EXPERIMENT TITLE: QUICK SORT

Student Name: Parikshit sharma **UID:** 19BCS4520
Branch: CSE (IOT) **Section/Group:** 1/A
Semester: 3 **Date of Performance:**
Subject : DESIGN AND ANALYSIS OF ALGORITHMS LAB
Subject Code: CSP-240

1. **Aim/Overview of the practical:**
   write a program and analyse to sort an array of integers using quick sort.
2. **Task to be done:**
   Sort an array containing integers using quick sort.

3.Algorithm/Flowchart:

l: initial value
h: final value
q: position of the pivot value
arr: array

```
quicksort (arr,l,h)
If (l<h)
  then q =partition (arr,l,h)
  quicksort (arr, l, q-1)
  quicksort (arr, q+1, h)
```

where partition algorithm is:

```
Partition(arr,l,h)
1. x=arr[h]
2. i=l-1
3. For j=l to h-1
4. do if arr[j]<=x
5. then i=i+1
6. exchange arr[i]<-> arr[j]
7. exchange arr[i+1]<-> arr[r]
8. return i+1
```

4.Theme/Interests definition:

Quick sort is a divide-and-conquer algorithm. ... It works by selecting a 'pivot' element from the array and partitioning the other elements into two sub-arrays,

according to whether they are less than or greater than the pivot. The sub-arrays are then sorted recursively

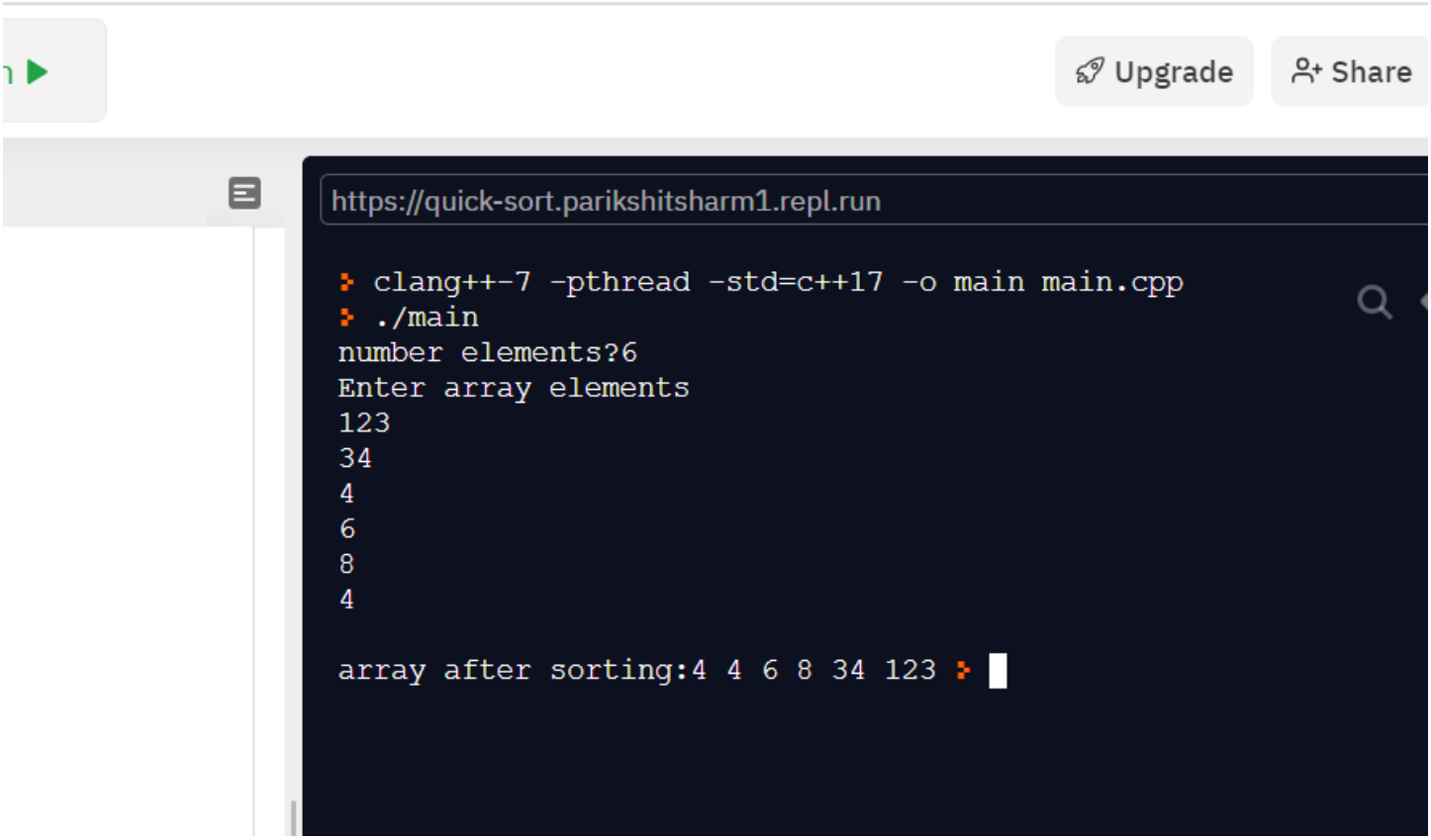
5. Steps for experiment/practical:

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

void quick_sort(int[],int,int);
int partition(int[],int,int);

int main()
{
    int a[50],n,i;
    cout<<"number elements?";
    cin>>n;
    cout<<"Enter array elements"<<endl;

    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;
      j=u+1;

        do
    {
        do
            i++;

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

           do
              j--;
          while(v<a[j]);

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

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

    return(j);
}
```

6. Percentage error (if any or applicable):NIL

7. Result/Output/Writing Summary:



8. Graphs (If Any): Image/Soft copy of graph paper to be attached here:NIL

Learning outcomes (What I have learnt):

1. To write an efficient algorithm.
2. To write an effect program of quick sort using divide and conquer in c++.
3. Worst case time complexity of quick sort is O(n^2).
4. Best and average time complexities of quick sort is O(n log n).

5. Worst case space complexity of quick sort is O(n).

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			