TOPOLOGICAL SORTING

AIM:

 To write a C program to perform the Topological sorting

ALGORITHM:

 Step 1: Start

Step 2: Create a text document named “graph.txt” and give the input of topological sorting

Step 3: Read the no. of vertices from the user side

Step 4: Find the indegree for each vertex

Step 5: Remove the vertices, whose indegree is zero

Step 6: Decrement the indegree of the vertices, at the end of the outgoing edges

Step 7: Repeat Step 2, until all the vertices are removed

Step 8: Stop

PROGRAM CODING:

#include<stdio.h>

#include<conio.h>

#define MAX 30

int stack [MAX],top=-1;

int push(int v);

int pop(int \*v);

int topologicalorder(int mat[MAX][MAX],int n,int order[MAX]);

void main()

{

int i,j,n,mat[MAX][MAX],order[MAX];

FILE \*f1;

printf("enter the number of vertices:");

scanf("%d",&n);

if((f1=fopen("c:\\graph.txt","r"))==NULL)

{

fprintf(stderr,"cannot open input file.\n");

return;

}

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

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

{

fscanf(f1,"%d",&mat[i][j]);

}

fclose(f1);

printf("\n success:%d\n",topologicalorder(mat,n,order));

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

printf("%c\t",order[i]+65);

}

int push(int v)

{

if(top==MAX-1)

return 0;

else

stack[++top]=v;

return -1;

}

int pop(int \*v)

{

if(top==-1)

return 0;

else

\*v=stack[top--];

return -1;

}

int topologicalorder(int mat[MAX][MAX],int n,int order[MAX])

{

int indegree[MAX],v,i,k,m;

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

{

indegree[i]=0;

for(k=0;k<n;k++)

if(mat[k][i]==1)

indegree[i]++;

if(indegree[i]==0)

push(i);

}

m=0;

while(pop(&v))

{

order[m++]=v;

for(k=0;k<n;k++)

if(mat[v][k]==1&&indegree[k]>0)

{

indegree[k]--;

if(indegree[k]==0)

push(k);

}

}

getch();

return(i==n);

}

OUTPUT:

The tex in the file graph.txt is

0 0 1 1 0 0

0 0 0 1 1 0

0 0 0 1 0 1

0 0 0 0 0 1

0 0 0 0 0 1

0 0 0 0 0 0

The program displays the following output

enter the number of vertices:6

 success:1

B E A C D F