Started on	Saturday, 19 April 2025, 10:00 AM
State	Finished
Completed on	Saturday, 19 April 2025, 10:59 AM
Time taken	59 mins 38 secs
Grade	**80.00** out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

Write a python program to create a stack with a maximum size of 7 using Lifo Queue. Get the input from the user and check whether the stack is full and then display the stack values in reverse order

For example:

Input	Result
4 Maths Physics Chemistry Biology	False Biology Chemistry Physics Maths
7 Maths Physics Chemistry Biology History Economics English	True English Economics History Biology Chemistry Physics Maths

Answer: (penalty regime: 0 %)

Reset answer

```
from queue import LifoQueue
stack = LifoQueue(maxsize=7)
n= int(input())
for i in range(n):
    stack.put(input())
print(stack.full())
for j in range(n):
    print(stack.get())

```

	Input	Expected	Got	
✔	4 Maths Physics Chemistry Biology	False Biology Chemistry Physics Maths	False Biology Chemistry Physics Maths	✔

	Input	Expected	Got	
✔	7 Maths Physics Chemistry Biology History Economics English	True English Economics History Biology Chemistry Physics Maths	True English Economics History Biology Chemistry Physics Maths	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **2**

Correct

Mark 20.00 out of 20.00

Develop a python program to get 5 values from the user and display the values using circular queue

For example:

Input	Result
1 2 3 4 5	1 2 3 4 5
10 20 30 40 50	10 20 30 40 50

Answer: (penalty regime: 0 %)

Reset answer

```
class MyCircularQueue():
    def __init__(self, k):
        self.k = k
        self.queue = [None] * k
        self.front = self.rear = -1

    def enqueue(self, data):
        if (self.rear + 1) % self.k == self.front:
            print("Overflow")
        elif self.front == -1:
            self.front = self.rear = 0
            self.queue[self.rear] = data
        else:
            self.rear = (self.rear + 1) % self.k
            self.queue[self.rear] = data

    def printCQueue(self):
        if self.front == -1:
            print("Empty Queue")
        elif self.front <= self.rear:
            for i in range(self.front, self.rear + 1):
                print(self.queue[i], end=" ")
```

	Input	Expected	Got	
✔	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	✔
✔	10 20 30 40 50	10 20 30 40 50	10 20 30 40 50	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **3**

Not answered

Mark 0.00 out of 20.00

Write a python code to create a student class with default constructor and a user defined function to display the welcome with student name given to the function.

For example:

Input	Result
saveetha	This is non parametrized constructor Hello saveetha

Answer: (penalty regime: 0 %)

```
1
```

Question **4**

Correct

Mark 20.00 out of 20.00

Write a python program to delete two neighboring identical letters.

For example:

Input	Result
abbaca	ca

Answer: (penalty regime: 0 %)

```
def removeDuplicates(S):
    list_1 = []
    for i in S:
        if list_1 and i==list_1[-1]:
            list_1.pop()
        else:
            list_1.append(i)
    return "".join(list_1)

S = input()
print(removeDuplicates(S))
```

	Input	Expected	Got	
✔	abbaca	ca	ca	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **5**

Correct

Mark 20.00 out of 20.00

Develop a python programming to add a few fruits name in the queue(from rear end) without any duplication

For example:

Input	Result
5 Papaya Mango Guava Apple Mango	['Apple', 'Guava', 'Mango', 'Papaya']
3 Grapes Banana Grapes	['Banana', 'Grapes']

Answer: (penalty regime: 0 %)

```
l=[]
x=int(input())
for i in range(x):
    b=input()
    if b not in l:
        l.append(b)
l.reverse()
print(l)
```

	Input	Expected	Got	
✔	5 Papaya Mango Guava Apple Mango	['Apple', 'Guava', 'Mango', 'Papaya']	['Apple', 'Guava', 'Mango', 'Papaya']	✔
✔	3 Grapes Banana Grapes	['Banana', 'Grapes']	['Banana', 'Grapes']	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.