

Thank you. Your test submitted.

You have cleared this assessment.

Obtained Percentage Obtained Marks

100 %

13 / 13

Best Attempt Score:100 % on 23-03-2025

```
Given the following stack:
```

```
(Top--->Bottom)
7->12->1->5->4->2
```

What will be the updated content of the above stack when passed to the below Python function?

Assumption: All the references to the necessary files are available

```
def fun(stack1):
    if(stack1.is_empty()):
        return
    else:
        s1=Stack(7)
        while(not stack1.is_empty()):
            s1.push(stack1.pop())
            s1.push(5)
            s1.push(s1.pop()+s1.pop())
        while(not s1.is_empty()):
            stack1.push(s1.pop())
```

Note: The options show the stack content from top to botto

Warning

This operation is disabled.

Ok

- 0 12->17->6->10->9->17
- 0 12->17->16->10->9->7
- 12->17->6->10->19->7
- 12->17->6->10->9->7

```
Consider the below Python code:
def process(input stack):
    output_stack=Stack(5)
    count=0
    while(not input_stack.is_empty()):
         temp=input stack.pop()
         for i in temp:
              count+=1
         output stack.push(str(count)+temp)
    return output stack
input_stack=Stack(5)
                                                     Wai
input stack.push("India")
input stack.push("Australia")
                                                     This
input stack.push("England")
input_stack.push("SouthAfrica")
process(input stack).display()
What will be the status of output_stack after execution of a
Assumption: All the references to the necessary files are available
Note: Consider that the elements of stack in the options are shown
     11SouthAfrica-->18England-->27Australia-->32India
    5India-->9Australia-->7England-->11SouthAfrica
     11SouthAfrica-->7England-->9Australia-->5India
                                                         pringboard
```

```
Consider the below Python code:
Assumption: All the references to the necessary files are available
def func(input queue):
     q1=Queue(input queue.max size)
     q2=Queue(input queue.max size)
     for i in range(input_queue.front,input_queue.rear+1):
         item=input queue.dequeue()
         if(item>10):
              q1.enqueue(item)
         else:
              q2.enqueue(item)
                                                             Warr
input queue=Queue(5)
                                                             This o
input queue.enqueue(12)
input queue.enqueue(10)
input queue.enqueue(15)
input queue.enqueue(1)
input queue.enqueue(20)
func(input queue)
What is the status of Queues q1 and q2 after execution of above code?
The order of elements in options are from Front to Rear
    q1 contains: 12-->20-->15 and q2 contains: 10-->1
    q1 contains: 12-->15-->20 and q2 contains: 1-->10
q1 contains: 12-->15-->20 and q2 contains: 10-->1
                                                                   ingboard
    q1 contains: 12-->10-->15-->20 and q2 contains: 1
```

Visit For

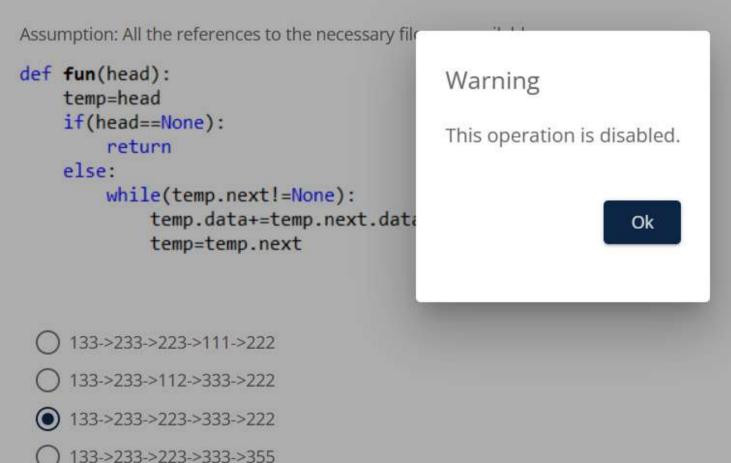
Consider a treasure hunt game in which you are har of next clue. The second clue links to the location of	nded a clue that leads you to another clue and so on until you find the treasure. The first clue links (or points) to the location third, and so on.
Which data structure can be optimally used to mode	el the treasure hunt game?
○ Stack	Warning
Linked List	This operation is disabled.
○ Graph	
O Queue	Ok

Given the following linked list: 1->4->6->7->9 What would be the state of the linked list when the following Python function is invoked by passing the head node of the above linked list and 5? def fun(head,n): temp=head if(head==None): return Warning else: i=1 while(i<n): This operation is disabled. val=temp.get data() temp=temp.get_next() head.set data(val) i+=1 Ok temp1=head.get data() temp.set data(temp1) Assumption: All the references to the necessary files are available. Ivieurou get_next() returns the link to the next node. Method get_data() returns data of the node.

