Image

BATCH - I

Adithya R Sreedhar
ICE20 MCA-2002
Date : 30 June 21
Time : 9 30 - 10 30

FIRST SEMESTER MCA (2020 SCHEME)
PRACTICAL EXAMINATION JUNE - JULY
20MCA135 DATA STRUCTURE LAB

1. Sorting of an Integer Array
2. Implementing Prims Algorithm

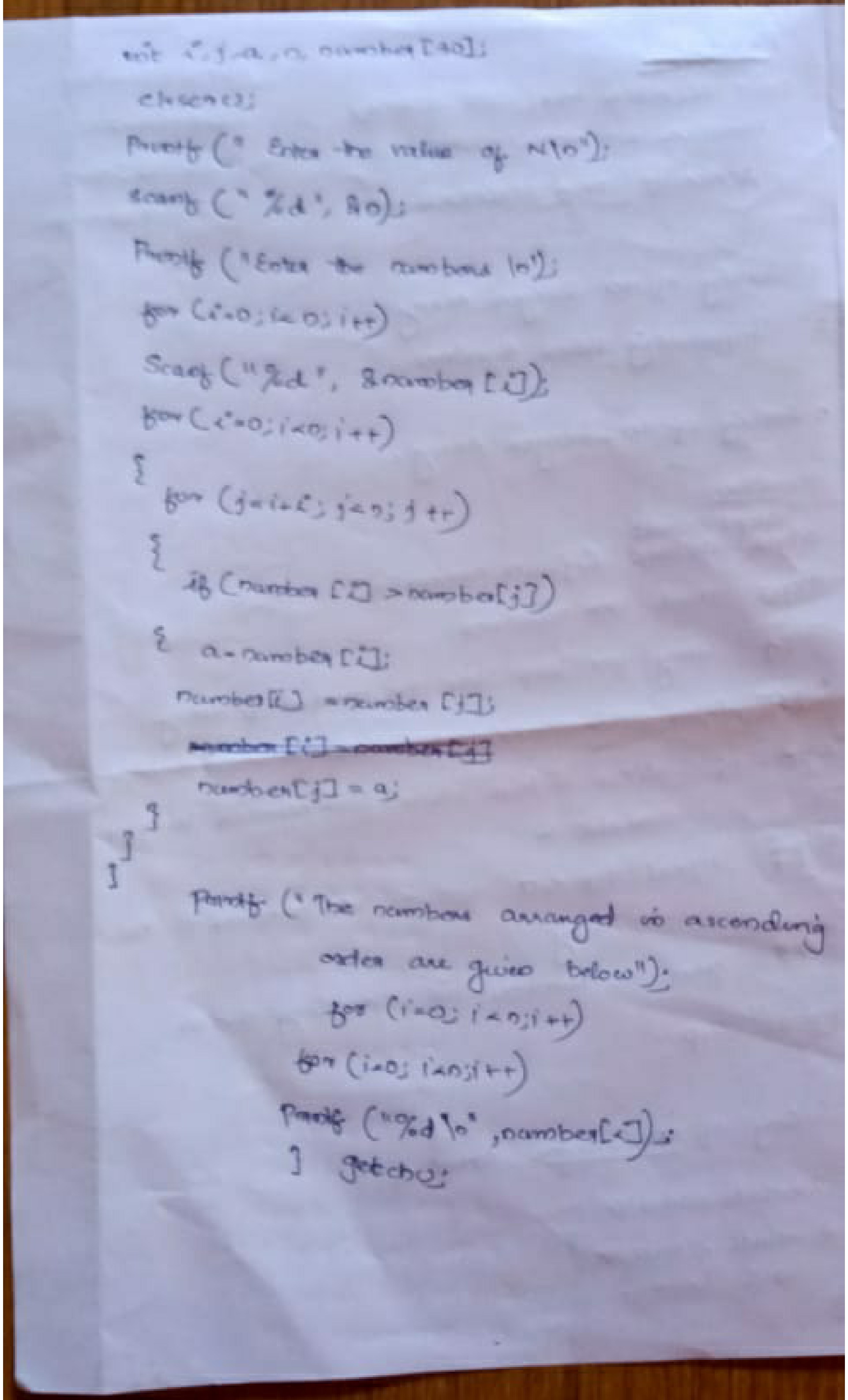
1 Sorting of an Integer Array.

Algorithm:

```
   step1 : start
step2 : declare i, j, a, n, number[30].
step3 : read ~~[illegible]~~ numbers
step4 : using for loop store the number of array
step5 : After sorting check first number and second number
step6 : if condition is true, slore first number to temp
            variable a.
step7 : repeat step 4,5 sort all the numbers in the
            Array.
Step8 : print the numbers in ascending order.
Step 9; stop.
```

1 Sorting of an Integer Array.

```
# include <stdio.h>
# include <conio.h>
   void main ()
{
```



```
int i,j,a,n, number[40];
 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+i; j<n; j++)
   {
     if (number [i] >number[j])
   {   a=number[i];
    number[i] =number [j];
    ~~number [i] =number[j]~~
     number[j] = a;
   }
  }
}
      Printf ("The numbers arranged in ascending
                 order are given below");
                  for (i=0; i<n;i++)
               for (i=0; i<n;i++)
              Printf ("%d \n" ,number[i]);
              } getch();
```

