CS061 – Programming Assignment 05

Objective

The purpose of this assignment is to challenge you with a program that assumes total comfort and confidence with LC3.
You will also be presented with an algorithm challenge, and a minor logical puzzle.

High Level Description

Multiply two decimal numbers, either of which could be negative, and output the result to the console. The range of acceptable multiplicands is [#-32767, #+32767].
If you are really smart, you will also be able to handle #-32768 (x1000).
If the result overflows the bounds of a 16-bit 2's complement number, report an error message and quit.

Examples

If the user enters "+1234" and "+18", your program should compute the product of the two numbers (#22212) and print something like

```
+1234 * +18 = +22212
```

to the console.

If the user enters "+10025" and "-3", your program should compute the product of the two numbers (#-30075) and print

```
+10025 * -3 = -30075
```

If the user enters "+12345" and "+0", your program should compute the product of the two numbers (#0) and print

```
+12345 * +0 = +0
```

If the user enters "+550" and "+123", your program should try to compute the product of the two numbers (#67650 would overflow) and print something like

```
Woes! Overflows!
```

If the user enters "-98" and "+876", your program should try to compute the product of the two numbers (#-85848 would *really really* underflow) and print something like

```
Woahs! Underflows!
```

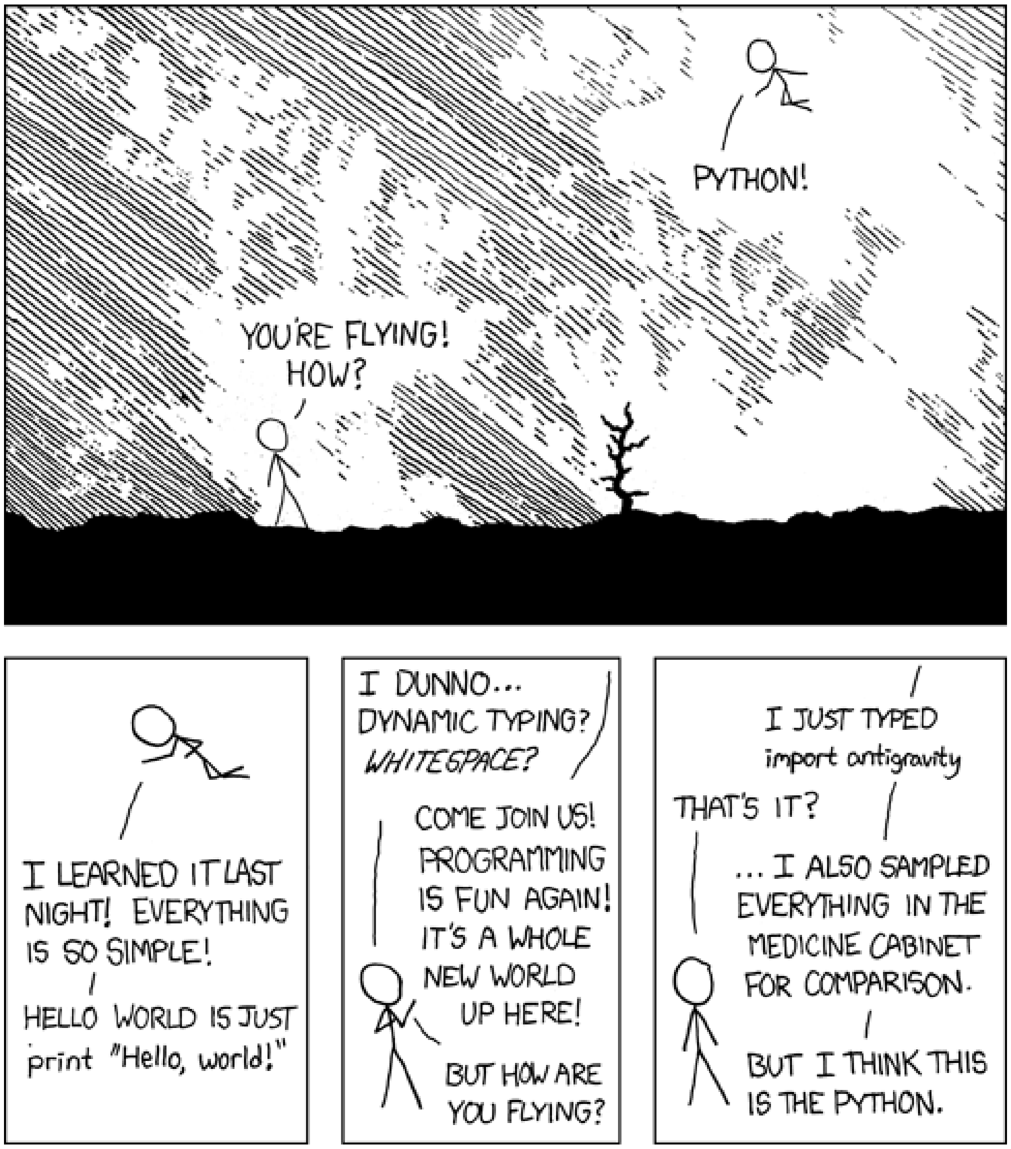
Your Tasks

This programming assignment can be broken into four steps:

1. Read two decimals from the keyboard, either of which can be negative.
   As usual, input of a number should be terminated with ENTER.
2. Convert the strings into the numbers they represent (i.e. "-1234" becomes #-1234) – exactly what you did for assignment 4
3. Multiply the two numbers together (remember lab 01?)
   Note: this can be a catastrophically inefficient algorithm – think about the two different ways you could go about multiplying 2 by 10,000.
   Can you think of any ways to improve on it?
4. If the result overflows or underflows the LC3 16-bit 2's complement number range, output an appropriate error message. Otherwise, print the answer.

Uh…help?	● It helps to read Chapter 10 of the book ● If you don't have one already, write a ***subroutine*** that gets a string from the user and converts it to a number (lab5 + assn4) ● Handle the sign (+/-) of the number and the magnitude (the number part) separately. That is, if both numbers are positive or both numbers are negative, you know the result should be positive. Else, the result should be negative. ● If you expect the result to be positive and it ***ever*** becomes negative (and vice versa), what does that mean?
Rubric	● Code does not assemble: -10 points *(no reshow)* ● No header: -10 points *(no reshow)* ● Well commented code: +2 points ● Numbers properly read from user: +2 points ● Proper overflow and underflow error messages: +2 points ● Correct output of product: +4 points

Comics?!Sweet!!


PYTHON!
YOU'RE FLYING! HOW?
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

Source: http://xkcd.com/353/