CS286 Computer Organization & Architecture
Fall 2026
Programming Project #2

Submission Due: November 18, 2026

Assignment Descriptions:

In this assignment, we will develop "Reverse Fibonacci Number Generator" using recursive subroutine calls, which calculates and displays as many as *N* Fibonacci numbers backward from a given Fibonacci number *X*. Fibonacci numbers are a sequence of integers that are generated using the following formula:

$$\alpha_{(n+2)} = \alpha_{(n+1)} + \alpha_{(n)}, \text{ where } n \geq 1, \alpha_1 = 1 \text{ and } \alpha_2 = 1 \quad (1)$$

The above formula generates the following sequence of integers:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 ……

Assignment Requirements:

Requirement 1: In using your Reverse Fibonacci Number Generator, you specify two numbers of *X* and *N*. The first parameter, *X*, is one of the Fibonacci numbers and your program is supposed to display *N* Fibonacci numbers ("*N*" as including "*X*") backward from the given Fibonacci number *X*. For example, if *X* = 55 and *N* = 4, your program should display:

Enter a valid Fibonacci number: 55
Enter the number of Fibonacci numbers to be displayed: 4
55, 34, 21, 13

Requirement 2: Your Reverse Fibonacci Number Generator needs to check if a given "*X*" is a valid Fibonacci number as shown below (i.e., your program should behave in the following way):

Enter a valid Fibonacci number: 54
Enter the number of Fibonacci numbers to be displayed: 4

"54" is not a valid Fibonacci number.

Enter a valid Fibonacci number: 55
Enter the number of Fibonacci numbers to be displayed: 4
55, 34, 21, 13

The above process should be repeated until a valid Fibonacci number is entered.

Requirement 3: The lower-bound of "*N*" should be detected and when your Fibonacci Number Generator reaches the first Fibonacci number (i.e., $\alpha_1 = 1$), your Fibonacci number generator should be stopped at that point. For example, your program should behave in the following way:

> **Enter a valid Fibonacci number:** 8
>
> **Enter the number of Fibonacci numbers to be displayed:** 10
>
> 8, 5, 3, 2, 1, 1
>
> No more Fibonacci number.

Requirement 4: Your program must have exactly one recursive subroutine module for calculating (and printing) Fibonacci numbers. Any Fibonacci number, except for α_1 and α_2, must be generated by recursive subroutine call(s) (i.e., none of the Fibonacci numbers (except for α_1 and α_2) should be stored in your program).

Requirement 5: Fibonacci numbers must be displayed only in your recursive subroutine module (i.e., none of the Fibonacci numbers, except for α_1 and α_2, can be displayed outside of your recursive subroutine module).

Requirement 6: The number of the Fibonacci numbers that should be displayed in the reverse order ('*N*') should be $3 < N < 40$. Any invalid input should be detected and be re-prompted to the user until a valid N is input.

Requirement 7: To simplify this programming assignment, you can assume *n* in formula (1) should be less than 29 ($n \leq 28$).

Requirement 8: Your program correctly terminates after it finishes execution (i.e., "crash exit" should not happen). Make sure that there is no error message from PC-SPIM when your program completes.

Submission:

Your source code file should be e-mailed to the TA (vthota@siue.edu) by 1:30 p.m. (section 001) or 3:00 p.m. (section 002) on November 18th.

Note: In your program source code, include your name, student ID (the last 3 digits) in your program header.

Late Submission:

- Penalty of -10% will be given for every 12 hours after the due (i.e., -10% for a submission within the first 12 hours after the due).
- Submission more than 48 hours after the due will not be accepted.

Grading Criteria:

- **Assembler error**: If the TA can not assemble your assembler source code file due to any assembler error: 0 ~ 20% of the credit for this assignment phase will be given.

- **Run-time error**: After your assembly source code file is successfully assembled to machine codes, if your program does not satisfy any requirement: 5% penalty for each minor problem. For any major failure, penalty depends on each such major problem.

Notes:

Note 1: The required function ("the reverse Fibonacci numbers") should be implemented based on the required recursive structure. Any submission that does not use the recursive structure or that uses an incomplete recursive structure is subject to major or minor penalties.

Note 2: It is required to use PC-SPIM posted to the CS286-001 course homes (programs developed for different versions of PC-SPIM will NOT be accepted for grading).

Note 3: This programming project is an individual programming project (no collaboration to complete the programming project is allowed). Violation of this rule is considered academic dishonesty.

Note 4: Academic dishonesty in Project #1 will at least result in 0 point for the entire programming project credit for this course (not only for Project #1). Serious academic dishonesty will result in a failing grade ('F' grade) for this course and the names of people will be reported to the CS department and the School of Engineering Dean's Office.

Expectations when a question is asked:

(1) Dr. Fujinoki will not debug your programs. Please do not e-mail me your source code file(s) unless you are requested to do so.

(2) If you have questions, stop by Dr. Fujinoki's office during the office hours (make sure to bring your *.asm source code files. Please avoid using e-mails for asking questions about the programming projects (there may be some exceptions).

(3) To make your visits productive, the followings are suggested:

- Identify where (in your source code) the problem exists
- Describe the symptom(s) of the problem
- Describe how the problem happens (always happen, sometime happen, the condition(s) for the problem to happen, etc.)
- Describe what you tried (to understand and/or solve the problem)

(4) Within 24 hours from the final submission due, Dr. Fujinoki will not respond to new questions (except any continued discussions for the questions you asked prior to the 24-hour time window).

(5) Some questions about Project #2 may be asked in the final exam.

APPENDIX: The first 31 Fibonacci numbers

n	**FBN**
1	1
2	1
3	2
4	3
5	5
6	8
7	13
8	21
9	34
10	55
11	89
12	144
13	233
14	377
15	610
16	987
17	1597
18	2584
19	4181
20	6765
21	10946
22	17711
23	28657
24	46368
25	75025
26	121393
27	196418
28	317811
29	514229
30	832040
31	1346269