CSC 120 **ICA-18**

Work with your neighbor. (This will be graded for participation only.)

1. Implement a queue with a Python list. Make the front of the queue the last item in the list.

```
class Queue:
    def __init__(self):



    def enqueue(self, item):




    def dequeue(self):
```

Given the statement below:

```
q = Queue()
```

write what `print(q)` would output after each of the statements below:

```
q.enqueue(10)        ________________________

q.enqueue(20)        ________________________

q.enqueue(30)        ________________________

q.dequeue()          ________________________

q.enqueue(8)         ________________________
```

What is the size of the `Queue` `q` at this point? ________

2. Hot potato simulation. Write a function `hot_potato(q, num)` that takes a queue `q` and the number of rounds of simulation `num` and eliminates the correct element after `num` rounds.

Note: We may not get to all of these problems below today.

3. Write a recursive function `sumlist(alist)` that returns the sum of the elements in `alist`.

   What is the base case?

   What is the recursive case?

4. Write a recursive function `string_len(s)` that returns the length of the string `s`.

   What is the base case?

   What is the recursive case?

5. Write a recursive function `join_all(alist)` that takes a list `alist` and returns a string consisting of every element of `alist` concatenated together.

   What is the base case?

   What is the recursive case?

6. Write a recursive function `join_all(alist,sep)` that takes a list `alist` and returns a string consisting of every element of `alist` concatenated together separated by the string `sep`.

   What is the base case?

   What is the recursive case?