

Divide and conquer

Program – 3

### **Aim:**

To implement the Quick Sort algorithm to sort a list of elements.

---

### **Input:**

- First line: An integer nnn – the number of elements in the list.
- Next nnn lines: nnn integers – the elements of the list.

### **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);  
    }  
}
```

```
int main() {  
    int n;  
  
    scanf("%d", &n);  
  
    int arr[n];  
  
    for (int i = 0; i < n; i++) {  
        scanf("%d", &arr[i]);
```

```
}
```

```
quickSort(arr, 0, n - 1);
```

```
for (int i = 0; i < n; i++) {
```

```
    printf("%d ", arr[i]);
```

```
}
```

```
return 0;
```

```
}
```

Output:

	Input	Expected	Got	
✓	5 67 34 12 98 78	12 34 67 78 98	12 34 67 78 98	✓
✓	10 1 56 78 90 32 56 11 10 90 114	1 10 11 32 56 56 78 90 90 114	1 10 11 32 56 56 78 90 90 114	✓
✓	12 9 8 7 6 5 4 3 2 1 10 11 90	1 2 3 4 5 6 7 8 9 10 11 90	1 2 3 4 5 6 7 8 9 10 11 90	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.