NAME: Rahul Singh

ROLL NO: S-56

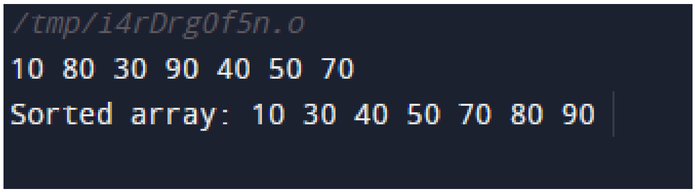
SUBJECT: AOA

EXPERIMENT NO: 3

To implement Quicksort and comparative analysis for large values of 'n' using DAC technique

```
#include <stdio.h>
void swap(int* a, int* b)
{
  int temp = *a;
  *a = *b;
  *b = temp;
}
int partition(int arr[] , int low, int high){
  int pivot = arr[low];
  int i = low;
  int j = high;

  while (i < j) {
        // condition 1: find the first element greater than
        // the pivot (from starting)
        while (arr[i] <= pivot && i <= high - 1) {
            i++;
        }

        // condition 2: find the first element smaller than
        // the pivot (from last)
        while (arr[j] > pivot && j >= low + 1) {
            j--;
        }
        if (i < j) {
            swap(&arr[i], &arr[j]);
        }
    }
    swap(&arr[low], &arr[j]);
    return j;
}
void Quicksort(int arr[],int low,int high){
    if (low <= high){
        int q = partition(arr,low,high);
        Quicksort(arr,low,q-1);
        Quicksort(arr,q+1,high);
    }
}



int main() {


   int data[] = {10,80,30,90,40,50,70};
   int n = sizeof(data)/sizeof(data[0]);
   for(int i = 0;i<n;i++){
       printf("%d ",data[i]);
   }
Quicksort(data,0,n-1);
    printf("\nSorted array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", data[i]);
    }

    return 0;
}
```

# OUTPUT:



```
/tmp/i4rDrg0f5n.o
10 80 30 90 40 50 70
Sorted array: 10 30 40 50 70 80 90
```