Lab 6

Purpose: This week we will be focusing on the while loop. We will be practicing the end of input file as we discussed in class.

Details

We will be redoing last week's lab with the addition of having more than one piece of input. We will also change how the output is done to make it easier to read.

This program will compute basic integer math. We will use five commands, add, subtract, multiply, divide and remainder. Your program will read this word into a string, then perform the correct calcuation on the following two integers. For example, if you read the word "add", then you should add the following integers and print the result. The rest of the operations are performed as you would expect, "subtract" you use subtraction, "multiply" you multiply, etc.

Requirements

- Input file: Your program must use: **problem.txt**
- Output file: Your program must use: **solution.txt**
- Do not use any `cin` statements.
- Do not use `system("pause");`

Format for the input: The input will have the following form:
`<operation>tab<integer>tab<integer>`
Where `<operation>` will be one of the following five commands: add, subtract, multiply, divide or remainder
`<integer>` will be an integer number.
There will only be an unknown number of commands and two integers per line in the file.

Sample:

```
add 1 1
subtract 1 1
multiply 1 1
```

```
divide 1 1
remainder 1 1
```

Output Sample

For this will we make a table so we can condense our output. So the corrosponding output file to the sample from above would look like this:

```
#1 Operation #2 Answer
1      +      1 2
1      -      1 0
1      *      1 1
1      /      1 1
1      %      1 0
```

Again, your output file will only have one of these outputs depending on which input you get. Notice that we are using the symbols to represent the command you were given. You do not need to worry about the spaces between the numbers and signs as long as there is at least one space. The header line will have to match as it is written.

In order to complete this lab you will need to use a loop that terminates when it finishes. We discussed this loop the other day in class. Please refer to your notes on how to do this. If you don't know how, you will need to work with someone who does.

I will attach a Zip file containing samples for each type of problem and output.

Submission:

We will turn this into the Curator and I will make sure that I give you 10 submissions.

Please make sample input files before you submit your work. You can do this in Visual Studio by adding a txt file to your project.

You do this from the Project Menu and Choose Add New Item. Look for Text File in the list and use the appropriate name, i.e. problem.txt. Then you can put a problem in there for testing.