Problem 3: Implement a **Stack** using a single **Queue**.

```
class Stack:
    def __init__(self):
        self.queue = []

    def push(self, value):
        self.queue.append(value)

    def pop(self):
        if not self.is_empty():
            size = len(self.queue)
            for _ in range(size - 1):
                self.queue.append(self.queue.pop(0))
            return self.queue.pop(0)

    def top(self):
        if not self.is_empty():
            size = len(self.queue)
            for _ in range(size - 1):
                self.queue.append(self.queue.pop(0))
            return self.queue[0]

    def size(self):
        return len(self.queue)

    def is_empty(self):
        return len(self.queue) == 0

    def print_stack(self):
        print("Stack:", self.queue)

stack = Stack()
stack.push(1)
```

```
stack.push(2)

stack.push(3)

stack.print_stack()

print("Size:", stack.size())

print("Top:", stack.top())

print("Pop:", stack.pop())

stack.print_stack()
```