7->4->6->7->9

Given the following Linked List: 12->121->112->111->222

What would be the state of the linked list when the following Python function is invoked by passing the head node of the above linked list?

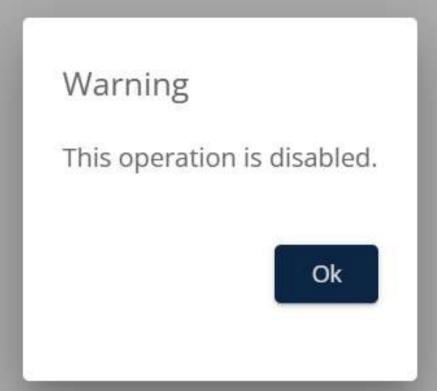


is also placed on hold and this continues till the ope When the operator becomes available, the first call	rator becomes available.	to him/her.
Which of the below data structure is optimal to rep	741 11 1 1 15	
	Warning	
○ Tree	This operation is disabled.	
Graph		
Queue	Ok	
○ Stack		

What will be the output of the below Python function when it is invoked by passing n as 4?

Assumption: All the references to the necessary files are available and n is a non-zero value.

```
def fun(n):
    q=Queue(n)
    if(n==0):
        return 1
    else:
        for i in range(1,n+1):
            q.enqueue(i*i)
        temp=1
        while(not q.is_empty()):
            temp+=q.dequeue()
        return temp
```



- 31
- \bigcirc 30
- () 25
- Function fun() would always return 1

What does the below Python function do using Queue data structure?

Assumption: All the references to the necessary files are available and n is a non-zero value.

```
def fun(n):
    q=Queue(n) # This statement creat
    if(n==0):
                                         Warning
        return 1
    else:
        for i in range(1,n+1):
                                         This operation is disabled.
            q.enqueue(i)
        temp=1
        while(not q.is_empty()):
                                                            Ok
            temp*=q.dequeue()
        return temp
```

- Function fun() returns the binary equivalent of a number 'n'
- Function fun() returns the factorial of a number 'n'
- Function fun() returns the sum of first 'n' numbers
- Function fun() would always return 1

Which of the following statements are FALSE regarding Array data structure? Select TWO correct options. Warning Deleting an element from the beginning of a Inserting an element at the end of an array d This operation is disal Elements of the array cannot be accessed rai Elements of an array cannot be accessed sec

Which of the following statements are TRUE with respect to the Array Data Structure? Select THREE CORRECT options.

- Inserting an element at the end requires shif
- Elements of an array can be accessed randor
- Inserting an element at beginning requires sl
- Deleting an element from beginning requires

Warning

This operation is disabled.

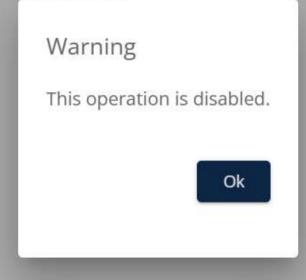
Given the following linked list:

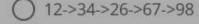
12->34->26->67->98

What would be the state of the linked list when the following Python function is invoked by passing the head node of the above linked list and 3?

Assumption: All the references to the necessary files are available

```
def fun(head,n):
    temp=head
    if(head==None):
        return
    else:
        i=1
        while(i!=n):
            temp=temp.next
        i+=1
        temp1=head.data
        head.data=temp.data
        temp.data=temp1
```







- 0 67->34->26->12->98
- 98->34->26->67->12

```
Given the following stack: (Top--->Bottom)
```

7->12->1->5->4->2

What will be the output of the below Python function when it is invoked by passing the above stack? Assumption: All the references to the necessary files are available

```
def fun(stack1):
    if(stack1.is_empty()):
        return
    else:
        s1=Stack(4)
        while(not stack1.is_empty())
            s1.push(stack1.pop())
            s1.push(stack1.pop())
            s1.push(s1.pop()*s1.pop()
            return s1

Note: The options show the stack output from top

Warning
This operation is disabled.

Ok
```

84->8->5

5->84->8

8->5->84

O No change