Experiment 10

Objective: Implementation of Heap sort and Quick sort

Code:

Heap sort:

```
#include<stdio.h>


void Adjust(int Heap[],int i)
{
   int j,temp,n;
   int r= 1;
   n=Heap[0];
   while(2*i<=n && r==1)
   {
      j=2*i;
      if(j+1<=n && Heap[j+1] > Heap[j])
      j=j+1;
      if( Heap[j] < Heap[i])
      r=0;
      else
         {
            temp=Heap[i];
            Heap[i]=Heap[j];
            Heap[j]=temp;
            i=j;
         }
   }
}
```

```
void Make_Heap(int heap[])
{
   int i,N;
   N=heap[0];
   for(i=N/2;i>=1;i--)
   Adjust(heap,i);
}
int main()
{
   int heap[30],N,L,temp,i;
   printf("Enter the number of elements to be sorted:");
   scanf("%d",&N);
   printf("Enter the elements:");
   for(i=1;i<=N;i++)
   scanf("%d",&heap[i]);
   heap[0]=N;
   Make_Heap(heap);
   while(heap[0] > 1)
   {
      L=heap[0];
      temp=heap[1];
      heap[1]=heap[L];
      heap[L]=temp;
      heap[0]--;
      Adjust(heap,1);
   }
   printf("Sorted Array:");
```

```
    for(i=1;i<=N;i++)
    printf("%d ",heap[i]);
    return 0;
}
```

Output:

Quick sort:

```
#include <stdio.h>

void quicksort (int [], int, int);

int main()
{
    int list[50];
    int size, i;

    printf("Enter the number of elements: ");
    scanf("%d", &size);
    printf("Enter the elements to be sorted:\n");
    for (i = 0; i < size; i++)
    {
        scanf("%d", &list[i]);
    }
```

```
    quicksort(list, 0, size - 1);
    printf("After applying quick sort\n");
    for (i = 0; i < size; i++)
    {
        printf("%d ", list[i]);
    }
    printf("\n");

    return 0;
}
void quicksort(int list[], int low, int high)
{
    int pivot, i, j, temp;
    if (low < high)
    {
        pivot = low;
        i = low;
        j = high;
        while (i < j)
        {
            while (list[i] <= list[pivot] && i <= high)
            {
                i++;
            }
            while (list[j] > list[pivot] && j >= low)
            {
                j--;
```

```
            }
            if (i < j)
            {
                temp = list[i];
                list[i] = list[j];
                list[j] = temp;
            }
        }
        temp = list[j];
        list[j] = list[pivot];
        list[pivot] = temp;
        quicksort(list, low, j - 1);
        quicksort(list, j + 1, high);
    }
}
```

Output:

Submitted by: Gelle Hruthesh Reddy(20BCB7031)