

## Quick Sort ADA Lab

```
#include <stdio.h>
#include <stdlib.h>
int partition(int A[], int low, int high)
{
    int i, j, pivot, temp;
    i = low + 1;
    pivot = A[low];
    j = high;
    while (i <= j)
    {
        while (A[i] <= pivot)
            i++;
        while (A[j] > pivot)
            j--;

        if (i < j)
        {
            temp = A[i];
            A[i] = A[j];
            A[j] = temp;
        }
    }
    temp = A[low];
    A[low] = A[j];
    A[j] = temp;
    return j;
}
```

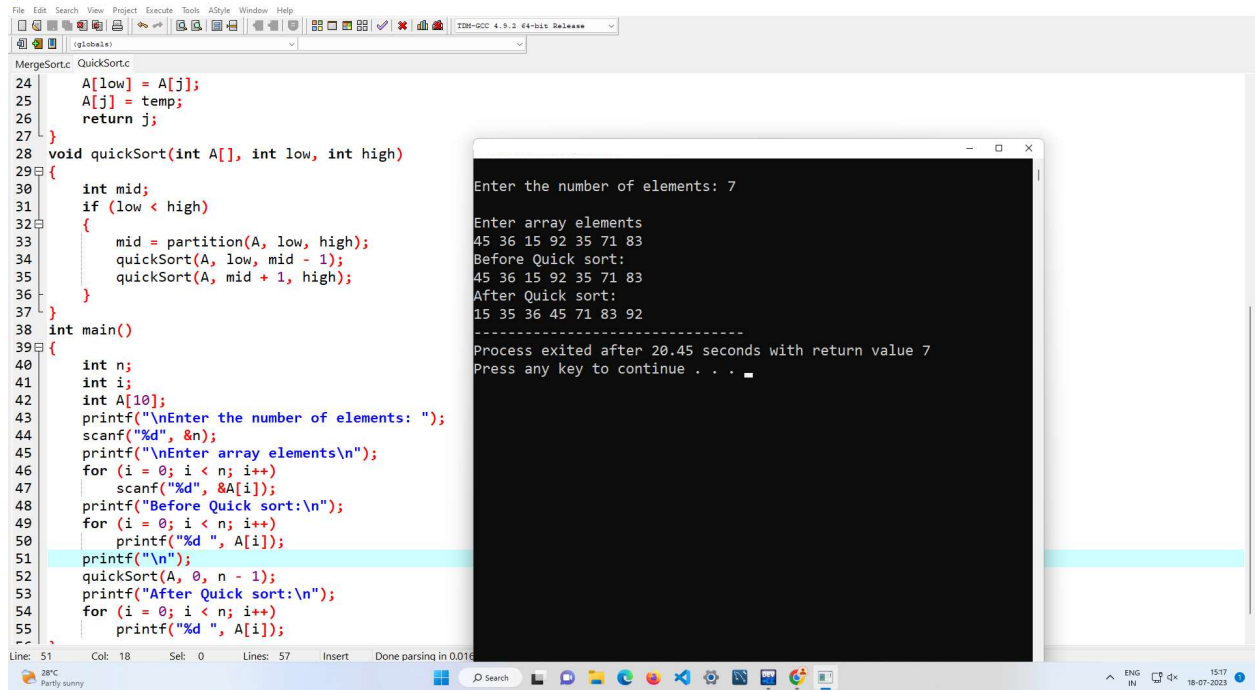
```

void quickSort(int A[], int low, int high)
{
    int mid;
    if (low < high)
    {
        mid = partition(A, low, high);
        quickSort(A, low, mid - 1);
        quickSort(A, mid + 1, high);
    }
}

int main()
{
    int n;
    int i;
    int A[10];
    printf("\nEnter the number of elements: ");
    scanf("%d", &n);
    printf("\nEnter array elements\n");
    for (i = 0; i < n; i++)
        scanf("%d", &A[i]);
    printf("Before Quick sort:\n");
    for (i = 0; i < n; i++)
        printf("%d ", A[i]);
    printf("\n");
    quickSort(A, 0, n - 1);
    printf("After Quick sort:\n");
    for (i = 0; i < n; i++)
        printf("%d ", A[i]);
}

```

# Output:



The image shows a screenshot of a C++ IDE with two windows. The background window is a code editor showing the implementation of MergeSort and QuickSort algorithms. The foreground window is a terminal displaying the program's execution output.

**Code Editor Content:**

```
24     A[low] = A[j];
25     A[j] = temp;
26     return j;
27 }
28 void quickSort(int A[], int low, int high)
29 {
30     int mid;
31     if (low < high)
32     {
33         mid = partition(A, low, high);
34         quickSort(A, low, mid - 1);
35         quickSort(A, mid + 1, high);
36     }
37 }
38 int main()
39 {
40     int n;
41     int i;
42     int A[10];
43     printf("\nEnter the number of elements: ");
44     scanf("%d", &n);
45     printf("\nEnter array elements\n");
46     for (i = 0; i < n; i++)
47         scanf("%d", &A[i]);
48     printf("Before Quick sort:\n");
49     for (i = 0; i < n; i++)
50         printf("%d ", A[i]);
51     printf("\n");
52     quickSort(A, 0, n - 1);
53     printf("After Quick sort:\n");
54     for (i = 0; i < n; i++)
55         printf("%d ", A[i]);
56 }
```

**Terminal Output:**

```
Enter the number of elements: 7
Enter array elements
45 36 15 92 35 71 83
Before Quick sort:
45 36 15 92 35 71 83
After Quick sort:
15 35 36 45 71 83 92
-----
Process exited after 20.45 seconds with return value 7
Press any key to continue . . .
```