Started on	Wednesday, 6 November 2024, 11:05 AM
State	Finished
Completed on	Wednesday, 6 November 2024, 12:05 PM
Time taken	59 mins 19 secs
Grade	100.00 out of 100.00

```
Question 1
Correct
Mark 20.00 out of 20.00
```

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

For example:

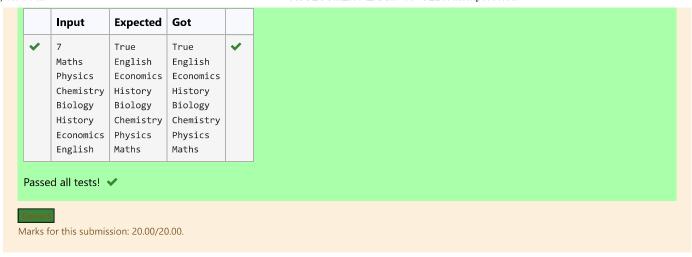
Input	Result
4	False
Maths	Biology
Physics	Chemistry
Chemistry	Physics
Biology	Maths
7	True
Maths	English
Physics	Economics
Chemistry	History
Biology	Biology
History	Chemistry
Economics	Physics
English	Maths

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 i in range(n):
    print(stack.get())
```

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



Question 2
Correct
Mark 20.00 out of 20.00

Write a python program to delete two neighboring non-identical letters(lower case and upper case).

Example: AbBbA

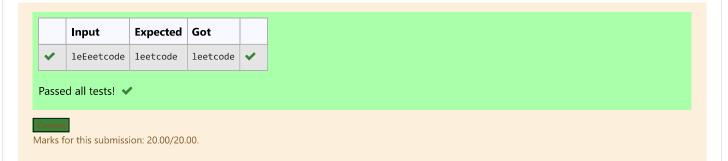
lowercase b and uppercase B will get removed

For example:

Input	Result	
leEeetcode	leetcode	

Answer: (penalty regime: 0 %)

```
1 def delete(n):
        stack=[]
 2
 3 -
        for i in n:
            if stack and stack[-1]==i.upper():
 4
 5
                stack.pop()
            else:
 6 ,
 7
                stack.append(i)
        return ''.join(stack)
 8
 9
    n=input()
10 | print(delete(n))
```



```
Question 3
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 %)

```
1 v class Queue:
 2 ,
        def __init__(self):
 3
           self.queue=list()
 4 ,
        def add_element(self,val):
 5
            if val not in self.queue:
 6
                self.queue.insert(0,val)
 7
                return True
 8
            return False
 9 ,
        def size(self):
10
            return len(self.queue)
   TheQueue=Queue()
11
12
   n=int(input())
13 v for i in range(n):
        TheQueue.add_element(input())
14
15 print(TheQueue.queue)
```

	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! ✓

```
Question 4

Correct

Mark 20.00 out of 20.00
```

Develop a python program to remove 3 values from the user and display the values using circular queue

For example:

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

Answer: (penalty regime: 0 %)

```
Reset answer
```

```
class MyCircularQueue():
 1 🔻
 2
        def __init__(self, k):
            self.k = k
 3
 4
            self.queue = [None] * k
 5
            self.head = self.tail = -1
 6
        def enqueue(self, data):
            if ((self.tail + 1) % self.k == self.head):
 7
 8
                print("The circular queue is full\n")
 9
            elif (self.head == -1):
10
                self.head = 0
11
                self.tail = 0
                self.queue[self.tail] = data
12
13
            else:
                self.tail = (self.tail + 1) % self.k
14
15
                self.queue[self.tail] = data
        def dequeue(self):
16 •
17 🔻
            if(self.head==-1):
                print("The circular queue is empty\n")
18
19
            elif(self.head==self.tail):
20
                temp=self.queue[self.head]
21
                self.head=-1
つつ
                celf tail=-1
```

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

Passed all tests! 🗸

Marks for this submission: 20.00/20.00.

Question **5**Correct

Mark 20.00 out of 20.00

Add destructor in the following python code

For example:

Input	Result	
Kevin	Person Created	
24	Kevin 24	
	Kevin Object Destroyed	

Answer: (penalty regime: 0 %)

```
Reset answer
```

```
1 v class Person:
2 🔻
        def __init__(self,name,age):
3
            print("Person Created")
4
            self.name = name
5
            self.age = age
        def printInfo(self):
6
7
           print(self.name, self.age)
8
       #add destructor
            print(self.name,"Object Destroyed")
9
10
   name=input()
11
   age=int(input())
12
13
   P1=Person(name,age)
14
   #P2=Person("Joe",34)
15
   P1.printInfo()
   #P2.printInfo()
16
   del P1
17
```

	Input	Expected	Got	
•	Kevin 24	Person Created Kevin 24 Kevin Object Destroyed	Person Created Kevin 24 Kevin Object Destroyed	~
~	John 46	Person Created John 46 John Object Destroyed	Person Created John 46 John Object Destroyed	~

Passed all tests! 🗸

Marks for this submission: 20.00/20.00.