Objectives:
- Y86 assembly language

REMINDERS and GRADING CRITERIA:
- Read these sections from lab1... they haven’t changed.

LAB DESCRIPTION

Mandatory filename → lab4.ys

PROBLEM: Write an assembly language program using Y86 assembly to multiply each value from 1 to a given input value.

You should define multiple input values in memory. For instance, to define the input values as part of your program, you should have something that looks like:

```
.align 4
Input: .long 0x5       # answer = 1*2*3*4*5 = 120
       .long 0xc        # answer = 1*2*.... *10*11*12 = 479,001,600
       .long 0x3        # etc
       # could add more input data here
Done:  .long 0xFFFFFFFF
```

INPUT:
- The given input values should/will all be non-zero positive numbers.
- Your program should be able to handle an unknown number of input value lines and still work. That is, you cannot assume that you know how many input values there will be.
- You can check for the end of the input data by using -1 as the value to designate the end of the input; or check for the “Done” label location. You can use any name you like for the labels instead of “Input” and “Done”.

HINT: Since Y86 does not have a multiplication statement, you want to remember that 2*4 is the same as 2+2+2+2.

CONSTRAINTS:
- The maximum limit of a signed 32-bit integer value (as designated by a .long Y86 value) is $2^{31}-1$ which is equal to 2,147,483,647. You can assume that the result of any input value will not exceed this value.

OUTPUT:
- Be sure to return the result of each input, the multiplication of each value from 1 to the input value, in the register %eax and comment it heavily so the grader is left in no doubt as to where to look for the result of each input value.

LAB SUBMISSION

Read this section from lab1... it hasn’t changed except for:
- The labname for this lab is: lab4
- Be sure to submit the following files: lab4.ys and a README file