Final Project (100 pts)

• Your job is to implement an nasm x86 assembly program that inputs a number, computes \( N! \), and outputs the result.
  o Your program must work for \( N=\{0, 1, 2, ..., 12\} \).
  o You are NOT responsible for error checking the input for negative numbers, numbers greater than 12, or non-numeric input.
  o You may assume that I will test your code with valid input only.
  o Your code must assemble, link, and execute using nasm running on 32-bit Linux.

• In particular, your program must:
  1. read a number \( (N) \) from standard input,
  2. convert \( N \) from an ASCII character sequence to an integer value,
  3. calculate \( N! \) (\( N \) factorial),
  4. convert the result from an integer value to an ASCII character sequence, and
  5. output the result to standard output.

• To accomplish this, you will need to modify several source files given to you.
  • Download and decompress final_project.tgz located on the homework page.
  • There are several files within the final_project directory:
    ▪ atoi.asm – procedure to convert an ASCII string to integer value
    ▪ count_digits.asm – procedure to count the number of digits in an integer
    ▪ factorial.asm – procedure to compute \( N! \)
    ▪ itoa.asm – procedure to convert an integer to an ASCII string
    ▪ main.asm – driver program; contains _start and all procedure calls
    ▪ Makefile – makefile to assemble / load your source
    ▪ read_write.mac – contains macros to read from stdin, write to stdout

• Your task is to correctly implement the following procedures and macro (in their associated files).
  • atoi
  • count_digits
  • factorial
  • itoa
  • read_input (macro)

• Please review all provided source files for details regarding desired functionality, input argument(s), and return value(s). Note that the provided source code is hardcoded to output the result of \( 8! \) (8 factorial). Your solution must work for all \( N \).

• I will use the provided main.asm source and Makefile to test your code.
  • If you submit a modified main.asm or Makefile, it will be overwritten.
  • DO test your code incrementally, but remember to test your code with the original (provided) main.asm and Makefile.
Examples: red for stdin, blue for stdout.

UNIX> ./main
0
1

UNIX ./main
5
120

UNIX ./main
9
362880

UNIX> ./main
12
479001600