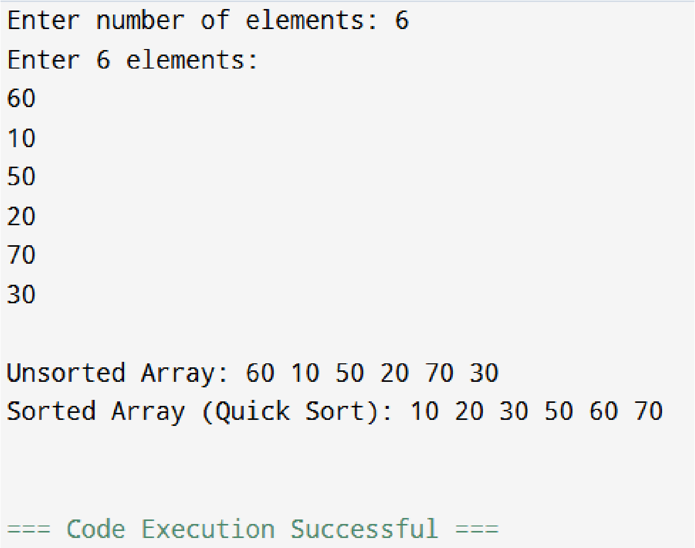
Experiment 25: Quick Sort

Code:

```
#include <stdio.h>
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
int partition(int arr[], int low, int high) {
    int pivot = arr[high];
    int i = (low - 1);
    for (int j = low; j < high; j++) {
        if (arr[j] < pivot) {
            i++;
            swap(&arr[i], &arr[j]);
        }
    }
    swap(&arr[i + 1], &arr[high]);
    return (i + 1);
}
void quickSort(int arr[], int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high);
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}
```

```
void printArray(int arr[], int n) {
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
}
int main() {
    int arr[50], n;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    printf("\nUnsorted Array: ");
    printArray(arr, n);
    quickSort(arr, 0, n - 1);
    printf("Sorted Array (Quick Sort): ");
    printArray(arr, n);
    return 0;
}
```

Output:



```
Enter number of elements: 6
Enter 6 elements:
60
10
50
20
70
30

Unsorted Array: 60 10 50 20 70 30
Sorted Array (Quick Sort): 10 20 30 50 60 70


=== Code Execution Successful ===
```