

ADA-LAB-2

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

Code-

```
#include <stdio.h>
#include <stdlib.h>
#define MAX_VERTICES 100
typedef struct {
    int vertices[MAX_VERTICES];
    int count;
} Stack;
void initialize(Stack* stack) {
    stack->count = 0;
}
int isEmpty(Stack* stack) {
    return (stack->count == 0);
}
void push(Stack* stack, int value) {
    stack->vertices[stack->count++] = value;
}
int pop(Stack* stack) {
    if (isEmpty(stack)) {
        printf("Error: Stack underflow\n");
        exit(0);
    }
    return stack->vertices[--stack->count];
}
void topologicalSortDFS(int vertex, int** graph, int* visited, Stack* stack, int numVertices) {
    visited[vertex] = 1;
```

```

int i;
for (i = 0; i < numVertices; i++) {
    if (graph[vertex][i] && !visited[i]) {
        topologicalSortDFS(i, graph, visited, stack, numVertices);
    }
}

push(stack, vertex + 1);
}

void topologicalSort(int** graph, int numVertices) {
    Stack stack;

    int visited[MAX_VERTICES];

    int i;

    initialize(&stack);

    for (i = 0; i < numVertices; i++) {
        visited[i] = 0;
    }

    for (i = 0; i < numVertices; i++) {
        if (!visited[i]) {
            topologicalSortDFS(i, graph, visited, &stack, numVertices);
        }
    }

    printf("Topological Ordering of Vertices:\n");

    while (!isEmpty(&stack)) {
        printf("%d ", pop(&stack));
    }

    printf("\n");
}

int main() {
    int numVertices, i, j;

```

```

printf("Enter the number of vertices in the graph: ");
scanf("%d", &numVertices);

int** graph = (int**)malloc(numVertices * sizeof(int*));
for (i = 0; i < numVertices; i++) {
    graph[i] = (int*)malloc(numVertices * sizeof(int));
}

printf("Enter the adjacency matrix of the graph:\n");
for (i = 0; i < numVertices; i++) {
    for (j = 0; j < numVertices; j++) {
        scanf("%d", &graph[i][j]);
    }
}

topologicalSort(graph, numVertices);
return 0;
}

```

```

Enter the number of vertices in the graph: 4
Enter the adjacency matrix of the graph:
0
1
1
1
0
0
0
0
0
0
1
0
0
1
0
0
Topological Ordering of Vertices:
1 4 3 2

```

Write a C-program to perform topological sort using DFS method.

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

```
#include <conio.h>
```

```
void source_removal (int n, int a [10][10]) {
```

```
int i, j, k, u, v, top, s[10], t[10], indeg[10], sum;
```

```
for (i=0; i<n; i++) {
```

```
sum=0;
```

```
for (j=0; j<n; j++)
```

```
sum+= a[j][i];
```

```
indeg[i]=sum;
```

```
}
```

```
top = -1;
```

```
for (i=0; i<n; i++) {
```

```
if (indeg[i]==0)
```

```
s[++top] = i;
```

```
}
```

```
k=0;
```

```
while (top != -1) {
```

```
u = s[top--];
```

```
t[k++] = u;
```

```
for (v=0; v<n; v++) {
```

```
if (a[u][v]==1) {
```

```
indeg[v] = indeg[v] - 1;
```

```
if (indeg[v]==0)
```

```
s[++top] = v;
```

```
}
```

```
}
```

```
}
```

```
printf ("Topological order: ");
```

```
for (i=0; i<n; i++)
```

```
printf ("%d", t[i]+1);
```

```
}
```

```
void main ()
```



```

$
int i, j, a[10][10], n;
printf("Enter number of nodes \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]);
source - removal(n, a);
getch();
}

```

Output

Enter number of nodes 5

Enter 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 : 2 1 3 4 5



