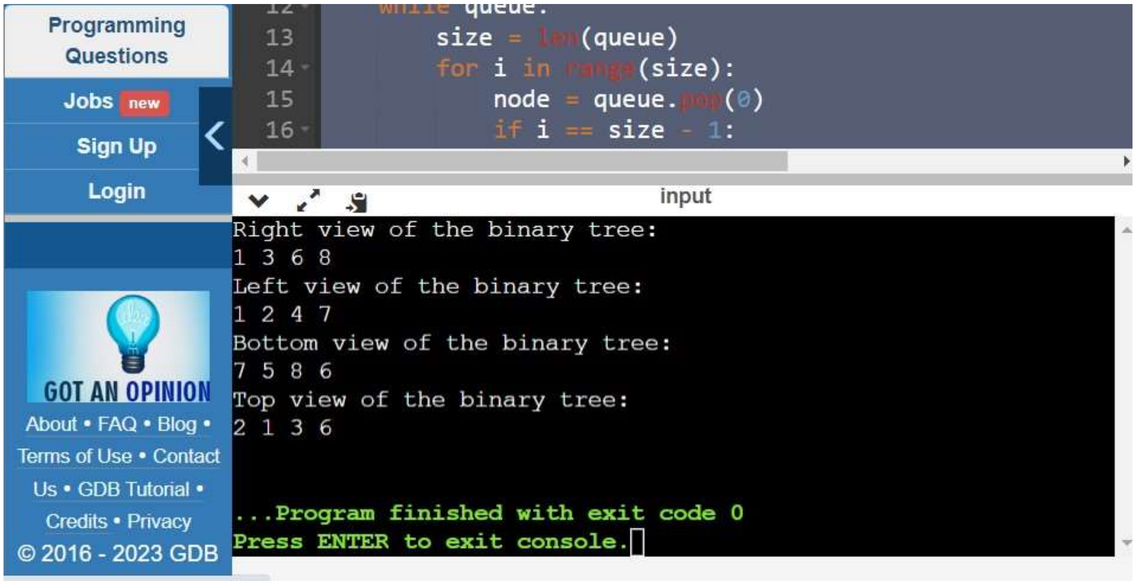
Problem 6,7,8,9: Given a Binary Tree, find the

1. Right view
2. Left view
3. Top view
4. Bottom view

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


def printRightView(root):
    if root is None:
        return

    queue = [root]
    while queue:
        size = len(queue)
        for i in range(size):
            node = queue.pop(0)
            if i == size - 1:
                print(node.data, end=" ")
            if node.left:
                queue.append(node.left)
            if node.right:
                queue.append(node.right)
    print()


def printLeftView(root):
    if root is None:
        return

    queue = [root]
    while queue:
        size = len(queue)
        for i in range(size):
            node = queue.pop(0)
            if i == 0:
                print(node.data, end=" ")
            if node.left:
                queue.append(node.left)
            if node.right:
                queue.append(node.right)
    print()

def printBottomView(root):
    if root is None:
        return

    queue = [(root, 0)]
    bottom_view = {}
    while queue:
        node, hd = queue.pop(0)
        bottom_view[hd] = node.data
        if node.left:
            queue.append((node.left, hd - 1))
        if node.right:
            queue.append((node.right, hd + 1))

    for hd in sorted(bottom_view.keys()):
        print(bottom_view[hd], end=" ")
    print()

def printTopView(root):
    if root is None:
        return

    queue = [(root, 0)]
    top_view = {}
    while queue:
        node, hd = queue.pop(0)
        if hd not in top_view:
            top_view[hd] = node.data
        if node.left:
            queue.append((node.left, hd - 1))
        if node.right:
            queue.append((node.right, hd + 1))

    for hd in sorted(top_view.keys()):
        print(top_view[hd], end=" ")
    print()

root = Node(1)
root.left = Node(2)
root.right = Node(3)
root.left.right = Node(4)
root.right.left = Node(5)
root.right.right = Node(6)
root.right.left.left = Node(7)
root.right.left.right = Node(8)

print("Right view of the binary tree:")
printRightView(root)

print("Left view of the binary tree:")
printLeftView(root)

print("Bottom view of the binary tree:")
printBottomView(root)
```

```
print("Top view of the binary tree:")

printTopView(root)
```



Programming Questions
Jobs new
Sign Up
Login
GOT AN OPINION
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2023 GDB
input

```
    while queue:
        size = len(queue)
        for i in range(size):
            node = queue.pop(0)
            if i == size - 1:
```

```
Right view of the binary tree:
1 3 6 8
Left view of the binary tree:
1 2 4 7
Bottom view of the binary tree:
7 5 8 6
Top view of the binary tree:
2 1 3 6


...Program finished with exit code 0
Press ENTER to exit console.
```