Booth’s Algorithm Option:
The goal of this assignment is to provide you with experience in using the "Logical and Shift Instructions". For this assignment, you are to implement Booth's multiplication algorithm on two 32-bit signed integers and get a 64-bit integer result.

I want you to write a function called "Multiply" that is passed two signed integers (in $a0 and $a1) and returns their 64-bit product across registers $v0 and $v1. Register $v0 should return the most-significant 32-bits of the product and $v1 should contain the least-significant 32-bits of the product. Be sure that you DO NOT use any form of the multiply (i.e., MUL) assembly language instruction, and be sure to follow the MIPS register conventions when implementing "Multiply."

Your "main" program should:
1) Read two 32-bit integer variables MULTIPLICAND and MULTIPLIER from memory. Initialize the value -30 for the MULTIPLICAND and 983 for the MULTIPLIER.
2) call your Multiply function with these values
3) Store the 64-bit product back to memory in a variable called PRODUCT

The .data section of your program should look something like:

```
data
MULTIPLICAND: .word -30
MULTIPLIER: .word 983
PRODUCT: .word 0,0
```

You should submit your homework via the Internet by following the directions at: http://www.cs.uni.edu/~fienup/cs1410s16/homework/submissionDirections.htm
Basically, you put the following files in a hw9EC folder and zip the folder to create a hw9EC.zip file containing:
• the MIPS assembly language program, e.g., hw9.s from any text-editor (e.g., WordPad)
a window capture of the QtSpim simulator after running your assembly language program with values above. Make sure the values are visible in the data section of the screen capture.