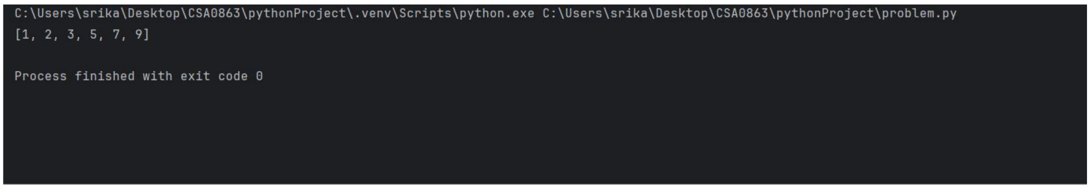
6.Given an array of integers nums, sort the array in ascending order and return it.You must solve the problem without using any built-in functions in O(nlog(n)) time complexity and with the smallest space complexity possible.

Program:

```
def merge_sort(nums):
    if len(nums) <= 1:
        return nums
    mid = len(nums) // 2
    left_half = nums[:mid]
    right_half = nums[mid:]
    merge_sort(left_half)
    merge_sort(right_half)
    i = j = k = 0
    while i < len(left_half) and j < len(right_half):
        if left_half[i] < right_half[j]:
            nums[k] = left_half[i]
            i += 1
        else:
            nums[k] = right_half[j]
            j += 1
        k += 1

    while i < len(left_half):
        nums[k] = left_half[i]
        i += 1
        k += 1
    while j < len(right_half):
        nums[k] = right_half[j]
        j += 1
        k += 1
def sort_array(nums):
    merge_sort(nums)
    return nums
nums = [5, 2, 9, 3, 7, 1]
sorted_nums = sort_array(nums)
print(sorted_nums)
```

Output:



```
C:\Users\srika\Desktop\CSA0863\pythonProject\.venv\Scripts\python.exe C:\Users\srika\Desktop\CSA0863\pythonProject\problem.py
[1, 2, 3, 5, 7, 9]

Process finished with exit code 0
```

Time complexity:O(log(n))