

# CS 271 Computer Architecture and Assembly Language

## Programming Assignment #2

### Objectives:

- 1) Getting string input
- 2) Designing and implementing a counted loop
- 3) Designing and implementing a post-test loop
- 4) Keeping track of a previous value
- 5) Implementing data validation

### Problem Definition:

Write a program to calculate Fibonacci numbers.

- Display the program title and programmer's name. Then get the user's name, and greet the user.
- Prompt the user to enter the number of Fibonacci terms to be displayed. Advise the user to enter an integer in the range [1 .. 46].
- Get and validate the user input ( $n$ ).
- Calculate and display all of the Fibonacci numbers up to and including the  $n^{\text{th}}$  term. The results should be displayed 5 terms per line with at least 5 spaces between terms.
- Display a parting message that includes the user's name, and terminate the program.

### Requirements:

- 1) The programmer's name and the user's name must appear in the output.
- 2) The loop that implements data validation must be implemented as a post-test loop.
- 3) The loop that calculates the Fibonacci terms must be implemented using the MASM *loop* instruction.
- 4) The *main* procedure must be modularized into at least the following sections (procedures are not required this time):
  - a. introduction
  - b. userInstructions
  - c. getUserData
  - d. displayFibs
  - e. farewell
- 5) Recursive solutions are not acceptable for this assignment. This one is about iteration.
- 6) The upper limit should be defined and used as a constant.
- 7) The usual requirements regarding documentation, readability, user-friendliness, etc., apply.
- 8) Submit your text code file (*.asm*) to Canvas by the due date.

### Notes:

- 1) It is not necessary to store the Fibonacci numbers in an array. The terms may be displayed as they are generated.
- 2) The second-order Fibonacci sequence is defined as:
  - a. The first two terms are both 1.
  - b. All other terms are calculated as the sum of the two previous terms.
  - c. The reason for restricting  $n$  to [1 .. 46] is that the 47<sup>th</sup> Fibonacci number is too big for *DWORD* data type.

**Example** (see next page)

**Example** (user input in *italics*):

Fibonacci Numbers

Programmed by Leonardo Pisano

What's your name? *Paul*

Hello, Paul

Enter the number of Fibonacci terms to be displayed

Give the number as an integer in the range [1 .. 46].

How many Fibonacci terms do you want? *50*

Out of range. Enter a number in [1 .. 46]

How many Fibonacci terms do you want? *14*

1	1	2	3	5
8	13	21	34	55
89	144	233	377	

Results certified by Leonardo Pisano.

Goodbye, Paul.

**Extra-credit options** (original definition must be fulfilled):

1. Display the numbers in aligned columns.
2. Do something incredible.