Output

```
Enter the value of ~~a~~ N:
 4
Enter the numbers;
 77
 11
 99
 22
The numbers arranged in ascending order are given
below.
 11
 22
 77
 99.
```

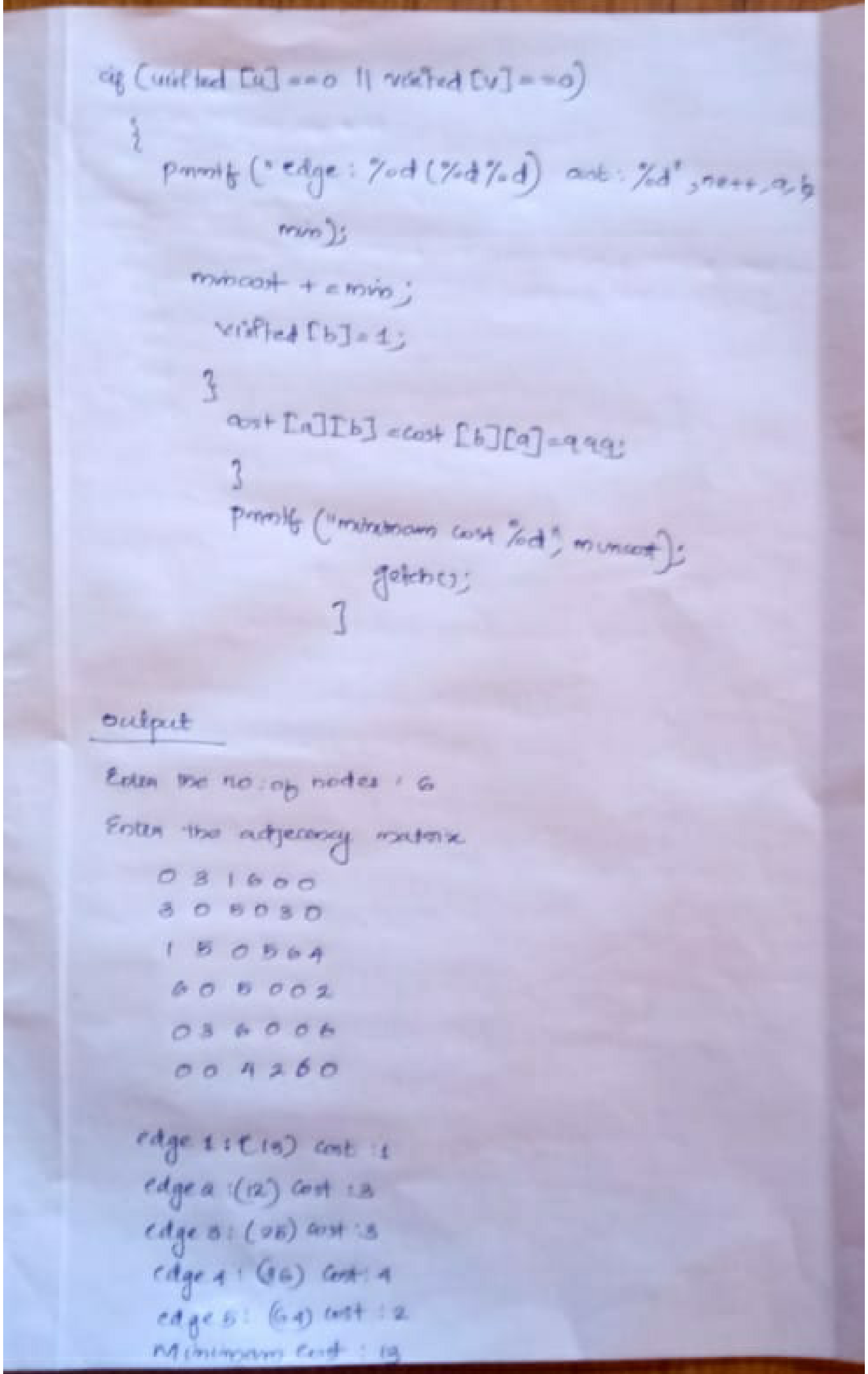
(2) Implementation of prims Algorithm.

Algorithm

```
Step1 : start
Step2 : Declare ~~no~~ ~~var~~ nodes (min, mincost =0, cost [10][10]).
Step3 : read the number of nodes.
Step4 : Enter Adjacency matrix using for loop.
Step5 : find the vertex that is ~~[illegible]~~ nearest to
            starting vertex
Step6 : check vertex set is empty
Step7 : output minimum spanning tree
            : else, exit
Step8 : stop
```

* 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()
  {
   printf("Enter the number of nodes");
    scanf("%d", &n);
    printf("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;
         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 ("edge : %d (%d%d) cost : %d", ne++, a, b
            min);
     mincost + = min;
      visited [b] = 1;
     }
      cost [a][b] =cost [b][a]=999;
      }
      printf ("minimum cost %d", mincost);
              getch();
          }
```

Output

```
Enter the no. of nodes : 6
Enter the adjecency 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 : (13) cost : 1
  edge 2 :(12) cost : 3
  edge 3 : (25) cost : 3
  edge 4 : (36) cost : 4
  edge 5 : (64) cost : 2
  Minimum cost : 13
```