Started on	Monday, 19 May 2025, 1:17 PM
State	Finished
Completed on	Monday, 19 May 2025, 1:34 PM
Time taken	16 mins 53 secs
Grade	**80.00** out of 100.00

Question **1**

Incorrect

Mark 0.00 out of 20.00

Write a python program to compute the percentage and class of students given the average of six subject marks . The Maximum mark for each subject is 100.

[above 70% - First Class with Distinction
60-70% - First Class
50-60% - Second Class
35-50% - Passed
below 35% - Failed]

For example:

Input	Result
490	You have scored 81.67% of marks First Class with Distinction

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
def score(marks):
    k=marks/6
    print(f"You have scored {k:.2f}% of marks")
    if k>70:
        print("First Class with Distinction")
    elif ( k>=60) and k>=70:
        print("First Division")
    elif k>=50 and k<60:
        print("Second Division")
    elif k>=35 and k<50:
        print("Passed")
    elif k<35:
        print("Failed")
marks=int(input())
score(marks)
```

	Input	Expected	Got	
✔	490	You have scored 81.67% of marks First Class with Distinction	You have scored 81.67% of marks First Class with Distinction	✔
✔	350	You have scored 58.33% of marks Second Division	You have scored 58.33% of marks Second Division	✔
✔	280	You have scored 46.67% of marks Passed	You have scored 46.67% of marks Passed	✔
✔	120	You have scored 20.00% of marks Failed	You have scored 20.00% of marks Failed	✔
✖	360	You have scored 60.00% of marks First Division	You have scored 60.00% of marks	✖

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

Show differences

Incorrect

Marks for this submission: 0.00/20.00.

Question 2

Correct

Mark 20.00 out of 20.00

After inserting few elements in stack, check and display whether the stack is full or not.

Answer: (penalty regime: 0 %)

Reset answer

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
from queue import Queue
queue = Queue(maxsize = 4)

queue.put('a')
queue.put('b')
queue.put('c')
if (queue.full()):
    print("Stack is full")
else:
    print("Stack is not full")
```

	Expected	Got	
✔	`Stack is not full`	`Stack is not full`	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **3**

Correct

Mark 20.00 out of 20.00

Type a python code to insert 3 elements. Also check and print the index value of the elements stored in the stack.

Answer: (penalty regime: 0 %)

Reset answer

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
stack = []
stack.append('a')
stack.append('b')
stack.append('c')
print('Initial stack: ' + str(stack))
for i in range(len(stack)):
    print(i,stack[i])
```

	Expected	Got	

✔	`Initial stack: ['a', 'b', 'c']`
	`0 a`
	`1 b`
	`2 c`
		`Initial stack: ['a', 'b', 'c']`
		`0 a`
		`1 b`
		`2 c`
			✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **4**

Correct

Mark 20.00 out of 20.00

List out the slots allotted for all the candidates who came for the interview.

Answer: (penalty regime: 0 %)

Reset answer

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
interview = ['Ram', 'Siva', 'Joseph', 'Ijaz', 'Sasi']

print(interview)

print("Slot numbers are: ")
for i in range(len(interview)):
    print("[Slot number: "+str(i+1)+", "+interview[i]+"]")
```

	Expected	Got	
✔	['Ram', 'Siva', 'Joseph', 'Ijaz', 'Sasi'] Slot numbers are: [Slot number: 1, Ram] [Slot number: 2, Siva] [Slot number: 3, Joseph] [Slot number: 4, Ijaz] [Slot number: 5, Sasi]	['Ram', 'Siva', 'Joseph', 'Ijaz', 'Sasi'] Slot numbers are: [Slot number: 1, Ram] [Slot number: 2, Siva] [Slot number: 3, Joseph] [Slot number: 4, Ijaz] [Slot number: 5, Sasi]	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **5**

Correct

Mark 20.00 out of 20.00

Type a python code to add 4 elements in a queue.

Print the element present in the front and rear of queue.

Answer: (penalty regime: 0 %)

Reset answer

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
queue = []

queue.append('a')
queue.append('b')
queue.append('c')
queue.append('d')

print('Initial Queue: ' + str(queue))
print("\nElement at the front of the queue is .... ",queue[0])
print("\nElement at the rear of the queue is .... ",queue[3])
```

	Expected	Got	
✔	Initial Queue: ['a', 'b', 'c', 'd'] Element at the front of the queue is .... a Element at the rear of the queue is .... d	Initial Queue: ['a', 'b', 'c', 'd'] Element at the front of the queue is .... a Element at the rear of the queue is .... d	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.