

# CS 240 Project 2

**Project is due 4/7/2022 5 pm**

The code on pages 20-21 of the Text book asks you to input a number, and it will then sum the numbers from 1 to that number. The Prompt asks you to input a number, not necessarily an integer. The program will abort if a floating point number is entered.

Your project, is to “fix” the program and allow for a floating point number to be entered. The program will NOT run, so your task is to convert the floating point number to an integer.

If a floating point number is entered, truncate the number and use that integer to run the program. Also, you must inform the user that they entered a floating point and it was truncated, output to the user, the floating point they entered and the integer you used.

**Note: You must only use Shift, Rotate to manipulate the bits, no conversion instructions.**

The following are some pointers as to what needs to be done.

- 1) The program reads an integer, that must be changed to read a floating point.
- 2) You will need to move that number into a floating point register and then that number must be copied into an integer register.
- 3) You will need to extract the exponent from the integer register and stored in another register.
- 4) You will need to insert the Implied bit. I would suggest, zero out the exponent part by shifting left 9 then shifting right 9. Then add 8388608 ( $2^{23}$ ) to the number.
- 5) You will need to extract the fractional portion of the of the number. You will need the exponent to determine the shift. You only need to test to see that this is NOT EQUAL to 0 (if it is we have an integer)
- 6) Extract the Integer. You may want to “Rotate” the bits to the left.

You may want to use the following Assembler instructions in your code:

srl, add, sll, rol, sub, srlv and sllv