Queue with linked list

```
class nd:
    def __init__(self,data):
        self.data=data
        self.next=None
class Queue:
    def __init__(self):
            self.front=None
            self.rear=None
    def isemp(self):
        return self.front is None
    def enqueue(self,data):
        nd1=nd(data)
        if self.rear is None:
            self.front=self.rear=nd1
            return
        self.rear.next=nd1
        self.rear=nd1
    def dequeue(self):
        if self.isemp():
            return None
        temp=self.front
        self.front=temp.next
        if self.front is None:
            self.rear=None
        return temp.data
    def display(self):
        cur=self.front
        while cur:
            print(cur.data, end=' ')
            cur=cur.next
        print()

q=Queue()

q.enqueue(20)
q.enqueue(10)
q.enqueue(30)
q.enqueue(40)
q.enqueue(90)
q.display()
```

```
20 10 30 40 90
```

```
q.dequeue()
q.display()
```

```
10 30 40 90
```

```
q.isemp()
```

```
False
```

Function to reverse a queue

```
def reverse(a):
    queue=a
    l=len(queue)
    print(queue[::-1])

inp=[20,30,100,60,340]
reverse(inp)
```

```
[340, 60, 100, 30, 20]
```

1. You are given a function. Check Password(char,n);The function accepts string str of size n as an argument. Implement the function which returns 1 if the given str is valid password else zero. str is a valid password if it satisfies te following conditions:
   - At least has 4 character
   - At least one numeric digit
   - at least one capital letter
   - must not have space or slash(/)
   - starting characterr must not be a number

```
def Check_Password(char,n):

    if len(char)<4:
        return 0
    digit=any(c.isdigit() for c in char)
    upper=any(c.isupper() for c in char)
    special_char=any(c in [' ','/'] for c in char)
    if digit and upper and not special_char:
        return 1
    else:
        return 0

Check_Password('aBc21b',6)
```

```
1
```

```
Check_Password('anc21b',6)
```

```
0
```