AP19110010380
CSE-H

1. Write a program for Insertion sort algorithm?

```
// C program for Insertion sort
#include<math.h>
#include<stdio.h>
/* Function to sort using Insertion sort */
void insertionsort (int arr[ ], int n)
{
  int i, key, j;
  for(i=1; i<n; i++)
  {
     key=arr[i];
      j=i-1;
    while (j>=0 && arr[j] >key)
    {
       arr[j+1]=arr[j];
        j=j-1;
    }
     arr[j+1]=key;
  }
}
```

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

/* Driver program to test insertion sort */
int main()
{
   int arr[] = {12, 11, 13, 5, 6};
   int n = sizeof(arr)/sizeof(arr[0]);

   insertionsort(arr, n);
   printArray(arr, n);
   return 0;
}
```

Output:-

```
5 6 11 12 13
```

2. Write a program for the selection sort algorithm.

```
/* C program for selection sort */
# include<stdio.h>
void swap(int *xp , int *yp)
{
 int temp = *xp;
     *xp = *yp;
     *yp = temp;

}
 void selectionsort(int arr[ ], int n)
 {
  int i, j, min_idx;
  for(i=0; i<n-1; i++)
  {
    min_idx = i;
    for(j= i+1; j<n; j++)
      if (arr[j]< arr[min_idx])
        min_idx = j;
    swap (&arr[min_idx], &arr[i]);
 }
}
```

```
/* Function to print an array */
void printArray (int arr[ ], int size)
{
  int i;
  for (i=0; i<size ; i++)
      printf ("%d", arr[i]);
  printf ("\n");
}
// program to test above function //
int main ()
{
  int arr[ ] = { 2, 7, 8, 11, 1};
  int n = sizeof (arr) / size of (arr[0]);
  selectionsort (arr, n);
  printf ("Sorted array: "\n");
  printArray (arr, n);
  return 0;
}
```

Output :

```
Sorted array :
1  2   7  8  11.
```

3. Write a program for Bubble sort algorithm.

```
/* C program for Bubble sort algorithm */
#include<stdio.h>
int main( )
{
 int count, temp, i, j, number[80];

 printf("How many numbers are you 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;
     }
   }
```

Scanned with CamScanner

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

Output :-

```
How many numbers are you going to enter 6
Enter 6 numbers: 66  0  12  89  65  99
Sorted elements:  0  12  65  66  89  99.
```

4. Write a program for the merge sort algorithm.

```
/* C program for merge sort */
#include <stdlib.h>
#include <stdio.h>
// Merges two subarrays of arr[].
void merge(int arr[ ], int l, int m, int r).
{
  int i, j, k;
  int n1 = m-l+1;
  int n2 = r-m;
  int L[n1], R[n2];
 /* copy data to temp arrays L[] and R[]*/
 for (i=0; i<n1; i++)
      L[i]=arr[l+i];
 for(j=0; j<n2; j++)
      R[j] = arr[m+1+j];
 /* Merge the temp arrays back into arr[l..r]*/
  i=0;
  j=0;
  k=l;
```

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

```
}
void mergesort (int arr[], int l, int r)
{
  if(l<r)
  {
    int m= l+(r-1)/2
    mergesort(arr, l, m);
    mergesort (arr, m+1, r);
    mergesort (arr, l, m, r);
  }
}
/* Function to print Array */
void printArray (int A[], int size)
{
  int i;
  for(i=0; i<size; i++)
      printf ("%d", A[i]);
  printf("\n");
}
/* Driver program to test above function*/
```

```
int main( )
{
  int arr[ ] = {2, 8, 14, 7, 16, 1};
  int arr_size = size of(arr)/sizeof (arr[0]);

  printf("Given array is \n");
   printArray (arr, arr_size);
   mergesort(arr, 0, arr_size -1);
   printf("\n Sorted array is \n");
    printArray(arr, arr_size);
    return 0;
 }
```

<u>Output</u> :~

```
Given array is
2  8  14  7  16  1

Sorted array is
1  2  7  8  14  16.
```

5. Write a program for the heap sort algorithm.

```
#include<stdio.h>
void create (int[ ]);
void down_adjust(int[ ], int);
void main( )
{
   int heap[30], n, i, last, temp;
    printf("Enter number of elements:");
    scanf("%d", &n);
    printf("\n Enter elements: ");
    for(i=1; i<=n; i++)
        scanf("%d", &heap[i]);
     // create a heap
     heap[0]=n;
     create (heap);

     // sorting
     while(heap[0]>1)
     {
        last = heap[0];
        heap[0]= heap[1];
        heap[1]= heap [last];
   heap[last]=temp;
   heap[0]--;
   down_adjust(heap,1);
 }
 printf("\n Array after sorting:\n");
 for(i=1;i<=n;i++)
     printf("%d",heap[i]);
}
void create(int heap[])
{
  int i,n;
  n=heap[0];
  for(i=n/2;i>=1;i--).
        down_adjust(heap,i);
 }
 void down_adjust(int heap[],int i)
 {
   int j,temp,n,flag=1
   n=heap[0]
```

```
while (2*i<=n && flag==1)
{
   j =2*i;
   if(j+1<=n && heap[j+1] >heap[j])
        j=j+1;
   if(heap[i] > heap[j])
       flag=0;
  else
   {
     temp=heap[i];
     heap[i]=heap[j];
     heap[j]=temp;
     i=j;
   }
 }
}
```

Output:-

```
Enter number of elements: 5
Enter elements : 12  8  46 23 7
Array after sorting:
7  8  12  23  46
```