II

First Semester MCA (2020 SCHEME)
PRACTICAL EXAMINATION JUNE-JULY 2021
20MCA135 DATA STRUCTURE LAB

MURSHID P ABDUL RASHEED
~~LGE20MCA~~ LGE20MCA-2029
DATE: 30 June 2021
TIME: 1:00PM - 4:00 PM
BATCH II

① Program to merge two sorted arrays:

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
 int array1[10],array2[10],array3[20];
 int i,j,m,n,k=0;
 clrscr();
 printf("\n Enter the size of first array :");
 scanf("%d",&m);
 printf("\n Enter elements of first array:");
 for(i=0; i<m; i++)
 { scanf("%d",&array1[i]);
 }
 printf("\n Enter the size of second array :");
 scanf("%d",&n);
 printf("\n Enter elements of second array :");
 for(i=0;i<n;i++)
 { scanf("%d",&array2[i]);
 }
 i=j=0;
 while(i<m && j<n)
 {
   if (array1[i] < array2[i])
   { array3[k]= array1[i];
     i++;
   }
   ~~if(array2[i]<array1[~~
   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("\n%d", array3[i]);
}
getch();
}
```

ALGORITHM:

step 1 : Start

step 2 : Declare three arrays array1, array2 & array3.

step 3 : Read size of first array and read the values from the user. (size of array1 stored in m)

step 4 : Read size of second array and read values from the user. (size of array2 stored in n)

step 5 : ~~[illegible]~~ Declare variables i, j=0.

step 6 : Repeat step 7 to step 10 until i<m and j<n.

step 7 : if array[i]< array2[i] do step 8 else do step 9.

step 9 : array3[k] = array1[i].
i++. Go to step 10.

Step 9 : array3[k] = array2[j].
j++;

step 10 : k++.

step 11 : if i >= m do step 12 else go to step 14.

step 12: while j<n repeat step 13.
step 13: array3[k] = array2[j].
j++; k++.
step 14: if j>=n then do step 15 else go to step 17.
step 15: while i<m repeat step 16.
step 16: array3[k] = array1[i].
i++, k++.
step 17: Display third array array3.
step 18: stop.

EXPECTED OUTPUT:

Enter the size of first array: 3
Enter the elements of first array: 1 2 3

Enter the size of second array: 3
Enter the elements of second array: 4 5 6

After merging:
1
2
3
4
5
6

2. Implement Prims 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 no.of nodes:");
    scanf("%d",&n);
    printf("\n Enter the adjacency matrix:");
    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(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",ne++,a,b,min);
              mincost+=min;
              visited[b]=1;
            }
            cost[a][b]=cost[b][a]=999;
        }
    }
    printf("\n Minimum cost:%d\n",mincost);
    getch();
}
```

EXPECTED OUTPUT:

Enter the no. of nodes: 6

Enter the adjacency matrix:

```
0 3 1 6 0 0
3 0 5 0 3 0
1 5 0 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