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.

ANS:

```
class Queue:
    # the front of the queue is the last item in the list

    def __init__(self):
        self._items = []

    def enqueue(self, item):
        self._items.insert(0, item)

    def dequeue(self):
         return self._items.pop()

     def is_empty(self):
              return self._items == []


     def __str__(self):
          return str(self._items)
```

Given the statement below:

```
q = Queue()
```

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

ANS:

`q.enqueue(10)`	`[10]`
`q.enqueue(20)`	`[20, 10]`
`q.enqueue(30)`	`[30, 20, 10]`
`q.dequeue()`	`[30, 20]`
`q.enqueue(8)`	`[8,30, 20]`

What is the size of the `Queue q` at this point? `3`

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.

   ANS:

```
def hot_potato(q, num):

    for i in range(num):
        x = q.dequeue()
        q.enqueue(x)

    return q.dequeue()
```

NOTE: Problems 3-6 were moved to ICA-19