

Started on Saturday, 4 October 2025, 10:14 AM

State Finished

Completed on Saturday, 4 October 2025, 10:54 AM

Time taken 40 mins 13 secs

Grade **80.00** out of 100.00

Question 1

Correct

Mark 20.00 out
of 20.00

A simple implementation of [Priority Queue](#) using Queue.

Complete the Insert Only

For example:

Input	Result
4	12 1 14 7
12	14
1	12
14	7
7	1

Answer: (penalty regime: 0 %)

Reset answer

```
1 # A simple implementation of Priority Queue
2 # using Queue.
3 class PriorityQueue(object):
4     def __init__(self):
5         self.queue = []
6
7     def __str__(self):
8         return ' '.join([str(i) for i in self.queue])
9
10    # for checking if the queue is empty
11    def isEmpty(self):
12        return len(self.queue) == 0
13
14    # for inserting an element in the queue
15    def insert(self, data):
16        self.queue.append(data)
17
18
19    # for popping an element based on Priority
20    def delete(self):
```

21
22

```
try:  
    max_val = 0
```

	Input	Expected	Got	
✓	4 12 1 14 7	12 1 14 7 14 12 7 1	12 1 14 7 14 12 7 1	✓
✓	5 13 6 27 4 2	13 6 27 4 2 27 13 6 4 2	13 6 27 4 2 27 13 6 4 2	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **2**

Correct

Mark 20.00 out
of 20.00

Write Python Program to take the radius from the user and find the area of the circle using class and function

For example:

Input	Result
5	Area of circle: 78.54

Answer: (penalty regime: 0 %)

```
1 import math
2 r=int(input())
3 area=(3.14159)*(r*r)
4 print("Area of circle: %0.2f"%area)
```

	Input	Expected	Got	
✓	5	Area of circle: 78.54	Area of circle: 78.54	✓
✓	10	Area of circle: 314.16	Area of circle: 314.16	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question 3

Incorrect

Mark 0.00 out of
20.00

Consider the following table of arrival time and burst time for five processes **P1, P2, P3, P4** and **P5**.

Processes	Arrival Time	Burst Time
P1	0	4
P2	1	3
P3	2	1
P4	3	2
P5	4	5

Waiting Time = Start time – Arrival time

$$\mathbf{P1} = 0 - 0 = 0$$

$$\mathbf{P2} = 4 - 1 = 3$$

$$\mathbf{P3} = 7 - 2 = 5$$

$$\mathbf{P4} = 8 - 3 = 5$$

$$\mathbf{P5} = 10 - 4 = 6$$

Average waiting time = $0 + 3 + 5 + 5 + 6 / 5 = 19 / 5 = 3.8$

Code a Python program to Calculate Waiting Time for given Processes.

For example:

Input	Result			
	P.No.	Arrival Time	Burst Time	Waiting Time
4	1	0	4	0
	2	1	3	3
	3	2	1	5
	4	3	2	5
	5	4	5	6
	Average waiting time = 3.8			

Answer: (penalty regime: 0 %)

Reset answer

```
1 # Python3 program to Calculate Waiting
2 # Time for given Processes
3
4 # Function to Calculate waiting time
5 # and average waiting time
6 def CalculateWaitingTime(at, bt, N):
7
8     # Declare the array for waiting
9     # time
10    wt = [0]*N;
11
12    # Waiting time for first process
13    # is 0
14    wt[0] = 0;
15
16    # Print waiting time process 1
17    print("P.No.\tArrival Time\t", "Burst Time\tWaiting Time")
18    print("1" , "\t\t" , at[0] , "\t\t" , bt[0] , "\t\t" , wt[0])
19
20    # Calculating waiting time for
21    # each process from the given
22
```

	Input	Expected	Got	
✗	4 3 1 2 5	P.No.\tArrival Time\t Burst Time\tWaiting Time 1 \t\t 0 \t\t 4 \t\t 0 2 \t\t 1 \t\t 3 \t\t 3 3 \t\t 2 \t\t 1 \t\t 5 4 \t\t 3 \t\t 2 \t\t 5 5 \t\t 4 \t\t 5 \t\t 6 Average waiting time = 3.8	P.No.\tArrival Time\t Burst Time\tWaiting Time 1 \t\t 0 \t\t 4 \t\t 0	✗

	Input	Expected	Got	
✖	5 4 1 2 3	P.No.\tArrival Time\t Burst Time\tWaiting Time 1 \t\t 0 \t\t 5 \t\t 0 2 \t\t 1 \t\t 4 \t\t 4 3 \t\t 2 \t\t 1 \t\t 7 4 \t\t 3 \t\t 2 \t\t 7 5 \t\t 4 \t\t 3 \t\t 8 Average waiting time = 5.2	P.No.\tArrival Time\t Burst Time\tWaiting Time 1 \t\t 0 \t\t 5 \t\t 0	✖

Your code must pass all tests to earn any marks. Try again.

Show differences

Incorrect

Marks for this submission: 0.00/20.00.

Question 4

Correct

Mark 20.00 out
of 20.00

Write a python function to insert, delete and display integer element from the [circular queue](#).

For example:

Input	Result
3	[17, 23, 45]
17	[23, 45]
23	
45	

Answer: (penalty regime: 0 %)

Reset answer

```
1 ▾ class Queue:  
2 ▾     def __init__(self,limit):  
3         self.queue=[]  
4         self.rear=0  
5         self.front=0  
6         self.limit=limit  
7 ▾     def isempty(self):  
8 ▾         if len(self.queue)==0:  
9             return True  
10 ▾        else:  
11            return False  
12 ▾     def enqueue(self,item):  
13 ▾         if len(self.queue)==self.limit:  
14             print("The queue is full")  
15 ▾         else:  
16             if self.front==self.limit:  
17                 self.front=self.rear-1  
18             self.queue.insert(self.front,item)  
19             self.front+=1  
20  
21 ▾     def dequeue(self):  
22 ▾         if len(self.queue)==0:
```

	Input	Expected	Got	
✓	3 17 23 45	[17, 23, 45] [23, 45]	[17, 23, 45] [23, 45]	✓

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 insert 14, 15 at **FRONT END** of deque using collection built-in function.

For example:

Input	Result
11	The deque after appending is :
12	deque([15, 14, 11, 12, 13])
13	

Answer: (penalty regime: 0 %)

```
1 import collections
2 n1=int(input())
3 n2=int(input())
4 n3=int(input())
5 de=collections.deque([n1,n2,n3])
6 de.appendleft(14)
7 de.appendleft(15)
8 print("The deque after appending is :")
9 print(de)
```

	Input	Expected	Got	
✓	11 12 13	The deque after appending is : deque([15, 14, 11, 12, 13])	The deque after appending is : deque([15, 14, 11, 12, 13])	✓
✓	23 76 32	The deque after appending is : deque([15, 14, 23, 76, 32])	The deque after appending is : deque([15, 14, 23, 76, 32])	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.