# 12323 - LAB 03

## Instructions

1. Access the auto-grader at https://c200.luddy.indiana.edu
2. Please write the code for the problems in python language
3. The code should be readable with variables named meaningfully
4. Plagiarism is unacceptable and we have ways to find it, so do not do it
5. Don't change the function signature (name of the function and number and types of arguments) provided in this file.
6. Once you pass all the tests on the auto grader, show your work to the teaching assistant

## Problem

### Question

Implement Queue data structure using two stacks with a given capacity. The class definition should have the following four methods.
**1. init(len):** This takes the length of the queue as argument. The method should have two stacks initialized named s1 and s2 where the push operation takes place on s1.
**2. enque(number):** This takes the number as the argument. Returns 1 on successful enque operation else return -1.
**3. deque():** This returns the element if the deque is successful else returns -1.
**4. print():** This return the status of the stacks as (contents of stack s1 , contents of stack s2, no of elements in the queue).

### Class signature

```
class Queue:
    def __init__(self,len):
        self.s1 = []
        self.s2 = []
    def enque(self,num):
        pass
    def deque(self):
        pass
    def print(self):
        pass
```

### Test cases

```
q = Queue(3)
# Makes a queue with 3 length

q.enque(10)
# Adds 10 to the queue
# Returns 1

q.enque(20)
# Adds 20 to the queue
# Returns 1
```

```
q.enque(30)
# Adds 30 to the queue
# Returns 1

q.enque(50)
# Does not add 50 to the queue
# Maximum capacity has been reached
# Hence should return -1


q.deque()
# Removes 10 from the queue
# Returns element that is removed else returns -1.

q.print()
# Returns ([],[30,20],2)
```