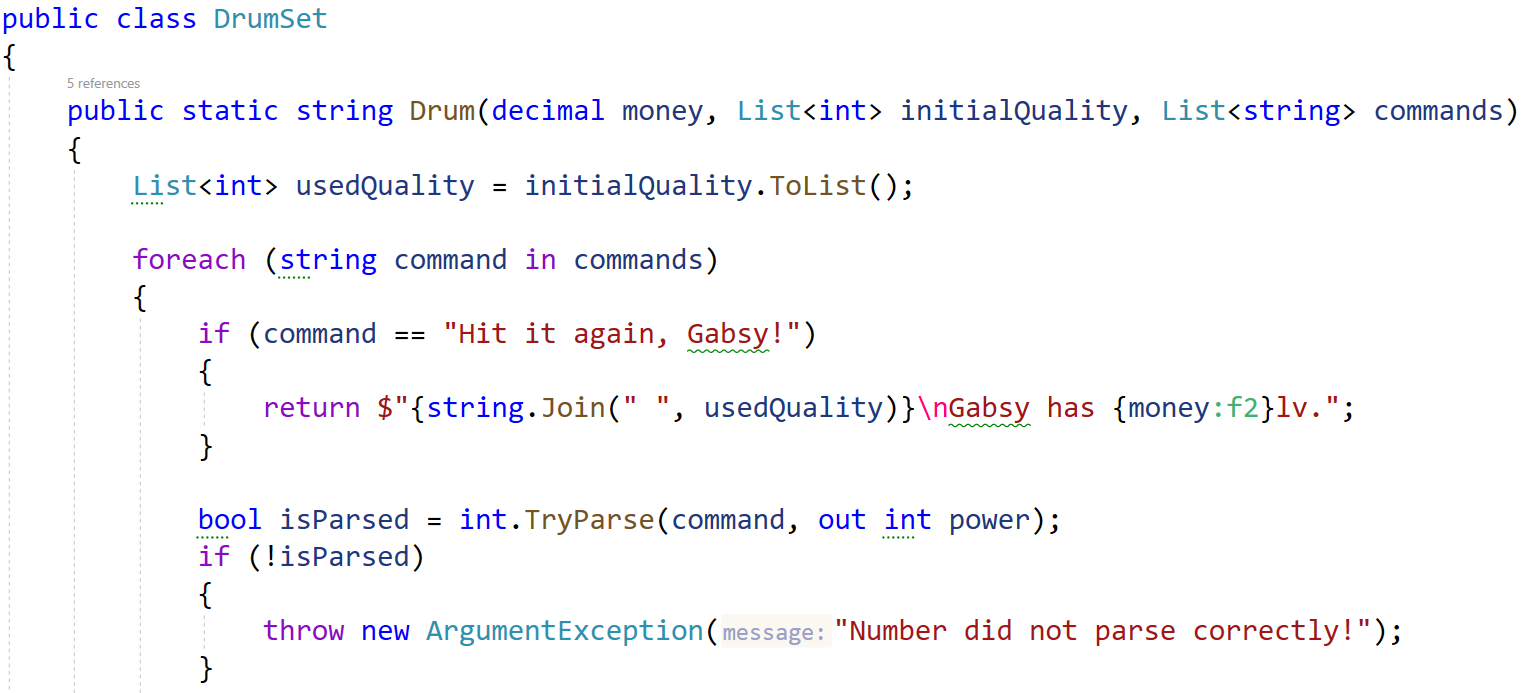
## **Unit Test List: Drum Set**

Test a given method which takes in **a decimal representing money, list of integers representing quality of drums and a list of strings representing commands**.

For each command **each** **drum is hit by the number indicated**. If a drum breaks (**goes below or equal to zero**) it must be replaced by the **initial amount,** then **initial amount times three** must be reduced from the money. If a new drum can’t be afforded it is removed from the list.

When the command **"Hit it again, Gabsy!"** is received, the method ends, and returns the current **state** of the **drum** **set** alongside the **leftover money**.

The method is found in the DrumSet.cs file:



A screenshot of a computer program

Description automatically generated

You are given a **test** **file** DrumSetTests.cs containing **5 empty tests**. Implement all the unit tests:

A screenshot of a computer program

Description automatically generated

When you are ready make sure your **tests run:**

A screenshot of a computer

Description automatically generated

**IMPORTANT:** **DO NOT REMOVE OR CHANGE ANY NAMESPACES AND USINGS.**