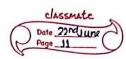
```
Q) Write program to obtain the Topological ordering of vertices in a given
digraph.
Code-
#include <stdio.h>
#include <stdlib.h>
#define MAX_VERTICES 100
typedef struct {
int vertices[MAX_VERTICES];
int count;
} Stack;
void initialize(Stack* stack) {
stack->count = 0;
}
int isEmpty(Stack* stack) {
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;
}</pre>
```

```
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
1
1
0
0
Topological Ordering of Vertices:
1 4 3 2
```



CONTRACTOR OF THE PARTY OF THE	
	Write a c-program to perform topological sout
	The state of the s
	# include <stdio.h></stdio.h>
	# Include /conio·h>
	void source removal (int n, int a [10][10]) &
	int i, i, K, u, v, top, S[10], t[10], indea [10] cum
	tor(1=0;14+)&
	for (j=0;j <n;j++) (nollowom)="" -="" 21="" mou<="" td=""></n;j++)>
	for (j=0; j < n; j++) (n al n v ocasi- 2 mos
	Sum+= a Ciacia;
	indeg Eij= sum;
	3
	top = -1;
	for (i=0; i< n; i+t) & so had de padama notali
	i'f lindeg [i]==0)
	S[++ top] = i;
	2
	K=0;
	while (top) = -1) &
	u=s[+op];
	t [k++] = u; E 1 & - rabio Justinosoft
1	for(v=0; V <n; td="" v++)="" {<=""></n;>
	3(1== [v3cu3 p] +1
1-3	Indea [v] = indea [v]-1;
	if (indeg [v]==0)
1,00	SC++10p]=V;
1.8	2
	3
	3
	Priot f ("Topological order:");
	for (i=0; ic n; i++) Print (" 1.d", (t [i]+1));
CS	3
C2 ,	Solided with DamScanner

