

**PRIYANSHI**

**E – 26**

**2016927**

## **WEEK - 4**

**Q1.**

```
#include <iostream> using
namespace std;int comp = 0;
int merge(int arr[], int l, int mid, int h) { int count =
    0;
    int i = l, j = mid + 1; int
    temp[1000];
    int k = 0;
    while (i <= mid && j <= h) { if (arr[i] <
        arr[j])
            temp[k++] = arr[i++];else {
            temp[k++] = arr[j++]; count +=
            mid - i + 1;
        }
        comp++;
    }
    for (; i <= mid;) temp[k++] =
        arr[i++];
    for (; j <= h;)
        temp[k++] = arr[j++];for (int f =
    0; f < k; f++)
        arr[f + l] = temp[f];return
    count;
}
int merge_sort(int arr[], int l, int h)
{
    int inversion = 0;
```

```

    if (l < h)
    {
        int mid = l + (h - l) / 2;
        inversion += merge_sort(arr, l, mid);
        inversion += merge_sort(arr, mid + 1, h);
        inversion += merge(arr, l, mid, h);
    }
    return inversion;
}

int main()
{
    int T;
    cin >> T;
    while (T--)
    {
        int n;
        cin >> n;
        int arr[1000];
        int prev_comp = comp;
        for (int i = 0; i < n; i++)
            cin >> arr[i];
        int inv = merge_sort(arr, 0, n - 1);
        for (int i = 0; i < n; i++)
            cout << arr[i] << " ";
        cout << endl
            << "comparisons = " << comp - prev_comp << endl
            << "inversions = " << inv << endl;
    }
    return 0;
}

```

## OUTPUT:

```
3
8
23 65 21 76 46 89 45 32
21 23 32 45 46 65 76 89
comparisons = 16
inversions = 13
10
54 65 34 76 78 97 46 32 51 21
21 32 34 46 51 54 65 76 78 97
comparisons = 22
inversions = 28
15
63 42 223 645 652 31 324 22 553 12 54 65 86 46 325
12 22 31 42 46 54 63 65 86 223 324 325 553 645 652
comparisons = 43
inversions = 54

...Program finished with exit code 0
Press ENTER to exit console.□
```

**2.**

```
#include <stdio.h>
```

```
void swap(int *x, int *y)
```

```
{
```

```
    int temp = *x;
```

```
    *x = *y;
```

```
    *y = temp;
```

```
}
```

```
int comp=0,swaps=0;
```

```
int partition(int array[], int low, int high) {
```

```
    int pivot = array[high];
```

```
    int i = (low - 1);
```

```
    for (int j = low; j < high; j++) {
```

```
        comp++;
```

```
        if (array[j] <= pivot) {
```

```
            i++;
```

```
            swap(&array[i], &array[j]);
```

```
            swaps++;
```

```
        }
```

```
    }
```

```
    swaps++;
```

```
    swap(&array[i + 1], &array[high]);
```

```
    return (i + 1);
```

```
}
```

```
void quickSort(int array[], int low, int high) {
```

```
if (low < high) {  
    int pi = partition(array, low, high);  
    quickSort(array, low, pi - 1);  
    quickSort(array, pi + 1, high);  
}  
}
```

```
int main()  
{  
    int t;  
    scanf("%d",&t);  
    while (t--)  
    {  
        int n;  
        scanf("%d",&n);  
        int arr[n];  
        for (int i=0;i<n;i++)  
            scanf("%d",&arr[i]);  
        swaps=0;  
        comp=0;  
        quickSort(arr,0,n-1);  
  
        for (int i=0;i<n;i++)  
            printf("%d",arr[i]);  
        printf("Swaps:%d\n",swaps);  
        printf("Comparisions:%d\n",comp);  
    }  
    return 0;  
}
```

## OUTPUT:

```
/tmp/eSLC9E00Ld.o
```

```
1
```

```
8
```

```
23 65 21 76 46 89 45 32
```

```
23 65 21 76 46 89 45 32
```

```
2123324546657689Swaps:10
```

```
Comparisions:14
```

**Q3.**

```
#include <iostream>

using namespace std;

void merge(int arr[], int l, int mid, int h)
{
    int count = 0;
    int i = l, j = mid + 1;
    int temp[1000];
    int k = 0;
    while (i <= mid && j <= h)
    {
        if (arr[i] < arr[j])
            temp[k++] = arr[i++];
        else
        {
            temp[k++] = arr[j++];
            count += mid - i + 1;
        }
    }
    for (; i <= mid;)
        temp[k++] = arr[i++];

    for (; j <= h;)
        temp[k++] = arr[j++];

    for (int f = 0; f < k; f++)
        arr[f + l] = temp[f];
}

void merge_sort(int arr[], int l, int h)
{

```

```
    if (l < h)
    {
        int mid = l + (h - l) / 2;
        merge_sort(arr, l, mid);
        merge_sort(arr, mid + 1, h);
        merge(arr, l, mid, h);
    }
}
```

```
int main()
{
    int T;
    cin >> T;
    while (T--)
    {
        int n;
        cin >> n;
        int arr[1000];
        for (int i = 0; i < n; i++)
            cin >> arr[i];
        int k;
        cin >> k;
        merge_sort(arr, 0, n - 1);
        int flag = 0;
        cout << arr[k - 1] << endl;
    }
}
```



## OUTPUT:

```
2
10
123 656 54 765 344 514 765 34 765 234
3
123
15
43 64 13 78 864 346 786 456 21 19 8 434 76 270 601
8
78

...Program finished with exit code 0
Press ENTER to exit console.□
```