DSA LAB ASSIGNMENT-3

1) Insertion sort algorithm in C.

```
#include <stdio.h>
int main()
{
int n, arr[100], c,b,a ,flag=0;
printf("Enter no.of elements");
scanf("%d", &n);
for(c=0; c<n;c++)
   scanf("%d", &array[c]);
for(c=1; c<=n-1; c++){
  t = array[c];
for(b=c-1; b>=0; b--){
 if(array[d] >t){
 array[d+1] = array[d]
  flag=1;
}
else
break;
}
if(flag)
  array[b+1]=a;
}
Printf("sorted list");
for(c=0; c<=n-1;c++)
 Printf("%d", &array[c]);
}
return(0);
```

2) selection sort algorithm in c

```
#include <stdio.h>
int main()
{
int array[100], n, c, b, position, a;
Printf ("Give no. of elements");
Scanf ("%d", &n);
Printf ("Enter
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)
{
a = array[c]
array[c] =array[position];
array[position] =a;
}
}

for (c=0; c<n; c++);
Printf ("%d", array[c]);
return 0;
}
```

3) C program for Bubble sort algorithm.

```
#include <stdio.h>
int main()
{
int array[100], n, c, b, swap;
Printf ("Give no. of elements");
scanf ("%d", &n);
for (c=0; c<n; c++)
 scanf ("%d", &array [c]);
for (c=0; c<n-1; c++)
{
for (b=0; b<n-1-c; b++)
 {
 if (array[b] > array[b+1])
 {
 swap    =array[d];
array[d] = array[d+1];
array[d+1]=swap;
 }
 }

for(c=0; c<n; c++)
 Printf ("%d", array [c]);
return 0;
}
```

1) MERGE SORT Algorithm in C.

```
#include <stdio.h>
#include <conio.h>
#define max-size 5
Void merge-sort (int, int);
Void merge-sort (int, int, int, int);
int arr-sort [max-size];
int main()
{
int i;
Printf ("Enter elements for sorting %d", max-size);
for (i=0; i<max-size; i++)
scanf ("%d", &arr-sort[i]);
for (i=0; i<max-size; i++)
{
Printf ("%d", arr-sort[i]);
}
merge-sort (0, max-size-1);
for (i=0; i<max-size; i++);
Printf ("%d", arr-sort[i]);
}
getch();
Void merge-sort (int i, int j)
{
int n
if (i<j)
n = (i+j)/2
merge-sort (i, n);
merge-sort (n+1, j);
merge-array (i, n, n+1, j);
}
}
```

```
void merge_sort (int x, y, z, p) {
 int t [50];
 int i = x, j = z, k = 0;
 while (i <= y && j <= p) {
  if (arr_sort[i] < arr_sort[j])
    t[k++] = arr_sort[i++];
  else
  t[k++] = arr_sort[j++];
  }
 while (i <= y)
 t[k++] = arr_sort[i++];
 while (j <= p)
  t[k++] = arr_sort[j++];
 for (i = x, j = 0; i <= p; i++, j++)
   arr_sort[i] = t[j];
  }
```

---

5) C program for heap sort algorithm.

```
#include <stdio.h>
void main()
{
int heap[10], n, i, j, c, root, temp;
printf ("Give no. of elements :");
scanf ("%d", &n);
for (i=0; i < n; i++)
    scanf ("%d", &heap[i]);
for (i=1; i<n; i++)
{
 c = i;
 do
```

```
do
{
root = (c-1)/2;
if (heap[root] < heap[c])
 {
 temp = heap[root];
 heap[root] = heap[c];
 heap[c] = temp;
 }
 c = root;
}while(c!=0);
}
for(i=0;i<n;i++)
  printf("%d", heap[i]);
for(j=n-1;j>=0;j--)
 {
 temp = heap[0]
 heap[0] = heap[j]
 heap[j] = temp;
 root = 0;
  do
  {
  c = 2*root+1
 if((heap[c] < heap[c+1]) && (c<j-1))
 c++;
 if(heap[root] < heap[c]) && (c<j))
  {
  temp = heap[root];
  heap[root] = heap[c];
  heap[c] = temp;
 root = c
  }while(c<j);
 for(i=0; i<n; i++)
  printf("%d", heap[i])
  }
 return(0);
 }
```

THANK YOU