PROGRAM 2

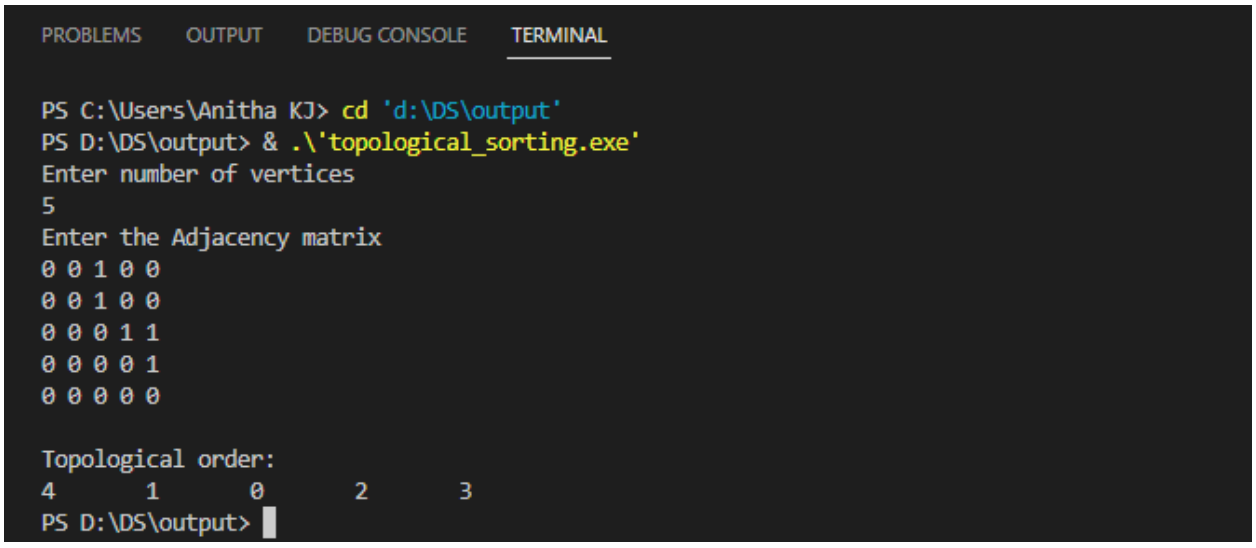
Write program to obtain the Topological ordering of vertices in a given digraph.

CODE:

```
#include<stdio.h>
#include<stdlib.h>
int visited[30], j, res[30];
void dfs(int u, int n, int Adj[30][30]) {
   int v;
   visited[u] = 1;
   for (v = 0; v < n - 1; v++) {
      if (Adj[u][v] == 1 && visited[v] == 0) {
         dfs(v, n, Adj);
      }
   }
   j += 1;
   res[j] = u;
}

void topological_order(int n, int Adj[30][30]) {
   int i, u;
   for (i = 0; i < n; i++) {
      visited[i] = 0;
   }
   j = 0;
   for (u = 0; u < n; u++) {
      if (visited[u] == 0) {
         dfs(u, n, Adj);
      }
   }
   return;
}
int main() {
   int Adj[30][30], 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",&Adj[i][j]);
 printf("\nTopological order:\n");
  topological_order(n, Adj);
    for (i = n; i >= 1; i--) {
        printf("%d\t", res[i]);
    }
    return 0;
}
```

OUTPUT:



OBSERVATION:

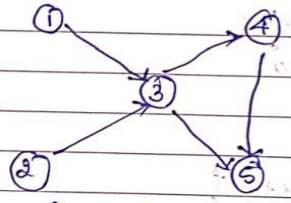
Topological Sorting.

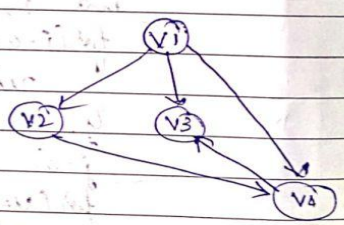
```
#include<stdio.h>
#include<stdlib.h>
int visited[30], j, res[30];

void dfs(int u, int n, int adj[30][30]){
int v;
    visited[u] = 1;
    for(v=0; v<n-1; v++){
        if (adj[u][v]==1 && visited[v]==0){
            dfs(v, n, adj);
        }
    }
    j++;
    res[j] = u;
}

void topological_order(int n, int adj[30][30]){
    int i, u;
    for(i=0; i<n; i++){
        visited[i] = 0;
    }
    j=0;
    for(u=0; u<n; u++){
        if (visited[u]==0){
            dfs(u, n, adj);
        }
    }
    return;
}
int main()
{
    int adj[30][30], 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", &Adj[i][j]);

printf("\n Topological Order:\n");

topological_order(n, Adj);
for(i=n-1; i>=1; i--){
    printf("%d\t", res[i]);
}
return 0;
}
```

Output :-

Graph :





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 4

Enter number of vertices
4
Adjacency matrix
0 1 1 1
0 0 0 1
0 0 0 0
0 0 1 0

22/6/23