

**18. 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  $O(n \log(n))$  time complexity and with the smallest space complexity possible.**

**Program:**

```
def merge_sort(arr):
    if len(arr) <= 1:
        return arr

    mid = len(arr) // 2
    left = merge_sort(arr[:mid])
    right = merge_sort(arr[mid:])

    return merge(left, right)

def merge(left, right):
    result = []
    i = j = 0

    while i < len(left) and j < len(right):
        if left[i] < right[j]:
            result.append(left[i])
            i += 1
        else:
            result.append(right[j])
```

```
j += 1
```

```
result.extend(left[i:])
```

```
result.extend(right[j:])
```

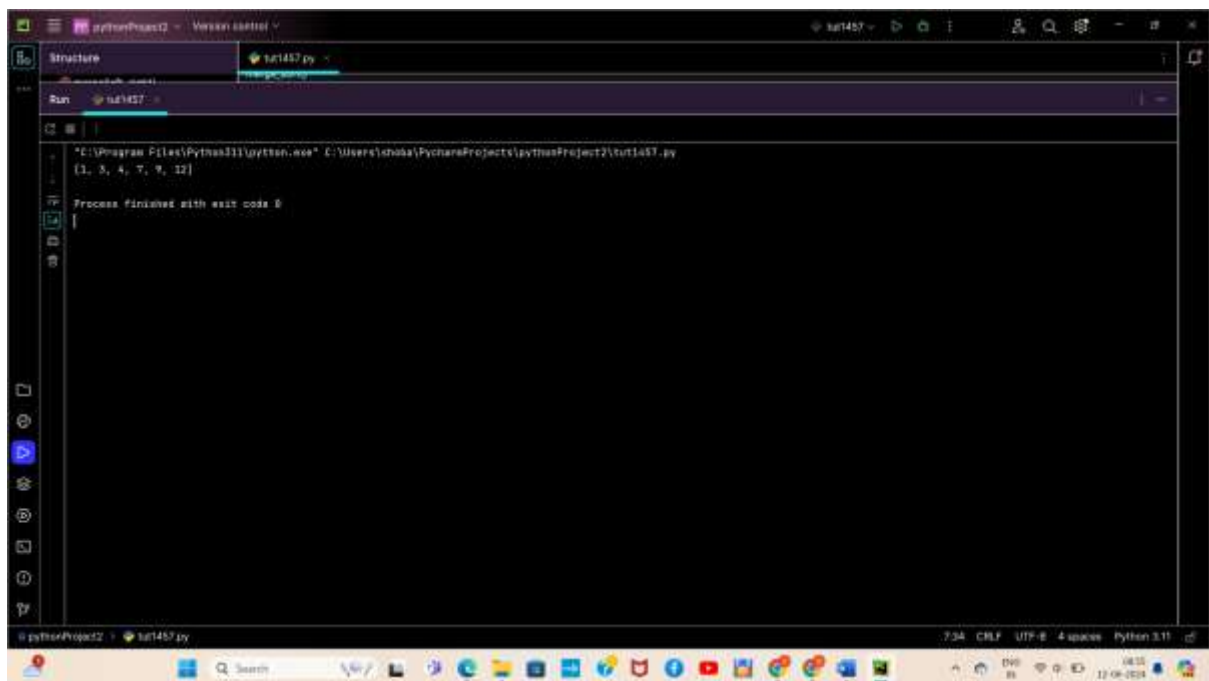
```
return result
```

```
nums = [12, 4, 7, 1, 9, 3]
```

```
sorted_nums = merge_sort(nums)
```

```
print(sorted_nums)
```

Output:



The screenshot shows a Python IDE window titled 'pythonProject2 - Visual Studio Code'. The 'Run' panel is active, displaying the output of the program. The output shows the sorted list [1, 3, 4, 7, 9, 12] and a message 'Process finished with exit code 0'. The file 'tut1457.py' is open in the editor, showing the code for the merge sort algorithm. The status bar at the bottom indicates the file is 'tut1457.py' and the Python version is 'Python 3.11'.

```
"C:\Program Files\Python311\python.exe" C:\Users\chaba\PycharmProjects\pythonProject2\tut1457.py
[1, 3, 4, 7, 9, 12]
Process finished with exit code 0
```