LAB PROGRAMS

D.BHARGAV
AP19110010437

1. Write a program for the Insertion sort algorithm.

```
#include<stdio.h>
int main()
{
   int i, j, size, temp, array[25];

   printf("enter size of elements: ");
   scanf("%d",&size);

   printf("Enter %d elements: ", size);
   for(i=0;i<size;i++)
      scanf("%d",&array[i]);
   for(i=1;i<size;i++){
      temp=array[i];
      j=i-1;
      while((temp<array[j])&&(j>=0))
      {
         array[j+1]=array[j];
         j=j-1;
      }
      array[j+1]=temp;
   }

   printf("Sorted elements: ");
   for(i=0;i<size;i++)
      printf(" %d",array[i]);

   return 0;
}
```

2. Write a program for the Selection sort algorithm.

```
#include <stdio.h>
int main()
{
  int array[100], size, i, j, position, t;

  printf("enter size of elements\n");
  scanf("%d", &size);

  printf("Enter %d integers\n", size);
```

```
  for (i=0; i<size;i++)
    scanf("%d", &array[i]);

  for (i=0;i<(size-1);i++)
  {
    position = i;

    for (j=i+1;j<size;j++)
    {
      if (array[position]>array[j])
        position=j;
    }
    if (position != i)
    {
      t = array[i];
      array[i] = array[position];
      array[position] = t;
    }
  }

  printf("Sorted list in ascending order:\n");

  for (i=0;i<size;i++)
    printf("%d\n", array[i]);

  return 0;
}
```

3. Write a program for the Bubble sort algorithm.

```
#include<stdio.h>
int main()
{
    int array[10],p,q,temp,size;
    printf("\n enter size of elements: \n");
    scanf("%d",&size);
    printf("\n enter the elements : \n");
    for(p=0; p<size; p++)
    {
        scanf("%d",&array[p]);
    }
    for(p=0; p<size; p++)
        for(q=p+1; q<size; q++)
        {
            if(array[p]>array[q])
            {
```

```
            temp=array[p];
            array[p]=array[q];
            array[q]=temp;
         }
      }
   for(p=0; p<size; ++p)
   {
      printf("%d",array[p]);
   }
   return 0;
}
```

4. Write a program for the Merge sort algorithm.

```
#include <stdio.h>
void merge(int arr[], int p, int q, int r)
{
  int n1 = q - p + 1;
  int n2 = r - q;

  int L[n1], M[n2];

  for (int i = 0; i < n1; i++)
    L[i] = arr[p + i];
  for (int j = 0; j < n2; j++)
    M[j] = arr[q + 1 + j];
  int i, j, k;
  i = 0;
  j = 0;
  k = p;

  while (i < n1 && j < n2)
  {
    if (L[i] <= M[j])
    {
      arr[k] = L[i];
      i++;
    }
    else
    {
      arr[k] = M[j];
      j++;
    }
    k++;
  }
  while (i < n1)
  {
```

```
    arr[k] = L[i];
    i++;
    k++;
  }

  while (j < n2)
  {
    arr[k] = M[j];
    j++;
    k++;
  }
}
void mergeSort(int arr[], int l, int r)
{
  if (l < r)
  {
    int m = l + (r - l) / 2;

    mergeSort(arr, l, m);
    mergeSort(arr, m + 1, r);
    merge(arr, l, m, r);
  }
}
void printArray(int arr[], int size)
{
  for (int i = 0; i < size; i++)
    printf("%d ", arr[i]);
  printf("\n");
}
int main() {
  int arr[] = {5, 7, 22, 11, 9, 4};
  int size = sizeof(arr) / sizeof(arr[0]);

  mergeSort(arr, 0, size - 1);

  printf("Sorted array: \n");
  printArray(arr, size);
}
```

5. Write a program for the Heap sort algorithm.

```
#include<stdio.h>
void heapsort(int[],int);
void heapify(int[],int);
void adjust(int[],int);
int main()
{
```

```
        int p,z,a[50];
        printf("\nEnter the limit:");
        scanf("%d",&p);
        printf("\nEnter the elements:");
        for (z=0;z<p;z++)
          scanf("%d",&a[z]);
        heapsort(a,p);
        printf("\nThe Sorted Elements Are:\n");
        for (z=0;z<p;z++)
          printf("\t%d",a[z]);
        printf("\n");
}
void heapsort(int a[],int p)
{
        int z,t;
        heapify(a,p);
        for (z=p-1;z>0;z--) {
                t = a[0];
                a[0] = a[p];
                a[p] = t;
                adjust(a,p);
        }
}
void heapify(int a[],int n) {
        int k,p,q,item;
        for (k=1;k<n;k++) {
                item = a[k];
                p = k;
                q = (p-1)/2;
                while((p>0)&&(item>a[q])) {
                        a[p] = a[q];
                        p = q;
                        q = (p-1)/2;
                }
                a[p] = item;
        }
}
void adjust(int a[],int n) {
        int p,q,item;
        q = 0;
        item = a[q];
        p = 2*q+1;
        while(p<=n-1)
        {
                if(p+1 <= n-1)
                  if(a[p] <a[p+1])
```

```
                p++;
            if(item<a[p])
            {
                    a[q] = a[p];
                    q = p;
                    p = 2*q+1;
            } else
              break;
        }
        a[q] = item;
}
```