BCSL404 : Analysis and Design of Algorithms Lab

Experiment No: 5

Design and implement C/C++ Program to obtain the Topological ordering of vertices in a given digraph.

```
#include<stdio.h>
#include<stdlib.h>

int visited[100], j, res[100];

void dfs(int u, int n, int a[][100])
  {
   int v;
   visited[u] = 1;
   for (v = 0; v < n - 1; v++)
        {
        if (a[u][v] == 1 && visited[v] == 0)
            dfs(v, n, a);
        }
       j += 1;
      res[j] = u;
   }

void topological_order(int n, int a[][100])
 {
   int i, u;
   for (i = 0; i < n; i++)
      visited[i] = 0;

   for (u = 0; u < n; u++)
     if (visited[u] == 0)
              dfs(u, n, a);
      return;
  }

 int main()
  {
   int a[100][100], n, i, j;

   printf("Enter number of vertices\n");
   scanf("%d", &n);

   printf("Enter the adjacency matrix -\n");
   for (i=0; i<n; i++)
   for (j=0; j<n; j++)
   scanf("%d",&a[i][j]);

   printf("\nTopological order:\n");

   topological_order(n, a);

   for (i = n; i >= 1; i--)
      printf("-->%d", res[i]);

   return 0;
  }
```

Output 1:

Demo: Topological Sorting Algorithm

Enter number of vertices: 5

Enter the adjacency matrix -
0 0 1 0 0
0 0 1 0 0
0 0 0 1 1
0 0 0 0 1
0 0 0 0 0

Topological order:
-->4-->1-->0-->2-->3

Output 2:

Enter number of vertices: 6
Enter the adjacency matrix -
0 0 0 0 0 0
1 0 0 0 0 0
1 0 0 0 0 0
0 1 0 0 0 0
0 0 0 0 0 0
1 1 1 1 1 0

Topological order:
-->5-->4-->3-->2-->1-->0
