First Semester M.C.A (2020 Scheme)

Practical Examination June - July 2021

20 MCA 135 DATA STRUCTURE LAB

Date: 30-06-2021 Batch - I

Kavya Kanakan

ICE 20MCA - 2026

Part I.

Sorting of an Integer array.

Algorithm.

Step 1 : Start

Step 2 : Initialize array as number [30]

Step 3 : print "Enter the value of N"

Step 4 : Set i=0. Repeat Step 5 and Step 6 until i < n

Step 5 : Print arr [i]

Step 6 : i = i + 1

Step 7 : Set j = i + 1. Repeat Step 8 until j < n.

Step 8 : if (arr [i] > arr[j]) then

temp = arr [i]

arr[i] = arr[j]

arr [j] = temp.

Step 9 : j = j+1

i = i+1

Step 10 : print "The numbers arranged in ascending order are given below"

Step 11 : Set i=0. Repeat step 12 and step 13 until i < n.

Step 12 : print a[i]

Step 13 : Return 0.

Step 14 : End.

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++)
{
   for (j=0; 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~~

Output :

```
Enter the value of N:
5
Enter the numbers:
6
```

2
9
12
3

The numbers arranged in ascending order are given below

2
3
6
9
12

Part II

Disjoint Set Operations.

```
#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("Parent Array");
for(i=0; i<dis.n; i++)
{
printf("%d ", dis.parent[i]);
}
Printf("Rank Array");
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
{
printf ("\n Menu \n");
printf (" 1. Union \n 2. Find \n 3. Display \n");
Printf ("Enter choice ");
Scanf ("%d", &ch);
[illegible]
[illegible]
[illegible]
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 [illegible] check if Connected
        components : ");
Scanf ("%d %d", &x, &y);
if (find(x) == find (y))
printf ("Connected components ");
else
printf (" Not connected components ");
  break;
```

```
case 3:
display set ();
break;
}
printf ("Do you wish to continue ? (1/0) ");
scanf ("%d", &wish );
}
while (wish == 1);
return 0;
}
```

Output.

```
how many elements? 3

 Menu
1. Union
2. Find
3. Display

Enter a choice : 1
Enter the elements to union : 3  5
do you wish to continue : (1/0) 1

Enter a choice : 2
enter elements to check connected components :
 4  5
 connected.
```

do you wish to continue : (1/0) 1

Enter a choice : 3

Parent array :

0 1 2

rank array :

0 0 0

do you wish to continue: (1/0) 0