Started on	Wednesday, 7 May 2025, 3:09 PM
State	Finished
Completed on	Wednesday, 7 May 2025, 3:30 PM
Time taken	21 mins 32 secs
Grade	**100.00** out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

Develop a python program to add only the even unique numbers using appendleft() from n given numbers

For example:

Input	Result
5 2 5 8 2 4	deque([4, 8, 2])
6 3 5 2 8 2 5	deque([8, 2])

Answer: (penalty regime: 0 %)

```
from collections import deque
class Queue:
  def __init__(self):
      self.queue = deque()
  def add_element(self,val):
      if val%2==0 and val not in self.queue:
          self.queue.appendleft(val)
          return True
      return False
TheQueue = Queue()
n=int(input())
for i in range(n):
    TheQueue.add_element(int(input()))
print(TheQueue.queue)
```

	Input	Expected	Got	
✓	5 2 5 8 2 4	deque([4, 8, 2])	deque([4, 8, 2])	✓
✓	6 3 5 2 8 2 5	deque([8, 2])	deque([8, 2])	✓

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 string values from the user and display the values using circular queue

For example:

Input	Result
4 Python Java C C++	Python Java C C++
5 Java C# C Python C++	Java C# C Python C++

Answer: (penalty regime: 0 %)

Reset answer

```
class MyCircularQueue():
    def __init__(self, k):
        self.k = k
        self.queue = [None] * k
        self.head = self.tail = -1
    def enqueue(self, data):
        if ((self.tail + 1) % self.k == self.head):
            print("The circular queue is full\n")
        elif (self.head == -1):
            self.head = 0
            self.tail = 0
            self.queue[self.tail] = data
        else:
            self.tail = (self.tail + 1) % self.k
            self.queue[self.tail] = data
    def printCQueue(self):
        if(self.head == -1):
            print("No element in the circular queue")
        elif (self.tail >= self.head):
            for i in range(self.head, self.tail + 1):
                print(self.queue[i], end=" ")
            print()
```

	Input	Expected	Got	
✓	4 Python Java C C++	Python Java C C++	Python Java C C++	✓
✓	5 Java C# C Python C++	Java C# C Python C++	Java C# C Python C++	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **3**

Correct

Mark 20.00 out of 20.00

Write a Python program to find the square of all elements in a list using list comprehension

For example:

Input	Result
3 11 22 33	[11, 22, 33] [121, 484, 1089]

Answer: (penalty regime: 0 %)

```
a=int(input())
l=[]
for i in range(a):
    b=int(input())
    l.append(b)
print(l)

s=[i*i for i in l]

print(s)
```

	Input	Expected	Got	
✓	3 11 22 33	[11, 22, 33] [121, 484, 1089]	[11, 22, 33] [121, 484, 1089]	✓
✓	5 2 3 6 9 45	[2, 3, 6, 9, 45] [4, 9, 36, 81, 2025]	[2, 3, 6, 9, 45] [4, 9, 36, 81, 2025]	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **4**

Correct

Mark 20.00 out of 20.00

Write a python program to reverse a string using stack concept

For example:

Input	Result
Python	nohtyP

Answer: (penalty regime: 0 %)

```
def swap(s, i, j):
    temp = s[i]
    s[i] = s[j]
    s[j] = temp

def reverse(s, i=0, j=0):
    if j == len(s):
        return i

    i = reverse(s, i, j + 1)
    if i <= j:
        swap(s, i, j)
        i += 1

    return i


s = input()

chars = [*s]
reverse(chars)
s = ''.join(chars)
```

	Input	Expected	Got	
✓	Python	nohtyP	nohtyP	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **5**

Correct

Mark 20.00 out of 20.00

Write a python program to create a stack with a maximum size of 5 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 10 20 30 40	False 40 30 20 10
5 2 4 6 8 3	True 3 8 6 4 2

Answer: (penalty regime: 0 %)

Reset answer

```
from queue import LifoQueue
stack = LifoQueue(maxsize=5)
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 10 20 30 40	False 40 30 20 10	False 40 30 20 10	✓
✓	5 2 4 6 8 3	True 3 8 6 4 2	True 3 8 6 4 2	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.