Started on	Friday, 31 October 2025, 3:25 PM
State	Finished
Completed on	Friday, 31 October 2025, 3:56 PM
Time taken	30 mins 52 secs
Grade	**80.00** out of 100.00

Question **1**

Not answered

Mark 0.00 out of 20.00

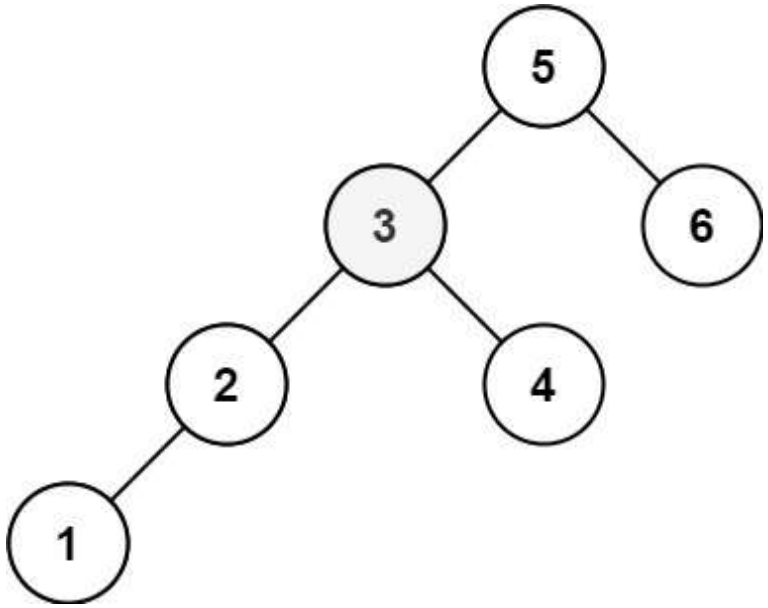
Write a Python program to build a binary search tree using a built in function

1. Use appropriate module to build a binary tree

2. **def _build_bst_from_sorted_values(sorted_values):** - to build a binary search tree for the below given tree.

3. **def insert_BST(val):** - to insert a node value and rebuilt a BST , get the value to be inserted from the user and pass it to the function.

4. **def display(T):** - to display the values of the tree nodes.

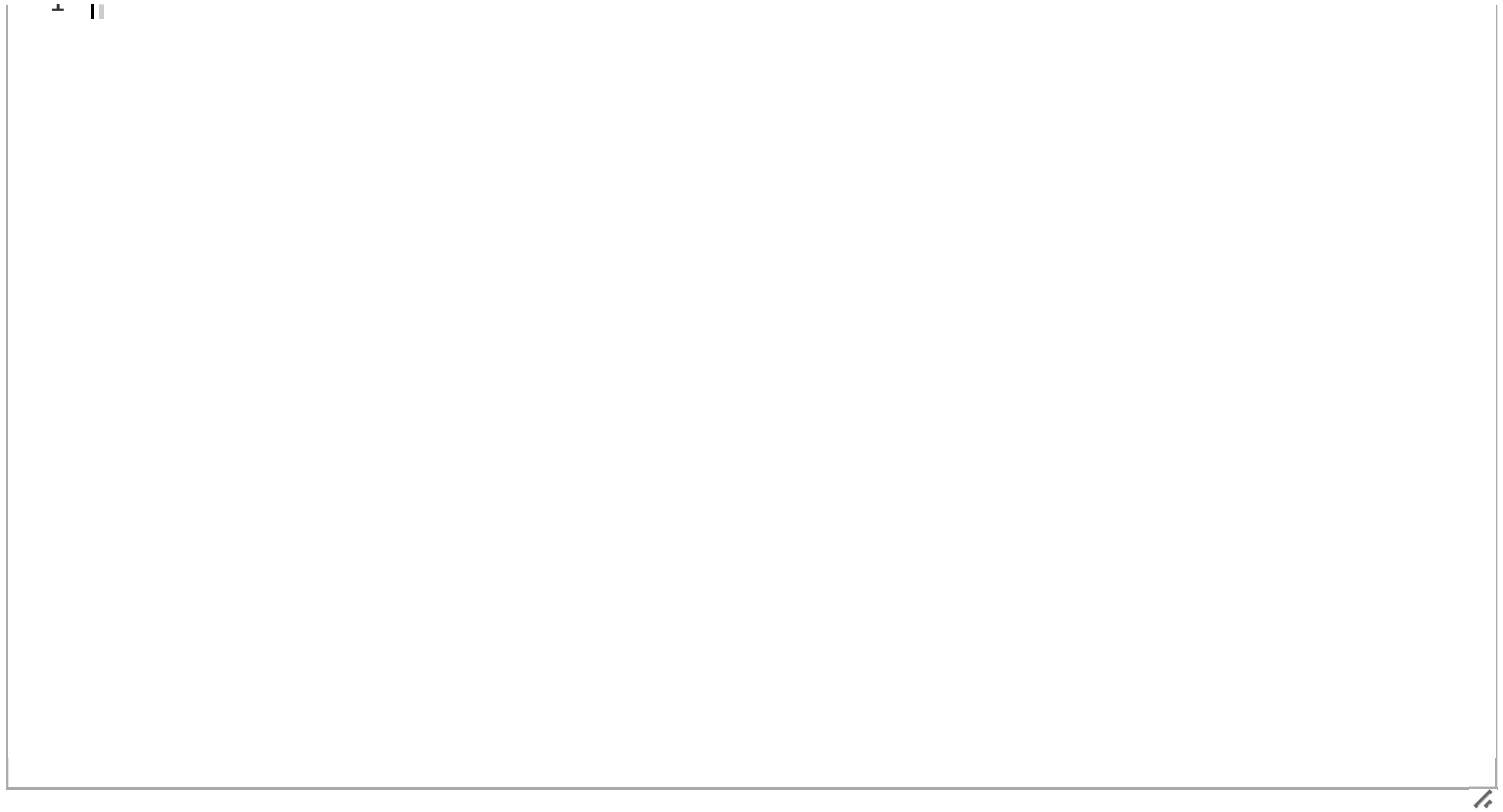


For example:

Input	Result
10	BST before insertion: 4 -->2 -->6 -->1 -->3 -->5 -->BST after insertion: 4 -->2 -->6 -->1 -->3 -->5 -->10 -->

Answer: (penalty regime: 0 %)

```
1
```



Question **2**

Correct

Mark 20.00 out of 20.00

Write a Python program to print a binary tree consist of a root, left and right node.

