

Programming Project 03

Assignment Overview

This assignment is worth 30 points (3.0% of the course grade) and must be completed and turned in before 11:59 PM on Monday, January 28th. The aim of this project is to familiarize you with the use of Boolean logic, branching statements and loops.

Background

For this assignment, you will be implementing the so-called “Russian Peasant” or “Ancient Egyptian” method for multiplication. It looks a little odd, but just think of it as an algorithm, a recipe for doing multiplication in a way other than what you learned in grade school.

For a detailed description of how the algorithm works, please visit the links :

- http://en.wikipedia.org/wiki/Ancient_Egyptian_multiplication
- <http://www.cut-the-knot.org/Curriculum/Algebra/PeasantMultiplication.shtml>

The algorithm is as follows. If A and B are the 2 integers (only integers) to be multiplied, we repeatedly multiply A by 2 and divide B by 2, until B can't be divided any further, that is until its value becomes 0 (remember, this is integer division). During each step, whenever B is an odd number, we add the corresponding A value to the product we are generating. In the end, the sum of the A values that had corresponding odd B values is the product. Get it?

Here's an example:

If the two integers to be multiplied are 34 and 19, the operations would be:

A	B	Comment
34	19	add A to the product, B is odd
68	9	add A to the product, B is odd
136	4	ignore this A value, B is even
272	2	ignore this A value, B is even
544	1	add A to the product, B is odd

Sum up all the A values that had odd B values and you get:

$$34 + 68 + 544 = 646 \Rightarrow \text{Final Product}$$

Specifications

1. The value entered by the user should be two integers separated by a space. Only integers are allowed and no error checking is required.
2. In calculating the product, show how the algorithm proceeds. See the example output at the end of this document. This ensures that you are using the required algorithm.
3. After the product is calculated, please indicate the sign of the product (positive, negative or zero)

4. After the result is printed, your program will ask if you wish to perform another calculation. If 'y' or 'Y' is entered, another calculation can be performed. If 'n' or 'N' is entered the program will quit and print a nice message. Any other input will print an error note and end the program.
5. The program should give correct results for all integers i.e. positive as well as negative numbers.
6. Constructs that ***must be*** in the final program include:
 - a. `raw_input` to prompt the user
 - b. `print` to print results
 - c. at least one branching mechanism (`if` statement)
 - d. at least one loop (`while` loop)
 - e. Boolean logic

Deliverables

Use handin to turn in a file called **proj03.py** – this is your source code solution; be sure to include your section, the date, the project number and comments describing your code. Please be sure to use the specified file name, and save a copy of your proj03.py file to your H drive as a backup.

Assignment Notes

1. `raw_input` should be used for prompting. Remember it returns a “**string**” containing the user’s choice.
2. ***Very important***: In Python integer division, $1/2 \rightarrow 0$ but $-1/2 \rightarrow -1$
3. The string method `lower()` may be useful to make the choice input lower case easier to check; `'ABC'.lower() => 'abc'`. Its use is optional, however.
4. In the project directory is a Python program that demonstrates a `while` loop handling input. We suggest that you copy this file and use it as the starting point for your program. Run the program and play with input until you are confident that you understand what it is doing.

Getting Started

1. Do all the standard startup things. Create a new file called proj03.py. Put your comments in at the top, save it.
2. ***Most important!*** Start with just positive numbers! Worry about the negative numbers after you get the algorithm working.
3. Now you need to break the problem down into parts. Read the description and identify the subtasks that need to be solved. In this project we have started that process by providing a starting program. Using comments mark where you will add subtasks to the file.
4. A logical subtask would be to do the actual calculations. Assuming that all input is perfectly correct, draw up the code to take the input and perform the operation.

Please input the 2 numbers separated by a space: 22 33

A = 22 and B = 33

B was odd, we add A to make the product: 22

A = 44 and B = 16

A = 88 and B = 8

A = 176 and B = 4
A = 352 and B = 2
A = 704 and B = 1
B was odd, we add A to make the product: 726
Product is positive
The product of the two numbers is: 726
Do you want to continue?(y/n)y
=====

Please input the 2 numbers separated by a space:-22 33
A = -22 and B = 33
B was odd, we add A to make the product: -22
A = -44 and B = 16
A = -88 and B = 8
A = -176 and B = 4
A = -352 and B = 2
A = -704 and B = 1
B was odd, we add A to make the product: -726
Product is negative
The product of the two numbers is: -726
Do you want to continue?(y/n)Y
=====

Please input the 2 numbers separated by a space:-22 -33
A = -22 and B = 33
B was odd, we add A to make the product: -22
A = -44 and B = 16
A = -88 and B = 8
A = -176 and B = 4
A = -352 and B = 2
A = -704 and B = 1
B was odd, we add A to make the product: -726
Product is positive
The product of the two numbers is: 726
Do you want to continue?(y/n)YES
Bad input, quitting
>>>