

```

# Function to print the array
def print_arr(arr):
    for i in arr:
        print(i, end=" ")
    print()

# Function to perform the QuickSort algorithm
def quick_sort(arr, si, ei):
    # Base case: if the starting index is greater or equal to the
    # ending index
    if si >= ei:
        return

    # Partition the array and get the pivot index
    p_idx = partition(arr, si, ei)

    # Recursively sort elements before and after partition
    quick_sort(arr, si, p_idx - 1) # Sort the left half
    quick_sort(arr, p_idx + 1, ei) # Sort the right half

# Function to partition the array around a pivot
def partition(arr, si, ei):
    pivot = arr[ei] # Choose the last element as the pivot
    i = si - 1      # Initialize index for elements smaller than
    # pivot

    # Traverse through the array, rearranging elements based on the
    # pivot
    for j in range(si, ei):
        if arr[j] <= pivot:
            i += 1
            # Swap arr[i] and arr[j]
            temp = arr[i]
            arr[i] = arr[j]
            arr[j] = temp

    # Place the pivot element in its correct position
    i += 1
    temp = arr[ei]
    arr[ei] = arr[i]
    arr[i] = temp

    return i # Return the index of the pivot element

# Main function
if __name__ == "__main__":
    arr = [6, 3, 9, 8, 2, 5] # Initialize array
    quick_sort(arr, 0, len(arr) - 1) # Sort the array
    print_arr(arr) # Print the sorted array

```

2 3 5 6 8 9