
Started on Thursday, 27 March 2025, 3:02 PM

State Finished

Completed on Thursday, 27 March 2025, 3:26 PM

Time taken 24 mins 43 secs

Grade 100.00 out of 100.00

Question 1

Correct

Mark 20.00 out of 20.00

Write a python program to read and then print the integer variable.

men_stepped_on_the_moon=

print()

For example:

| Input | Result |
|-------|--------|
| 120 | 120 |

Answer: (penalty regime: 0 %)

```

1 |
2 | def men_stepped_on_the_moon(n):
3 |     if n == n:
4 |         return(n)
5 |     else:
6 |         return('0')
7 | n=int(input())
8 | print(n)
9 |
10 |

```

| | Input | Expected | Got | |
|---|-------|----------|-----|---|
| ✓ | 120 | 120 | 120 | ✓ |
| ✓ | 104 | 104 | 104 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question 2

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 | False |
| 10 | 40 |
| 20 | 30 |
| 30 | 20 |
| 40 | 10 |
| 5 | True |
| 2 | 3 |
| 4 | 8 |
| 6 | 6 |
| 8 | 4 |
| 3 | 2 |

Answer: (penalty regime: 0 %)

Reset answer

```

1 from queue import LifoQueue
2 stack = LifoQueue(maxsize=5)
3 n= int(input())
4 for i in range(n):
5     stack.put(input())
6 print(stack.full())
7 for i in range(n):
8     print(stack.get())

```

| | Input | Expected | Got | |
|---|-------|----------|-------|---|
| ✓ | 4 | False | False | ✓ |
| | 10 | 40 | 40 | |
| | 20 | 30 | 30 | |
| | 30 | 20 | 20 | |
| | 40 | 10 | 10 | |

| | Input | Expected | Got | |
|---|-------|----------|------|---|
| ✓ | 5 | True | True | ✓ |
| | 2 | 3 | 3 | |
| | 4 | 8 | 8 | |
| | 6 | 6 | 6 | |
| | 8 | 4 | 4 | |
| | 3 | 2 | 2 | |

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 delete two neighboring identical letters.

For example:

| Input | Result |
|--------|--------|
| abbaca | ca |

Answer: (penalty regime: 0 %)

```

1 def removeDuplicates(S):
2     list_1 = []
3     for i in S:
4         if list_1 and i==list_1[-1]:
5             list_1.pop()
6         else:
7             list_1.append(i)
8     return "".join(list_1)
9
10 S = input()
11 print(removeDuplicates(S))

```

| | Input | Expected | Got | |
|---|--------|----------|-----|---|
| ✓ | abbaca | ca | ca | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

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 2 3 4 5 | 4 5 |
| 10 20 30 40 50 | 40 50 |

Answer: (penalty regime: 0 %)

Reset answer

```

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

Correct

Marks for this submission: 20.00/20.00.

Question 5

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 %)

```

1 from collections import deque
2 class Queue:
3     def __init__(self):
4         self.queue = deque()
5     def add_element(self, val):
6         if val%2==0 and val not in self.queue:
7             self.queue.appendleft(val)
8             return True
9         return False
10 TheQueue = Queue()
11 n=int(input())
12 for i in range(n):
13     TheQueue.add_element(int(input()))
14 print(TheQueue.queue)

```

| | Input | Expected | Got | |
|---|----------------------------|------------------|------------------|---|
| ✓ | 5 2 5 8 2 4 | deque([4, 8, 2]) | deque([4, 8, 2]) | ✓ |

| | Input | Expected | Got | |
|---|---------------------------------|---------------|---------------|---|
| ✓ | 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.