DSA_College\QuickSort.cpp

```cpp
#include<iostream>
using namespace std ;

int partition(int*arr, int s, int e)
{
    int start = s ;  // start variable rkh lo arr[0] or s ki position pr
    int end = e ;    // end variable rkh lo last index pr
    int pivot = arr[s] ; // first wale element ko pivot maan lo

    while(start <= end)  // jab tak start<end hai tab tak chlo
    {
        while(arr[start] <= pivot) // agar pivot se bda element milta hai tab rukna hai, otherwise start++ krte rho
        {
        start++ ;
        }
        while(arr[end] > pivot) // agr pivot se chotta element milta hai ab rukna hai, otherwise end-- krte rho
        {
        end-- ;
        }
        if(start<end)
        {
            swap(arr[start],arr[end]) ; // swap the arr[start] and arr[end] elements
        }
    }

    swap(arr[s],arr[end]) ; // jab start > end ho jaye
    return end ; // return the correct position of pivot element
}

void quickSort(int arr[], int s, int e)
{
    if(s<e)
    {
        int p = partition(arr,s,e) ; // p pivot element ki correct position btata hai
        quickSort(arr,s,p-1) ; // pivot element ke left array ko sort krlo using recursion
        quickSort(arr,p+1,e) ; // pivot element ke right array ko sort krlo using recursion
    }
}

void print(int arr[], int n)
{
    for(int i=0;i<n;i++)
    {
        cout<<arr[i]<<" ";
    }
    cout<<endl;
}

int main()
{
```

```
    int arr[] ={23,66,12,4,81,40,100} ;
    int n = sizeof(arr)/sizeof(arr[0]) ;

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

    return 0;
}
```