Your assignment is to create a basic calculator program, called calculator.asm.

Your program should take 3 separate inputs:

1. an integer
2. an operation (A, S, M, or D)
3. another integer

If the operation requested is somehow invalid, it will print a friendly error message and exit. It should also do this if the operation would result in an overflow or divide-by-0 error. (Note that you have to do your own detection. You can’t just use the built-in overflow error message that the assembler/simulator already has.)

Otherwise, print the answer, assuming a signed operation. In the case of a division operation, you must print out both the quotient and the remainder.

A special note about multiplication overflow: For this project, we’re considering the answer to have overflowed if it is greater than 32 bits. These 32 bits include the sign bit. So you can’t just make sure that hi holds 0 or −1. You also have to make sure that whichever value is there matches with the sign bit in lo. Don’t use the mul instruction—it can’t detect overflow.

Also, remember that there exists one case in which division will overflow.

Hint: it will likely be to your advantage to assign each math operation its own function. You don’t need to worry about the stack, but placing the integer operands into $a0 and $a1 and performing a jal is a good idea.