Print your name: ____________________________________________________________

- If you need more space to answer a question than we give you, you may use the additional blank sheet of paper attached to your exam. Make sure that we know where to look for your answer.

- Read each question carefully and make sure that you answer everything asked of you. Write legibly so that we can read your solutions. Please do not write anything in red.

- We suggest that for solutions that require you to write Python code, you include comments. They will help your grader understand what you intend, which can help you get partial credit.

- You have until noon to complete the exams
<table>
<thead>
<tr>
<th>Question</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Question 1    30 points

Answer each part.

(a) (6 points)
Using Big-O notation, give the run-time of sorting a list of length $n$ using mergesort.

(b) (6 points)
Using Big-O notation, give the run-time of appending a node to the end of a circular, singly-linked list with $n$ nodes and a sentinel.

(c) (6 points)
Using Big-O notation give the run-time of this Python function.

```python
def do_something(n):
    if n == 1:
        return 1
    else:
        return n + do_something(n-1)
```
(d) (6 points)
I write a hash function for a dictionary of size $n$. Every time that this hash function is called it simply returns a random number between 0 and $n - 1$. Is this a valid choice for a hash function? Briefly explain.

(e) (6 points)
I write a hash function for a dictionary of size $n$. Every time that this hash function is called it simply returns 0. Using Big-O notation give the run-time of finding a key-value pair for a dictionary that uses this hash function. Briefly justify your answer.
Question 2 10 points

Write an iterative (non-recursive) function that takes as input a list of $n$ numbers and returns True if the numbers are the first $n$ numbers of the Fibonacci sequence and False otherwise. Your function cannot modify the list in anyway. Recall that the first number in the Fibonacci sequence is 1, the second is 1, and each subsequent number is the sum of the two previous values in the sequence. Here, for example, are the first eight numbers in the sequence: 1, 1, 2, 3, 5, 8, 13, 21.
Question 3  10 points

Write an iterative (non-recursive) function called `max3D` that takes as input a three-dimensional (3-D) list of numbers and returns the maximum value in the entire 3-D list. Your function cannot use functions like `max` or `sort`, and must use either a for-loop or while-loop to iterate through the entire 3-D list looking for the maximum value. Your function cannot modify the list in anyway. An example of a 3-D list with three 2-D lists, each consisting of three, two, and one list is:

```
  [[[1, 2, 3], [4, 5, 6], [7, 8]], [[9], [10]], [[11, 12, 13, 14]]]
```
**Question 4**  
15 points

Write a Python function `histogram` that takes as input a single parameter, the name of a text file, and determines the frequency of each character in the file. Your function should read through the input file one line at a time. For each line, you should loop through each character of the line using a dictionary to keep track of how many times each character occurs. This includes spaces, punctuation, and newlines `\n`. You need only build the dictionary, you need not print anything. For example, if the text file is:

```
it was the best
of times, it was the
worst of times
```

Then the frequencies are:

```
'a': 2
'b': 1
'e': 5
'f': 2
'h': 2
'i': 4
'm': 2
'o': 3
'r': 1
's': 6
't': 8
'w': 3
' ': 9
',': 1
'\n': 3
```
Question 5 15 points

Shown below is some code for constructing a directed graph (using an adjacency list) and traversing and printing the value of each node in a graph. Draw the graph (using the graph vertices shown below) and then specify the output generated by the call to traverse_graph shown below.

```python
def traverse_graph(graph, start, path=[]):
    L = [start]
    while L != []:
        v = L[0]
        del L[0]
        if v not in path:
            path.append(v)
            for n in graph[v]:
                L.append(n)

    return(path)

graph = {}
graph["A"] = ["C"]
graph["B"] = ["A"]
graph["C"] = ["A", "B"]
graph["D"] = ["B", "E"]
graph["E"] = ["C"]
graph["F"] = ["A", "C", "D"]

print( traverse_graph(graph, "F") )
```

```
A -- C
B -- F
F -- A, C, D
E -- C
```
Question 6          20 points

Write some Python code for a singly-linked list (without a sentinel). Use the space below and/or the next blank page for your solution. Your code should consist of:

- (5 pts) A constructor for a Node class that takes two parameters val and nxt as input. The parameter val is the data to be stored in the node. The parameter nxt is another node to which the newly constructed node will be connected. This constructor should set the object instance variables of the same name to the value of these parameters.

- (7 pts) A constructor for an SLL class that takes one parameter n as input specifying how many nodes will be in the singly linked list. This constructor should build a singly linked list with a random integer between 0 and 100 in each of n nodes. The instance variable head should be assigned to the first node in the list.

- (8 pts) Write an SLL member function that iteratively (i.e., non-recursively) finds and returns the maximum element in a list. Your solution cannot sort or modify the list.
# THIS PAGE LEFT INTENTIONALLY BLANK