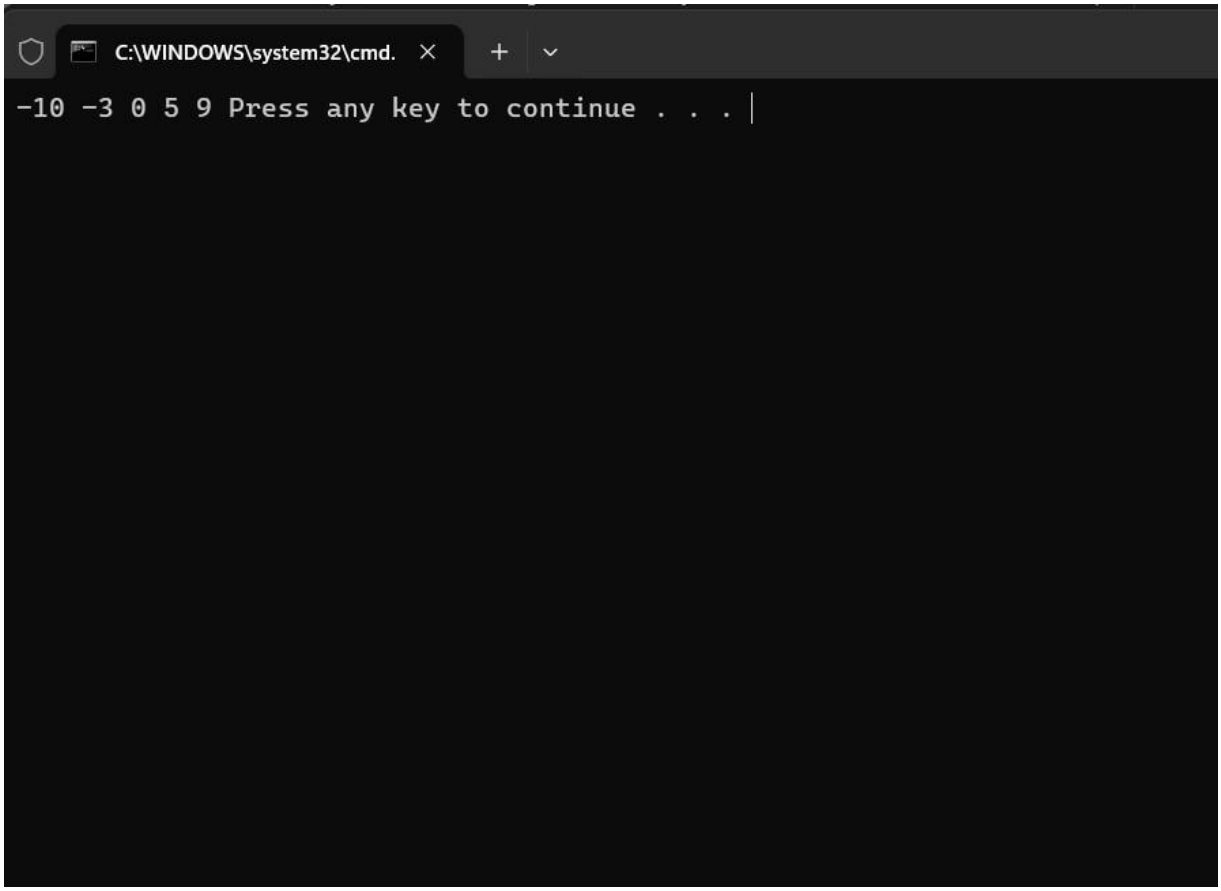
49) Convert Sorted Array to Binary Search Tree Given an integer array nums where the elements are sorted in ascending order, convert it to a height-balanced binary search tree.

CODE:

```
class TreeNode:     def __init__(self, val=0,
left=None, right=None):         self.val = val
self.left = left         self.right = right
 def
sortedArrayToBST(nums)
:     if not nums:
return None

    mid = len(nums) // 2

    # Create the root node with the middle element
root = TreeNode(nums[mid])

    # Recursively construct the left and right subtrees
root.left = sortedArrayToBST(nums[:mid])
root.right = sortedArrayToBST(nums[mid+1:])

    return root
 def inorderTraversal(root):     if
root:
inorderTraversal(root.left)
print(root.val, end=" ")
inorderTraversal(root.right)

# Example usage: nums = [-
10, -3, 0, 5, 9] root =
sortedArrayToBST(nums)
inorderTraversal(root)
```

OUTPUT:



TIME COMPLEXITY : O(n)