Fibonacci Sequence

Description

This project will test the understanding of the material that you have learned thus far. You will **design** and implement a program called, "fib.asm", that will calculate the first *n* values of the Fibonacci sequence.

Fibonacci Sequence

The Fibonacci sequence is defined as a piecewise recursive function.

$$f(n) = \begin{cases} 0, & n = 0 \\ 1, & n = 1 \\ f(n-1) + f(n-2), & n > 1 \end{cases}$$

The following loop would be used to calculate the sequence iteratively:

```
start
        int prv = 0
        int nxt = 1
        int tmp
        for i = 1 to n
                store prv
                tmp = nxt
                nxt = prv + nxt
                prv = tmp
exit
```

The first 10 values in the Fibonacci sequence is: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

Algorithm

Start:

1. Prompt the user for a value of *n* between 1 and 9 inclusive. "*n*" represents the number of values of the sequence to be stored. For example, if the users enters 1, then you will only store the value {0}, if 9 is entered then you will store, {0, 1, 1, 2, 3, 5, 8, 13, 21}
2. Get user input
    a) It is possible to get larger values from the user.
    b) Integers are read in as a string of characters.
    c) Each character must be converted to a decimal value
3. Implement the loop to calculate and store each value in the sequence in an array of double words
4. Print the sequence in the debugger

Sample Output (user input is in bold)

```
Enter a value between 1 and 9: 8
```

Activity

1. Implement an assembly language program called, *fib.asm*.
2. You must reserve pace in the bss section for 9 **double-words** in which to store the values of the sequence.
3. Start writing your program using the lab template provided. Be sure to fill out the comments.
4. Reserve as must space as you need to store other values.
5. You must use at least one loop. You can use as many as you like.

6. Do not use any advanced techniques that you may find on the internet. You should not be using any other code as a resource.

What to Submit

- Your source code file (fib.asm).
- A screenshot of the output of the debugger showing the execution of the program. Be sure to show the prompt and input as well as printing the output in the debugger. This may require more than one screenshot.

Reminder

You are responsible to do your own work. **This is not a team project**. **Do not show anyone your code and do not look at, or copy, any code from any source, except the lecture notes or code created in class; create your own code.** Our lectures and in-class exercises contain all the information you need to complete this project. Any violation of the school's academic integrity policy or the policy of this class will result in a zero grade and an academic misconduct report filed with the school; no excuse will be accepted. Under no circumstances should you use any resources other than the code provided with this exam and the lecture notes.

Your program must assemble and run to receive any credit. If I cannot create an executable from your program, then you will receive a zero for your score. Treat this like any other exam, start right away and put effort into it.

Rubric

This project/exam is worth 100 points. The points will be distributed according to the chart below. If the program does not compile or is missing, then no points will be awarded.

Program 100

Requirements	**30**
Correctness	**50**
Code Quality	**20 (comments and formatting)**

Resources

There are code snippets available with this project. They will demonstrate how to print output to the console and take input from the console.

Extra Credit (10 points)

Take from user input any value of *n* up to the size of the buffer (array) that you created to store the sequence. The buffer should be 30 double-words in size for extra credit. Please see me about how to accomplish this if you are serious about understanding how. Do not copy code from the internet.

Extra-extra Credit (10 points)

Output the values of the sequence to the console. The output should look like the following including the commas and the braces:

{ 0, 1, 1, 2, 3, 5, 8, 13, 21, … }

Please see me about how to accomplish this if you are serious about understanding how. Do not copy code from the internet. **Extra-extra credit is only available if you've successfully completed the extra credit option.**