

## 1 Exercises

1. Recall that we define the factorial of a non-negative integer as

$$n! = \begin{cases} 1 & \text{if } n = 0 \\ n \cdot (n-1) \cdot (n-2) \cdots 2 \cdot 1 & \text{if } n > 0 \end{cases}$$

For this exercise you will write a 64 bit assembly program that will be called by a C++ program to calculate a user input factorial. The C++ program will handle all the input and output and the assembly program will calculate the factorial.

Specifically, the C++ program will ask the user for the input, it will call the assembly program to do the calculation and return the result to the C++ program, then the C++ program will display the result.

The assembly program will check to see if the input number (that came from the C++ program) is greater than or equal to 0. If the input is valid, the assembly program will do the factorial calculation and return the result. If the input was negative, then the assembly program will return  $-1$ . The C++ program is not to check the user's input, it will simply check the output of the assembly program, if the assembly program output is  $-1$ , the C++ program will display an error message and if not it will display the output of the assembly program.

Several runs of the program are below.

```
./prog
Input n: -3
Error: Input must be greater than or equal to 0.
```

```
./prog
Input n: 0
0! = 1
```

```
./prog
Input n: 12
12! = 479001600
```

```
./prog
Input n: 10
10! = 3628800
```

```
./prog
Input n: 20
20! = 2432902008176640000
```

2. The Fibonacci sequence is a sequence of positive integers, the first two integers are both 1 and every entry in the sequence after that is the sum of the two previous entries.

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946 . . .

For this exercise you will write a 64 bit assembly program that will be called by a C++ program to calculate a Fibonacci number. The C++ program will handle all the input and output and the assembly program will calculate the Fibonacci number.

Specifically, the C++ program will ask the user for the input, it will call the assembly program to do the calculation and return the result to the C++ program, then the C++ program will display the result.

The assembly program will check to see if the input number (that came from the C++ program) is greater than 0. If the input is valid, the assembly program will calculate the  $n^{th}$  Fibonacci number (where  $n$  is the user's input) and return the result. If the input was 0 or negative, then the assembly program will return  $-1$ . The C++ program is not to check the user's input, it will simply check the output of the assembly program, if the assembly program output is  $-1$ , the C++ program will display an error message and if not it will display the output of the assembly program.

Several runs of the program are below.

```
./prog
Input n: -1
Error: input must be greater than or equal to 1.
```

```
./prog
Input n: 4
3
```

```
./prog
Input n: 10
55
```

```
./prog
Input n: 20
6765
```

```
./prog
Input n: 50
12586269025
```

## 2 Submissions

Upload each of your programs (4 total, 2 assembler and 2 C++) to the MyClasses page for this assignment.