1. Write a function fact(n) which computes n!.
   a. Write it recursively.
   b. Write it non-recursively.

2. Write a function fib(n) which computes the nth Fibonacci number.
   a. Write it recursively – no memorization.
   b. Write it recursively – with memorization.
   c. Write it non-recursively.

3. Python has a function type(n) which returns the “type” of object n.

   ```
   >>>
   >>> type(1)
   <class 'int'>
   >>> type([1])
   <class 'list'>
   >>> type(1)==type([1])
   False
   >>> type(1)==type([1][0])
   True
   >>>
   ```

   a. Explain each of the above.

   b. What will Python print what it evaluates the following list comprehension:

   ```
   sum([i for i in [1,2,3,'a',[1,2,3,4]] if type(i)==type(123)])
   ```

   Make sure that you understand why.

4. Look at the following code:

   ```
   def f(i,j):
       if i==0 or j==0:
           return 1
       return f(i-1,j)+f(i,j-1)
   ```

   a. What will be printed when you run it?

   b. This is the recursive version of another (non-recursive) program that we worked on. Which one?

   c. How is it working? In the non-recursive version we were storing the values in a 2 dimensional array. How/where are the values being stored in this version? Trace it in the simulator to see exactly what is happening.

   d. When we run the above with “print( f(19,19))”, it runs for a very long time. Why.

   e. When we memoize it, it runs almost instantaneously. The value 35345263800 is printed. Why?

   f. Write and run the memoized version of function f above. Use the memoized version of Fibonacci (on the handouts page of the lecture website) as a template.
5. What does the following function do, and how does it work?

```python
def f(m,n):
    if n==1:
        return m
    return m*f(m,n-1)
```

6. What does the following function do, and how does it work?

```python
def f(m,n):
    if n==1:
        return m
    return m+f(m,n-1)
```

7. What does the following function do, and how does it work?

```python
def f(m,n):
    if n==0:
        return m
    return 1+f(m,n-1)
```

8. What does the following function do, and how does it work?

```python
def f(s):
    if len(s)==0:
        return 
    return f(s[1:])+s[0]
```

9. Write a function, `make_random_array(n)`, which returns an nXn array filled with the integers 1 – nXn. Each number is assigned to a random place in the array, and each number appears only once. Each time you run your function for a given value of n, the numbers could appear in different places.