BATCH-I

First Semester MCA (2020 SCHEME)

PRATICAL EXAMINATION JUNE-JULY 2021

20MCA135 DATA STRUCTURE LAB

EBIN SABU
ICE20MCA-2022
Date: 30 June 2021
Time: 1.00 - 4.00

1. Sorting of integer array

Program

```
#include<stdio.h>
#include<conio.h>
void main()
 {
clrscr();
int i,j,a,n,number[30];
Printf("enter the size of the array:");
Scanf("%d",&n);
Printf("enter the elements of the array:");
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 sorted array is \n");
for(i=0;i<n;++i)
printf("%d", number[i]);
getch();
}
```

<u>output</u>

```
enter the size of the array : 6
enter the elements of the array : 3
5
1
4
8
7
the sorted array is
134578
```

2. 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("\nRank 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;
}
}
getch();
int main()
{
int x,y,n,ch,wish;
printf("How many elements ?");
scanf("%d",&dis.n);
makeset();
do
{
print("\n Menu \n");
```

```
printf("1. Union\n 2. find\n3. Display\n");
printf("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\n");
 else
 printf("Not connected components \n");
 break;
 case 3: displayset();
  break;
  }
 printf("\nDo you wish to continue ?(1/0)\n");
  scanf("%d", &wish);
  }
  while (wish==1);
  return 0;
  }
```

Output

```
How many 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 connetted components
Do you wish to continue ? (1/0)  1
   Menu
1. union
2. Find
3. Display
 enter choke
   3
```

```
Parent Array
0122
Rank Array
001-1
Do you wish to continue? (1/0)
 0
```