FIRST SEMESTER MCA (2020 SCHEME)

PRACTICAL EXAMINATION JUNE-JULY 2021

20MCA135 DATA STRUCTURES LAB.

HASEENA. P. A.
ICE20MCA-2024
Date: 30 June 2021
Time: 9:30-12:30
Regno:

1) Write a program to sort an integer array?

Program:

```
#include <stdio.h>
# include <conio.h>
 Void main()
 {
    int i, j, a, n, number[30];
    clrscr();
    Printf("Enter the value of N \n");
    Scanf("%d", &n);
    Printf ("Enter the numbers \n");
    for (i=0; i<n; ++i)
    Scanf("%d", & number[i]);
    for(i=0; i<n; ++i)
      {
          for (j= i+1; j<n; ++j)
          {
              if (number[i] >number[j])
               {
                   a= number[i];
```

```
        number[i] = number[j];
         number[j] = a;
      }
    }
  }
printf ("The numbers arranged in ascending order
            are given below \n");
  for (i=0; i<n; ++i)
   printf ("%d\n", number[i]);
    getch();
  }
```

Expected output

```
Enter the value of N
5
Enter the numbers
 2
 5
 8
 3
 1
The numbers arranged in ascending order are
 given below
 1
 2
 3
 5
 8
```

2) Write a program to implement disjoint set operations?

Program :

```
# include <stdio.h>

# include <conio.h>

Struct Disjset
{
  int parent [10];
  int rank [10];
  int n;
} dis;

Void makeSet ()
 {
   int i;
   for (i=0 ; i< dis.n ; i++)
    {
      dis.parent[i]= i;
      dis. rank [i]=0;
    }
 }

 Void displayset()
  {
     int i;
     PrintF ("\n Parent Array\n");
     for (i= 0; i< dis.n; i++)
```

```
{
printF (" %d ", dis. parent [i]);
}
printF ("\n Rank Array \n");
for(i=0; i<dis.n; i++)
 {
  printF("%d", dis.rank[i]);
 }
 printF ("\n");
  int  Find (int x)
   {
   iF (dis.parent(x]! = x)
    {
    dis. parent(x] = find (dis. parent[x]);
    }
    return  dis. parent(x];
   }
  Void  Union (int x, int y)
    {
      int xSet = find (x);
      int yset = find(y);
      iF (xset = = yset)
      return;
      iF (dis. rank[xset]<dis. rank[yset])
       {
        dis. parent [xSet] = yset;
        dis. rank [xset] = -1;
       }
```

```
else if( dis.rank [xset) > dis.rank [yset])
 {
  dis.parent [yset] = xset;
   dis.rank [yset] = -1;

 }
else
 {
  dis.parent [yset] = xset;

   dis. rank [xset) = dis. rank [xset] + 1 ;
   dis. rank [yset) = -1 ;
 }
 int main()
          {
             int x, y, n, ch, wish;

              clrscr();
            printf(" How  many elements?");
            scanf(" %d", & dis.n);
             makeSet();
             do
              {
             print f("\n.... menu.... \n");
             printf(" 1. Union \n 2. Find \n 3. Display \n");
             Print f(" enter choice \n");
             scanf(" %d ", & ch);
            switch (ch)
             {
```

```
case 1:
printF(" Enter elements to perform union:");
Scanf(" %d %d", &x,&y);
union(x,y);
break;
case 2:
printF(" Enter elements to check if connected components:");
Scanf(" %d %d",&x,&y);
IF (Find(x) == Find(y))
printF(" connected components");
break;
case 3:
displaySet();
break;
}
PrintF("\n Do you wish to continue? (1/0)\n");
Scanf("%d", &wish);
}
while (wish==1);
return 0;
}.
```

Expected Output

```
Howmany elements? 4

   --- Menu ---

  1. Union

  2. Find

  3. Display

 enter choice

 1

 Enter elements to perform union : 2  3

 Do you wish to continue? (1/0)

 1.


 --- Menu ---

1 union

2. Find

3. Display.

enter choice

 2

 Enter elements to check if connected components: 1 4

 Not connected components.

 Do you wish to continue? (1/0)
```

```
--- Menu---
1. union
2. Find.
3. Display.
enter choice
2
Enter elements to check if connected components: 1 4
Not connected components.
Do you wish to continue? (1/0)
1.
-- Menu--
1. Union
2. Find
3. Display.
enter choice
3
Parent Array.
0 1 2 2
Rank Array.
0 0 1 -1
Do you wish to continue? (1/0).
```