FIRST SEMESTER MCA (2020 SCHEME)

PRACTICAL EXAMINATION JUNE-JULY 2021.

Nivya Manikuttan
ICE20MCA - 2031
Date: 30 June 2021
Time: 9:00 - 1:00 - 4:00

20MCA135 DATA STRUCTURES LAB.

batch-2

1. Write a program to merge two sorted arrays.

Program:

```
#include<stdio.h>
#include<conio.h>
  void main()
  {
int array1[50], array2[50], array3[100], m, n, i, j, k=0;
clrscr();
Printf("\n Enter the size of the array Array1:");
Scanf("%d", &m);
Printf("\n Enter sorted elements of array1:\n");
For(i=0; i<m; i++)
{
Scanf("%d", &array1[i]);
}
Printf("\n Enter the size of the array Array2:");
Scanf("%d", &n);
Printf("\n Enter sorted elements of array2:\n");
  For(i=0; i<n; i++)
    {
Scanf("%d", &array2[i]);
```

```
}
scanf("%d", &array2
i=0;
j=0;
while (i<m && j<n)
{
if (array1[i] < array2[i])
{
array3[k] = array1[i];
i++;
}
else
{
array3[k] = array2[j];
j++;
}
k++;
}
if (i>=m)
{
while (j<n)
{
array3[k] = array2[j];
j++;
k++;
}
}
if (j>=n)
{
while (i<m)
{
array3[k] = array1[i];
```

```
i++;
k++;
}
}
printf("\n After merging: \n");
for (i=0; i<m+n; i++)
    {
printf("%d", array3[i]);
    }
getch();
}
```

Output:

Enter the size of the array Array 1: 3

Enter sorted elements of array 1:

1 2 3

Enter the size of the array Array 2: 3

Enter sorted elements of array 2:

5 6 7

After merging

~~1 2 3~~

1
2
3
5
6
7

2. write a program to implement prim's Algorithm.

Program:

```
#include <stdio.h>
#include <conio.h>
int a, b, u, v, n, i, j, ne=1;
int visited[10]={0}, min, mincost=0, cost[10][10];
void main()
{
clrscr();
Printf("\n Enter the number of nodes:");
Scanf("%d", &n);
Printf("\n Enter the adjacency matrix:\n");
For(i=1; i<=n; i++)
For(j=1; j<=n; j++)
{
Scanf("%d", &cost[i][j]);
if(cost[i][j]==0)
cost[i][j]=999;
}
visited[1]=1;
Printf("\n");
while(ne<n)
{
For(i=1, min=999; i<=n; i++)
For(j=1; j<=n; j++)
if(cost[i][j]<min)
if
```

```
if (visited[i]!=0)
{
min=cost[i][j];
    a=u=i;
    b=v=j;
}
if (visited[u]==0 visited[v]==0)
{
Printf("\n Edge %d:(%d %d) cost:%d",mc++,a,b,min);
  min cost+=min;
    visited[b]=1;
}
cost[a][b]=cost[b][a]=999;
}
printf("\n minimum cost:%d",min cost);
    getch();
}
```

output.

```
Enter the number of node:6
Enter the adjacency matrix:
 0 3 1 6 0 0
 3 0 5 0 3 0
 1 5 6 5 6 4
 6 0 5 0 0 2
 0 3 6 0 0 6
 0 0 4 2 6 0
Edge 1:(1 3) cost:1
Edge 2:(1 2) cost:3
```

Edge 3: (2 5) cost: 3

Edge 4: (3 6) cost: 4

Edge 5 : (6, 4) cost: 2

minimum cost: 13