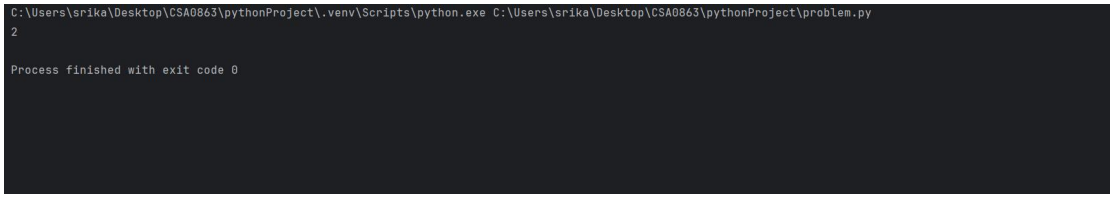
Q) ..Given two sorted arrays nums1 and nums2 of size m and n respectively, return the median of the two sorted arrays. The overall run time complexity should be O(log (m+n)).

Program:

```
def findMedianSortedArrays(nums1, nums2):
    if len(nums1) > len(nums2):
        nums1, nums2 = nums2, nums1
    m, n = len(nums1), len(nums2)
    total_length = m + n
    left, right = 0, m
    while left <= right:
        partition1 = (left + right) // 2
        partition2 = (total_length + 1) // 2 -
partition1
        max_left1 = float('-inf') if partition1 == 0
else nums1[partition1 - 1]
        min_right1 = float('inf') if partition1 == m
else nums1[partition1]
        max_left2 = float('-inf') if partition2 == 0
else nums2[partition2 - 1]
        min_right2 = float('inf') if partition2 == n
else nums2[partition2]
        if max_left1 <= min_right2 and max_left2
<= min_right1:
            if total_length % 2 == 0:
```

```
            return (max(max_left1, max_left2) +
min(min_right1, min_right2)) / 2
        else:
            return max(max_left1, max_left2)
    elif max_left1 > min_right2:
        right = partition1 - 1
    else:
        left = partition1 + 1
nums1 = [1, 3]
nums2 = [2]
print(findMedianSortedArrays(nums1, nums2))
```

Output:



Time complexity:O(log(min(m,n)))