1. Use appropriate Python library to build a Binary tree using Node function

2. Get the values of the nodes from the user and save it in a list.

3. Print the node values using the build in function

For example:

Input	Result
S A Y	Binary Tree : S --> A --> Y -->

Answer: (penalty regime: 0 %)

Reset answer

```
from binarytree import build
l=[]
for i in range(3):
    a=input()
    l.append(a)
print("Binary Tree : ")
for i in range(3):
    print(l[i],"--> ",end="")
```

	Input	Expected	Got	
✔	S A Y	Binary Tree : S --> A --> Y -->	Binary Tree : S --> A --> Y -->	✔
✔	W I N	Binary Tree : W --> I --> N -->	Binary Tree : W --> I --> N -->	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **3**

Correct

Mark 20.00 out of 20.00

Write a function which takes two arguments: a and b and returns the addition of them: a+b. Assign it to a variable named: f. using python

For example:

Input	Result
5 10	15

Answer: (penalty regime: 0 %)

```
a=int(input())
b=int(input())
c=a+b
print(c)
```

	Input	Expected	Got	
✔	5 10	15	15	✔
✔	11 10	21	21	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

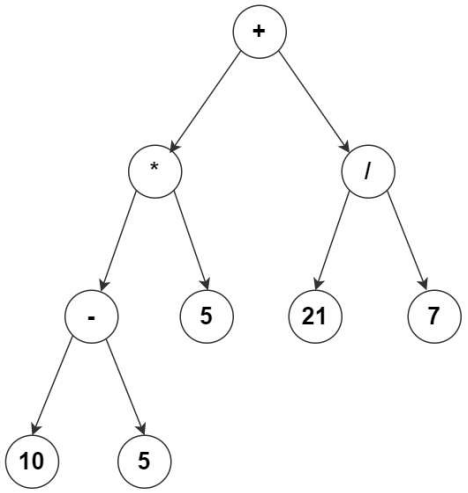
Question **4**

Correct

Mark 20.00 out of 20.00

Write a python program to build and evaluate the below given Expression tree.

Define def evaluate(root): - to evaluate the expression tree using recursion



For example:

Test	Result
`evaluate(root)`	`The value of the expression tree is 28.0`

Answer: (penalty regime: 0 %)

Reset answer

```
class Node:
    def __init__(self, val, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right

def isLeaf(node):
    return node.left is None and node.right is None

def process(op, x, y):
    if op == '+':
```

```
12          return x + y
13      if op == '-':
14          return x - y
15      if op == '*':
16          return x * y
17      if op == '/':
18          return x / y
19
20  def evaluate(root):
21      if root is None:
22          return 0
```

	Test	Expected	Got	
✔	evaluate(root)	The value of the expression tree is 28.0	The value of the expression tree is 28.0	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **5**

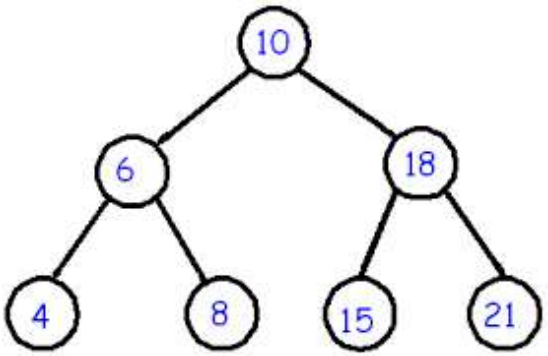
Correct

Mark 20.00 out of 20.00

Write a python program to implement push and pop an element into the below given Heap tree using a single built-in function.

1. Use appropriate Python package and functions.

2. Get the value to be modified from the user and insert it.



For example:

Input	Result
5	The created heap is : [4, 6, 15, 10, 8, 18, 21] Heap after pushpop operation: [5, 6, 15, 10, 8, 18, 21]
11	The created heap is : [4, 6, 15, 10, 8, 18, 21] Heap after pushpop operation: [6, 8, 15, 10, 11, 18, 21]

Answer: (penalty regime: 0 %)

Reset answer

```
import heapq
l=[4,6,15,10,8,18,21]
heapq.heapify
print("The created heap is : ",l)
a=int(input())
heapq.heappushpop(l,a)
print("Heap after pushpop operation: ",l)
```

```
8 |# Write your code here
```

	Input	Expected	Got	
✔	5	The created heap is : [4, 6, 15, 10, 8, 18, 21] Heap after pushpop operation: [5, 6, 15, 10, 8, 18, 21]	The created heap is : [4, 6, 15, 10, 8, 18, 21] Heap after pushpop operation: [5, 6, 15, 10, 8, 18, 21]	✔
✔	11	The created heap is : [4, 6, 15, 10, 8, 18, 21] Heap after pushpop operation: [6, 8, 15, 10, 11, 18, 21]	The created heap is : [4, 6, 15, 10, 8, 18, 21] Heap after pushpop operation: [6, 8, 15, 10, 11, 18, 21]	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.