

CS2810 Homework 7 - 60 points - Due by midnight October 28. Submit your program (source code .asm file) as a separate file through Eagle.

For this homework assignment you may work individually or in a group of up to 3 students. If working in a group, be sure that when you turn in your homework through Eagle you include the names of everyone in the group.

Write a MARS compatible MIPS program to first read in an operation integer (0 or 1) followed by two additional integers (positive or negative). You may assume that all three input integers are always representable integers. Your program should execute as follows:

- If, for the operation integer a value other than 0, or 1 is read in, output the message “ERROR, illegal input = <the value input>”, and then ask for another input.
- If the operation integer is 0, the program should output “GOODBYE” and terminate execution.
- If the operation integer is 1, the program should output the product of the two integers. If this product generates an overflow/underflow, i.e. requires more than 32 bits, output the message “ERROR <first integer> times <second integer> too large/small”. If the product does not generate an overflow or underflow, output the message “<first integer> times <second integer> = <the product>”. Then, ask for the next input.

IMPORTANT NOTE: In order to make this program “interesting”, there is one restriction. You **CANNOT** use MIPS or pseudo-MIPS multiplication instructions. This means you will have to use a counted loop of additions to perform multiplication.

Sample program execution: (user input underlined)

Input control integer (0-halt, or 1-multiply): -2
ERROR, illegal input = -2

Input control integer (0-halt, or 1-multiply): 1
Input first integer: 10
Input second integer: 20
10 times 20 = 200

Input control integer (0-halt, or 1-multiply): 1
Input first integer: -10
Input second integer: 20
-10 times 20 = -200

Input control integer (0-halt, or 1-multiply): 1
Input first integer: -10
Input second integer: -20
-10 times -20 = 200

Input control integer (0-halt, or 1-multiply): 1
Input first integer: -1000000
Input second integer: -2000000
ERROR -1000000 times -2000000 too large

Input control integer (0-halt, or 1-multiply): 1
Input first integer: -1000000
Input second integer: 2000000
ERROR -1000000 times -2000000 too small

Input control integer (0-halt, 1-multiply, or 2-divide): 0
GOODBYE

NOTES:

- At first glance, this program may seem fairly simple. It is not simple, so start early.
- I suggest that in writing this program, you do it in steps or stages. Do not try to write it all at once.
- As has been said in class, “The challenge of computer science is the exception”. In this case, carefully consider what the exceptions are (overflow and underflow), and how to detect them.

As examples, let's say we have only a 3-bit 2's complement integer representation scheme. Thus, the possible decimal values are 0,1,2,3,-1,-2,-3,-4.

- $2*3$ will generate an overflow
- $2*(-3)$ will generate an underflow
- $2*(-2)$ will not generate an overflow
- $(-2)*(-2)$ will generate an overflow
- Tricky isn't it!