```
def Password(str, n):

    if n < 4:
        return 0


    if str[0].isdigit():
        return 0


    has_digit = False
    has_uppercase = False


    for char in str:
        if char.isdigit():
            has_digit = True
        if char.isupper():
            has_uppercase = True
        if char == ' ' or char == '/':
            return 0
    if has_digit and has_uppercase:
        return 1
    else:
        return 0

password = "amrutha"
n = len(password)
print(Password(password, n))
```

```
0
```

```
password = "Amrutha2006"
n = len(password)
print(Password(password, n))
```

```
1
```

```
class queue:
    def __init__(self):
        self.queue=[]
    def enqueue(self,item):
        self.queue.append(item)
    def dequeue(self):
        if len(self.queue)!=0:
            return self.queue.pop(0)
        else:
            print("queue is empty")
    def isempty(self):
        return len(self.queue)==0
```

```
    def size(self):
        return len(self.queue)
    def display(self):
        if self.queue:
            return self.queue
        else:
            'queue is empty'



r=queue()

r.enqueue(7)
r.enqueue(9)
r.enqueue(6)
r.enqueue(3)

r.size()

4

r.display()

[7, 9, 6, 3]

r.isempty()

False

r.dequeue()

7

r.dequeue()

9

r.dequeue()

6

r.dequeue()

3

r.dequeue()

queue is empty

r.isempty()

True
```

```python
from queue import Queue

def reverseQueue(q):
    stack = []
    while not q.empty():
        stack.append(q.get())
    while stack:
        q.put(stack.pop())
q = Queue()
elements = [1, 2, 3, 4, 5,6,7,9,0]

for elem in elements:
    q.put(elem)

print("Original Queue:")
original_elements = []
for i in range(q.qsize()):
    elem = q.get()
    print(elem, end=" ")
    original_elements.append(elem)
    q.put(elem)
reverseQueue(q)
print("\nReversed Queue:")
for i in range(q.qsize()):
    elem = q.get()
    print(elem, end=" ")
    q.put(elem)
```

```
Original Queue:
1 2 3 4 5 6 7 9 0
Reversed Queue:
0 9 7 6 5 4 3 2 1
```