J.Naveen
AP19110010378
CSE-H

DSA Lab programs

1. Write a program for the Insertion sort algorithm.

Program:

```
#include <stdio.h>

int main()
{
  int n, array[1000], a, b, c, flag = 0;

  printf("Enter number of elements\n");
  scanf("%d", &n);

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

  for (a = 0; a < n; a++)
    scanf("%d", &array[a]);

  for (a = 1 ; a <= n - 1; a++) {
    c = array[a];

    for (b = a - 1 ; b >= 0; b--) {
      if (array[b] > c) {
        array[b+1] = array[b];
        flag = 1;
      }
      else
        break;
    }
    if (flag)
      array[b+1] = c;
  }
```

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

  for (a = 0; a <= n - 1; a++) {
    printf("%d\n", array[a]);
  }

  return 0;
}
```

Output:

```
Enter number of elements
5
Enter 5 integers
7
8
4
6
18
Sorted list in ascending order:
4
6
7
8
18
```

2. Write a program for the Selection sort algorithm.

Program:

```
#include<stdio.h>
int main(){

  int i, j, count, temp, a[25];

  printf("enter the number of elements: ");
  scanf("%d",&count);
```

```
  printf("Enter %d elements: ", count);

  for(i=0;i<count;i++)
    scanf("%d",&a[i]);


  for(i=0;i<count;i++){
    for(j=i+1;j<count;j++){
      if(a[i]>a[j]){
        temp=a[i];
        a[i]=a[j];
        a[j]=temp;
      }
    }
  }

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

  return 0;
}
```

Output:

```
enter the number of elements: 6
Enter 6 elements: 4
54
68
12
3
9
Sorted elements:  3 4 9 12 54 68
```

3. Write a program for the Bubble sort algorithm.

program:

```
#include<stdio.h>

int main(){

   int count, temp, i, j, a[30];

   printf("How many numbers are u going to enter?: ");
   scanf("%d",&count);

   printf("Enter %d numbers: ",count);

   for(i=0;i<count;i++)
   scanf("%d",&a[i]);

   for(i=count-2;i>=0;i--){
      for(j=0;j<=i;j++){
        if(a[j]>a[j+1]){
           temp=a[j];
           a[j]=a[j+1];
           a[j+1]=temp;
        }
      }
   }

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

   return 0;
}
```

Output:

```
How many numbers are u going to enter?: 6
Enter 6 numbers: 5
2
1
9
8
4
Sorted elements:  1 2 4 5 8 9
```

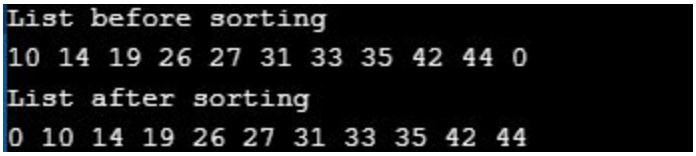
4. Write a program for the Merge sort algorithm

Program:

```
 #include <stdio.h>

#define max 10

int a[11] = { 10, 14, 19, 26, 27, 31, 33, 35, 42, 44, 0 };
int b[10];

void merging(int low, int mid, int high) {
  int l1, l2, i;

  for(l1 = low, l2 = mid + 1, i = low; l1 <= mid && l2 <= high; i++) {
    if(a[l1] <= a[l2])
      b[i] = a[l1++];
    else
      b[i] = a[l2++];
  }

  while(l1 <= mid)
    b[i++] = a[l1++];

  while(l2 <= high)
    b[i++] = a[l2++];
```

```
   for(i = low; i <= high; i++)
      a[i] = b[i];
}

void sort(int low, int high) {
   int mid;

   if(low < high) {
      mid = (low + high) / 2;
      sort(low, mid);
      sort(mid+1, high);
      merging(low, mid, high);
   } else {
      return;
   }
}

int main() {
   int i;

   printf("List before sorting\n");

   for(i = 0; i <= max; i++)
      printf("%d ", a[i]);

   sort(0, max);

   printf("\nList after sorting\n");

   for(i = 0; i <= max; i++)
      printf("%d ", a[i]);
}
```

Output:



5) Write a program for the Heapsort algorithm.

Program:

```
#include <stdio.h>

void swap(int *a, int *b) {
  int temp = *a;
  *a = *b;
  *b = temp;
}

void heapify(int arr[], int n, int i) {
  int largest = i;
  int left = 2 * i + 1;
  int right = 2 * i + 2;

  if (left < n && arr[left] > arr[largest])
    largest = left;

  if (right < n && arr[right] > arr[largest])
    largest = right;

  if (largest != i) {
    swap(&arr[i], &arr[largest]);
    heapify(arr, n, largest);
  }
}

void heapSort(int arr[], int n) {
  for (int i = n / 2 - 1; i >= 0; i--)
    heapify(arr, n, i);

  for (int i = n - 1; i >= 0; i--) {
    swap(&arr[0], &arr[i]);

    heapify(arr, i, 0);
  }
}
```

```
void printArray(int arr[], int n) {
  for (int i = 0; i < n; ++i)
    printf("%d ", arr[i]);
  printf("\n");
}

 int main() {
  int arr[] = {1, 12, 9, 5, 6, 10};
  int n = sizeof(arr) / sizeof(arr[0]);

  heapSort(arr, n);

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

Output: