Assignment No: 2

Name: Kedar Kale

Batch: B12

Roll No: SCETTYB129

Problem Statement: Design and implement Quick Sort algorithm using Divide and Conquer method for a given input. Determine the time required to search an element.

Algorithm:

Step 1 − Choose the highest index value has pivot

Step 2 − Take two variables to point left and right of the list excluding pivot

Step 3 − left points to the low index

Step 4 − right points to the high

Step 5 − while value at left is less than pivot move right

Step 6 − while value at right is greater than pivot move left

Step 7 − if both step 5 and step 6 does not match swap left and right

Step 8 − if left ≥ right, the point where they met is new pivot

Quick Sort Pivot Pseudocode:

```
function partitionFunc(left, right, pivot)
   leftPointer = left
   rightPointer = right - 1

   while True do
      while A[++leftPointer] < pivot do
         //do-nothing
      end while

      while rightPointer > 0 && A[--rightPointer] > pivot do
         //do-nothing
      end while
      if leftPointer >= rightPointer
         break
      else
         swap leftPointer,rightPointer
      end if

   end while

   swap leftPointer,right
   return leftPointer

end function
```

Quick Sort Algorithm:

Step 1 − Make the right-most index value pivot

Step 2 − partition the array using pivot value

Step 3 − quicksort left partition recursively

Step 4 − quicksort right partition recursively

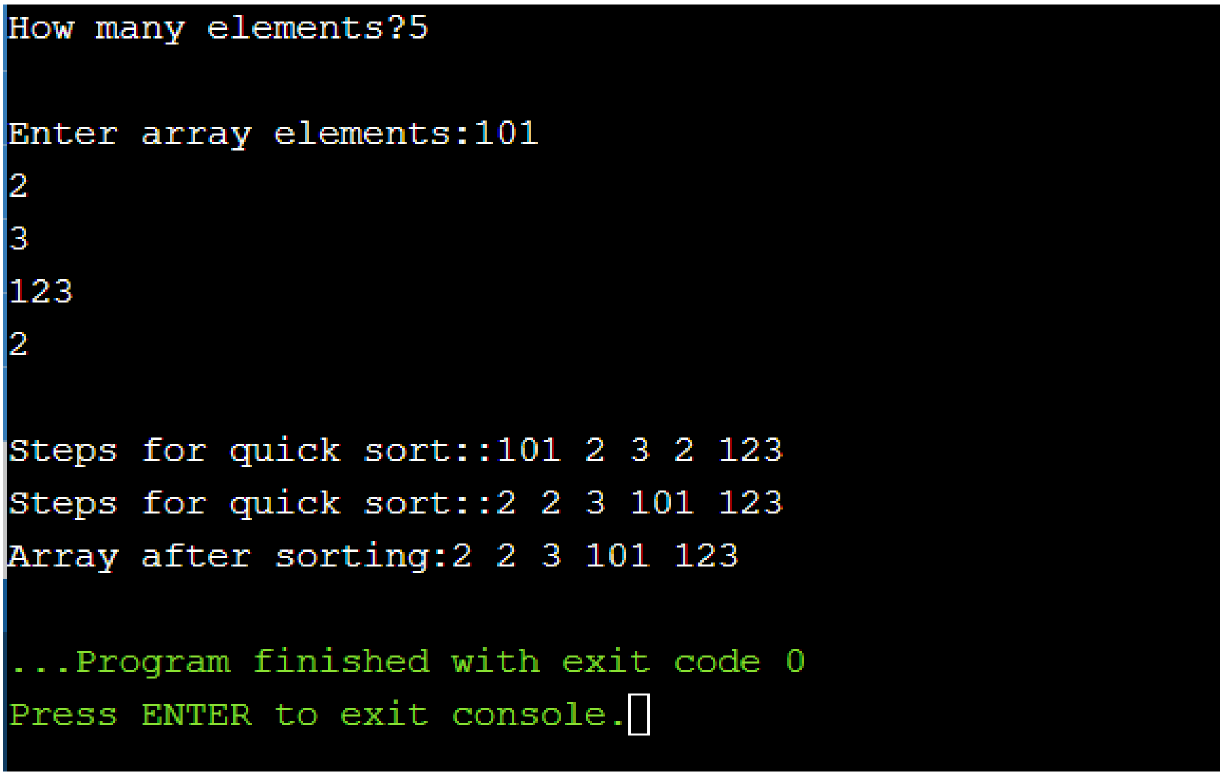
Quick Sort Pseudocode:

```
procedure quickSort(left, right)

   if right-left <= 0
      return
   else
      pivot = A[right]
      partition = partitionFunc(left, right, pivot)
      quickSort(left,partition-1)
      quickSort(partition+1,right)
   end if
end procedure
```

Program:

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




void quick_sort(int[],int,int);
int partition(int[],int,int);
int n;
int main()
{
    int a[50],i;
    cout<<"How many elements?";
    cin>>n;
    cout<<"\nEnter array elements:";

    for(i=0;i<n;i++)
        cin>>a[i];

    quick_sort(a,0,n-1);
    cout<<"\nArray after sorting:";

    for(i=0;i<n;i++)
        cout<<a[i]<<" ";

    return 0;
}

void quick_sort(int a[],int l,int u)
{
    int j;
    if(l<u)
    {
        j=partition(a,l,u);
        quick_sort(a,l,j-1);
        quick_sort(a,j+1,u);
    }
}

int partition(int a[],int l,int u)
{
    int v,i,j,temp;
    v=a[l];
    i=l+1;
    j=u;

    do
    {

        while(a[i]<v&&i<=u){
     i++;
}




        while(v<a[j]){
     j--;
}

        if(i<j)
        {
            temp=a[i];
            a[i]=a[j];
            a[j]=temp;
        }



    }

while(i<j);

cout<<"\nSteps for quick sort::";

 for(i=0;i<n;i++)
        cout<<a[i]<<" ";

    a[l]=a[j];
    a[j]=v;


    return(j);
}
```

Output:



```
How many elements?5

Enter array elements:101
2
3
123
2

Steps for quick sort::101 2 3 2 123
Steps for quick sort::2 2 3 101 123
Array after sorting:2 2 3 101 123

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