Recitation Assignment #5

Please do the following problems during your recitation session, including any additional problems given to you by your TA. Within 72hrs of your recitation session, you must upload the complete solutions to these problems to Sakai, so that your TAs can evaluate them in a timely fashion. **Please submit any .m files to Sakai. Your .m files should be named in the following format: NETIDRecitation5Problem#, where NETID is your NetID, and # is the problem number. For instance if your NetID was aaa111, and you were answering problem 3, your .m file would be named aaa111Recitation5Problem3.m. All of your .m files should be in the form of a function. If you do not meet these naming requirements, and do not save as a function, you will not receive credit for your submission.** Collaborative problems can be worked on in teams of up to 5 people, as long as each team member individually completes the problem uploads the solution as part of their own Sakai submission or shows the solution to their instructor individually, and lists the names of all collaborators in the Sakai submission. Collaboration and discussion of solutions is not permitted for questions labeled as individual problems.

1. [Collaboration] The Fibbonacci sequence is a sequence of numbers in which, after the first two numbers, each number is a sum of the proceeding numbers. The Fibonacci Sequence starts with 0 and then 1. For subsequent numbers, the sequence follows the following formula:

   $$Fibonacci(n) = Fibonacci(n-1) + Fibonacci(n-2)$$

   Where n is the current position in the sequence. The first 10 numbers in the sequence are:

   0 1 1 2 3 5 8 13 21 34

   Write a function that takes an integer number “n” as an input and returns a Fibonacci sequence of length n.

2. [Collaboration] Loops are very useful for doing the same calculation over and over, very quickly. Let’s assume that we are building a bridge and need to determine the grade of steel we need for each fastener. Assume that we have another set of programs that returns the tensile and shear forces that each joint will experience. We need to use the program we have already written to determine which kind of fastener to use en masse. As we want to store the data about fastener type in an array of numbers, please use the following values for each grade of steel:

Steel Grade String	Number Code
Grade 1 Low Carbon Steel	1
Grade 2 Low Carbon Steel	2
Grade 5 Medium Carbon Steel	5
Grade 8.2 Low Carbon Boron Steel	8.2

   Write program that takes in two arrays, the first with shear forces, the second array of the same length with tensile forces, for fastened joints. As an output, this function returns the type of metal to use in those joints. You can assume that no shear Forces are above 120000 and that no tensile forces are above 150000. Please call your function from Recitation 3 problem 1 to solve this. For example, if:

$$ShearForces = [10000, 100000, 50000]$$

And:

$$TensileForces = [25000, 120000,75000]$$

Then:

```
[Grades] = aaa111Recitation5Problem2(ShearForces,TensileForces)
```

Will return the array:

$$Grades = [1, 8.2, 5]$$

Remember that you can use strcmp to compare two strings, if you want to use it to solve this problem. You can also use length(arrayName) to determine the length of any array. You can download an example recitation problem 3.1 answer if yours was incorrect. Please rename it correctly. **PLEASE SUBMIT YOUR PROGRAM FROM RECITATION 3 PROBLEM 1 WITH THE REST OF YOUR SUBMISSIONS FOR THIS RECITATION.**

3. [Individual] Write a function that takes in a number, n, as an input and has one output, an array of length n that alternates 1's and 0's. If the length of the array is odd, start the array with a 1. If the length of the array is even, start the array with 0. Please use a loop to accomplish this. For example:

```
[alternating] = aaa111Recitation5Problem3(10)
```

Would return the array:

$$alternating = [0, 1, 0, 1, 0, 1, 0, 1, 0, 1]$$

While:

```
[alternating] = aaa111Recitation5Problem3(11)
```

Would return the array

$$alternating = [1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]$$

You can use the mod function to determine whether a number is even or odd. mod(n,2) will return 1 if n is odd and 0 if n is even.