Problem 122: Divide By Zero

Difficulty: Medium

Author: Danny Lin, Greenville, South Carolina, United States

Originally Published: Code Quest Australia 2019

Problem Background

As much as we enjoy hosting this website for you, we have to admit there is a flaw with how the problems work. For all of our problems, we provide you with a list of inputs that are guaranteed to be valid; all of the inputs fit within the constraints we define, and there's no need to write code to handle invalid input. In reality, however, one should almost never assume that the inputs are valid.

When writing a software application, the user should never be trusted. Users typically don't act maliciously - although that is still something to guard against - but simple mistakes or accidents can still cripple an application that doesn't guard against them. Even seemingly valid inputs can cause issues if combined in an unexpected way. Good software development should always validate and sanitize inputs, and be prepared to handle any errors that may occur.

Problem Description

Your task for this problem is to write a program that will read in two numbers and print their quotient; the result obtained when dividing the first number by the second number. Unlike our other problems, however, the input will not always be valid. You will need to check the provided inputs and ensure that they are usable; alternatively, you can attempt the division and simply catch any errors that are thrown by the program (if taking this approach, be careful that your program doesn't terminate earlier than expected). If any invalid inputs or errors are detected, your program will need to print an appropriate error message as follows.

If...	...Your Program Should Print...
The dividend (first number) is not a number	Invalid Dividend
The divisor (second number) is not a number	Invalid Divisor
The divisor (second number) is zero (0)	Divide By Zero

In the event multiple error conditions are present in a single test case, your program should only print one error message, selecting the first applicable error message from the list above.

Sample Input

The first line of your program's input, received from the standard input channel, will contain a positive integer representing the number of test cases. Each test case will include a line containing two input values separated by a space. The first value is the dividend; the second is the divisor. As noted previously, these are *not* guaranteed to be numeric values, and can contain any characters other than spaces. (We do guarantee that you will receive exactly two values.)

```
4
5 2
Five 2.0
5 Two
5.0 0
```

Sample Output

For each test case, your program must print the result obtained when dividing the dividend by the divisor, rounded to one decimal place. If an error condition exists that prevents the division from taking place, your program should print the first relevant error message from the list above.

```
2.5
Invalid Dividend
Invalid Divisor
Divide By Zero
```