LAB PROGRAMS

J.PRAHARSHINI
AP19110010461
CSE: F

Write a program for the Insertion sort algorithm.

```
#include <stdio.h>

int main()
{
  int n, array[1000], i, j, t, flag = 0;

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

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

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

  for (i = 1 ; i <= n - 1; i++) {
    t = array[i];

    for (j = i - 1 ; j >= 0; j--) {
      if (array[j] > t) {
        array[j+1] = array[j];
        flag = 1;
      }
      else
        break;
    }
    if (flag)
      array[j+1] = t;
  }

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

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

  return 0;
}
```

Output:
Enter the number of elements:
2
Enter two integers:
1
3
Sorted list in ascending order:
1
3

Write a program for the Selection sort algorithm.

```
#include <stdio.h>
int main()
{
  int array[100], n, c, d, position, t;

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

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

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

  for (c = 0; c < (n - 1); c++)   {
    position = c;

    for (d = c + 1; d < n; d++)
    {
      if (array[position] > array[d])
        position = d;
    }
    if (position != c)
    {
      t = array[c];
      array[c] = array[position];
      array[position] = t;
    }
  }

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

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

  return 0;
```

```
}
```

Output:
Enter the number of elements:
3
Enter the integers:
2
3
4
Sorted list in ascending order:
2
3
4

Write a program for the Merge sort algorithm.

```
#include<stdio.h>
#include<conio.h>
#define MAX_SIZE 5
void merge_sort(int, int);
void merge_array(int, int, int, int);
int arr_sort[MAX_SIZE];
int main() {
  int i;
  printf("Simple Merge Sort Example - Functions and Array\n");
  printf("\nEnter %d Elements for Sorting\n", MAX_SIZE);
  for (i = 0; i < MAX_SIZE; i++)
    scanf("%d", &arr_sort[i]);
  printf("\nYour Data   :");
  for (i = 0; i < MAX_SIZE; i++) {
    printf("\t%d", arr_sort[i]);
  }
  merge_sort(0, MAX_SIZE - 1);
  printf("\n\nSorted Data :");
  for (i = 0; i < MAX_SIZE; i++) {
    printf("\t%d", arr_sort[i]);
  }
  getch();
}
void merge_sort(int i, int j) {
  int m;
  if (i < j) {
    m = (i + j) / 2;
    merge_sort(i, m);
    merge_sort(m + 1, j);
    merge_array(i, m, m + 1, j);
  }
```

```
}
void merge_array(int a, int b, int c, int d) {
  int t[50];
  int i = a, j = c, k = 0;

  while (i <= b && j <= d) {
    if (arr_sort[i] < arr_sort[j])
      t[k++] = arr_sort[i++];
    else
      t[k++] = arr_sort[j++];
  }
  while (i <= b)
    t[k++] = arr_sort[i++];
  while (j <= d)
    t[k++] = arr_sort[j++];

  for (i = a, j = 0; i <= d; i++, j++)
    arr_sort[i] = t[j];
}
```

Output:

```
Simple merge sort Example - Functions and array
Enter 5 elements for sorting
2
3
6
7
1
Your data: 2 3 6 7 1
Sorted data:1 2 3 6 7
```

Write a program for Bubble sort algorithm.

```
#include<stdio.h>

int main(){

   int count, temp, i, j, number[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",&number[i]);
```

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

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

   return 0;
}
```

Output:

```
How many numbers are u going to enter:3
Enter 3 numbers:1
4
6
Sorted elements: 1 4 6
```