

Batch 1 First Semester MCA (2020 Scheme)  
PRACTICAL EXAMINATION JUNE/JULY  
2021  
20MCA135 DATA STRUCTURE LAB

Jerin Paul  
ICE 20MCA-2025  
Date: 30 Jun 2021  
Time: 1:00 to 4:30

1) Write a program to sort an integer array

```
#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 number arranged in ascending order are  
given below");
```

```
for(i=0; i<n; i++)
```

```
printf("%d", number[i]);
```

```
getchar();
```

```
}
```

Output

Enter the value of N  
5

Enter the numbers

5 7 1 2 9

The number arranged in ascending order are given

1

2

5

7

9

2) Write a program to implement disjoint set operation

```
#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("In Parent array : \n");
```

```
    for(i=0; i<dis.n; i++)
```

```
{  
    printf("o/d", dis.parent[i]);
```

```
}  
printf("In Rank array: \n");
```

```
for(i=0; i<dis.n; i++)
```

```
{  
    printf("o/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[yset] = dis.rank[xset] + 1;
```

```
    dis.rank[xset] = 1;
```

```
}
```

```
}
```

```
int main()
```

```
{
```

```
    int n, x, y;
```

```
    int ch, wish;
```

```
    clrscr();
```

```
    printf("how many elements?");
```

```
    scanf("%d", &n);
```

```
    makeSet();
```

```
printf("MENU");  
printf("1. union, 2. find, 3. display\n");
```

```
do
```

```
{
```

```
printf("Enter a choice");
```

```
scanf("%d", &ch);
```

```
switch (ch)
```

```
{
```

```
case 1: printf("Enter the elements to union: ");
```

```
scanf("%d%d", &x, &y);
```

```
Union(x, y);
```

```
break;
```

```
case 2: printf("Enter the elements to check connected  
comprer: ");
```

```
scanf("%d%d", &x, &y);
```

```
if (find(x) == find(y))
```

```
{
```

```
printf("connected");
```

```
}
```

```
else
```

```
printf("not connected");
```

```
break;
```

```
case 3: displaySet();
```

```
break;
```

```
}
```

```
printf("Do you wish to continue (1/0)");
```

```
scanf("%d", &cont);
```

```
}
```

```
while (cont == 1);
```

```
return 0;
```

```
}
```

Output

How many elements: 2 4

MENU

1. Union
2. Find
3. Display

enter choice

1

Enter the elements to perform union 2 3

Do you wish to continue 2 (1/0)

1

MENU

1. Union
2. Find
3. Display

Enter choice:

2

Enter the element to check if connected Component 1 4

Not Connected Components

Do you wish to continue? (Y/N)

1

MENU

1. Union

2. Find

3. Display

Enter choice  
3

Parent array

0122

Roots Array

001-1

Do you wish to continue? (Y/N)