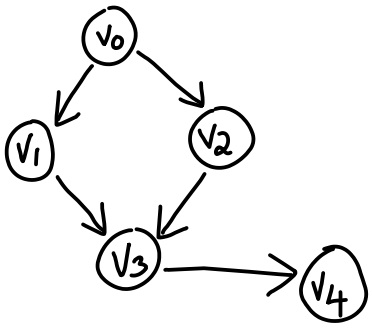
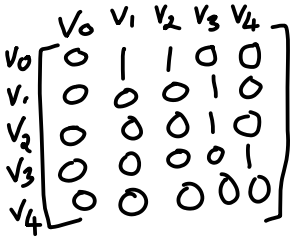
AIM: Write program to obtain the Topological ordering of vertices in a given digraph.

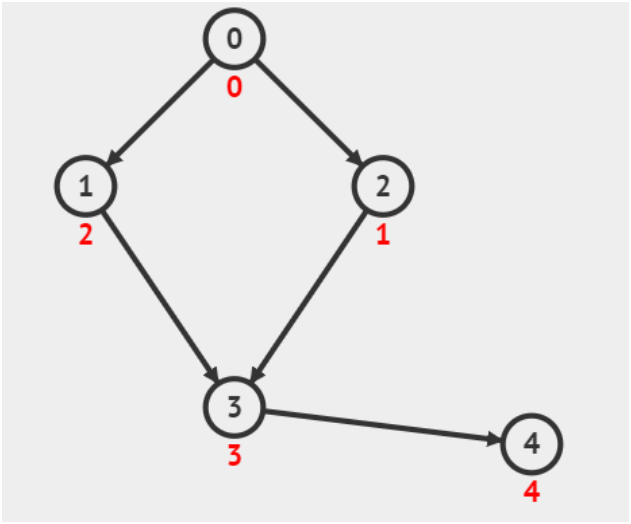
```
#include<stdio.h>
  int temp[10],k=0;

void topo(int n,int indegree[10],int a[10][10])
  {
  int i,j;

  for(i=1;i<=n;i++)
   {
          if(indegree[i]==0)
           {
           indegree[i]=1;
            temp[++k]=i;
                for(j=1;j<=n;j++)
                  {
                   if(a[i][j]==1&&indegree[j]!=-1)
                    indegree[j]--;
                  }
                  i=0;
            }
     }
  }

 void main()
  {
  int i,j,n,indegree[10],a[10][10];
  printf("enter the number of vertices:");
  scanf("%d",&n);
```

```
  for(i=1;i<=n;i++)
  indegree[i]=0;

  printf("\n enter the adjacency matrix\n");
  for(i=1;i<=n;i++)
  for(j=1;j<=n;j++)
  {
    scanf("%d",&a[i][j]);
    if(a[i][j]==1)
    indegree[j]++;
  }

  topo(n,indegree,a);

  if(k!=n)
  printf("topological ordering is not possible\n");

else
  {
    printf("\n topological ordering is :\n");
    for(i=1;i<=k;i++)
    printf("v%d\t",temp[i]);
   }
 }
```

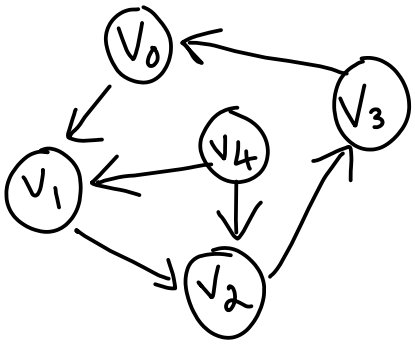


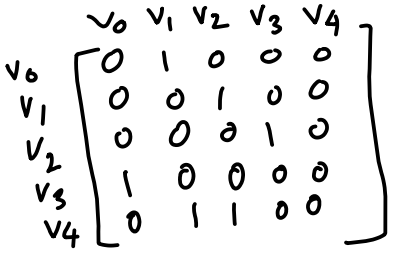



```
D:\codes\ADA LAb6.exe
enter the number of vertices:5

 enter the adjacency matrix
0 1 1 0 0
0 0 0 1 0
0 0 0 1 0
0 0 0 0 1
0 0 0 0 0

 topological ordering is :
v0      v1      v2      v3      v4
--------------------------------
Process exited after 34.63 seconds with return value 0
Press any key to continue . . .
```

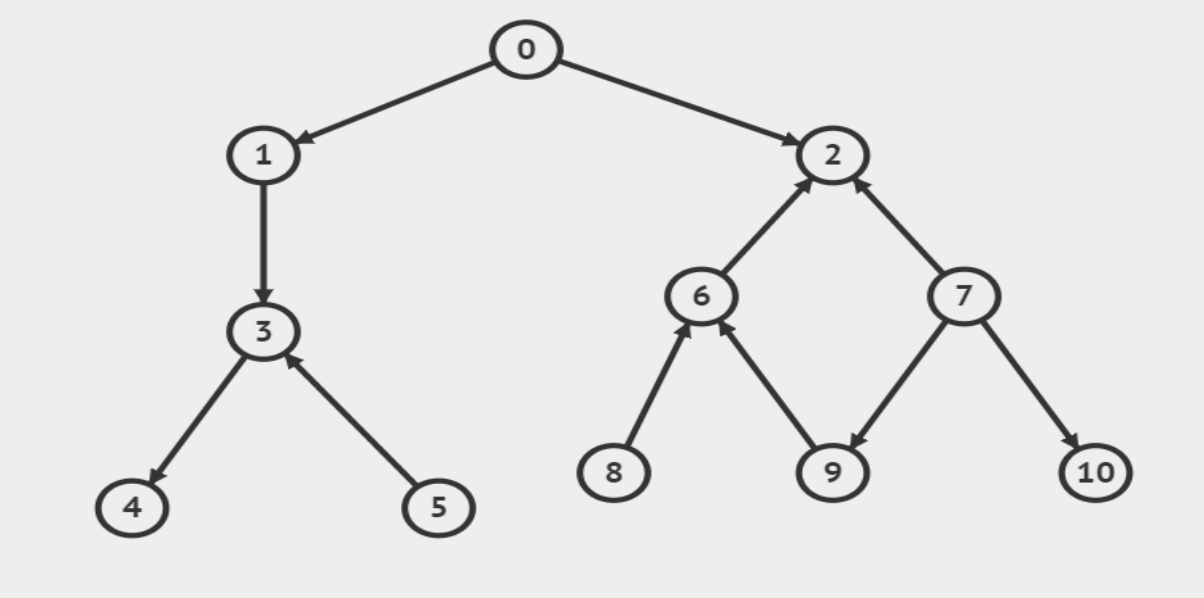


0 1 1 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0
0 0 1 0 0 0 0 0 1 0 1
0 0 0 0 0 0 0 0 0 0 0

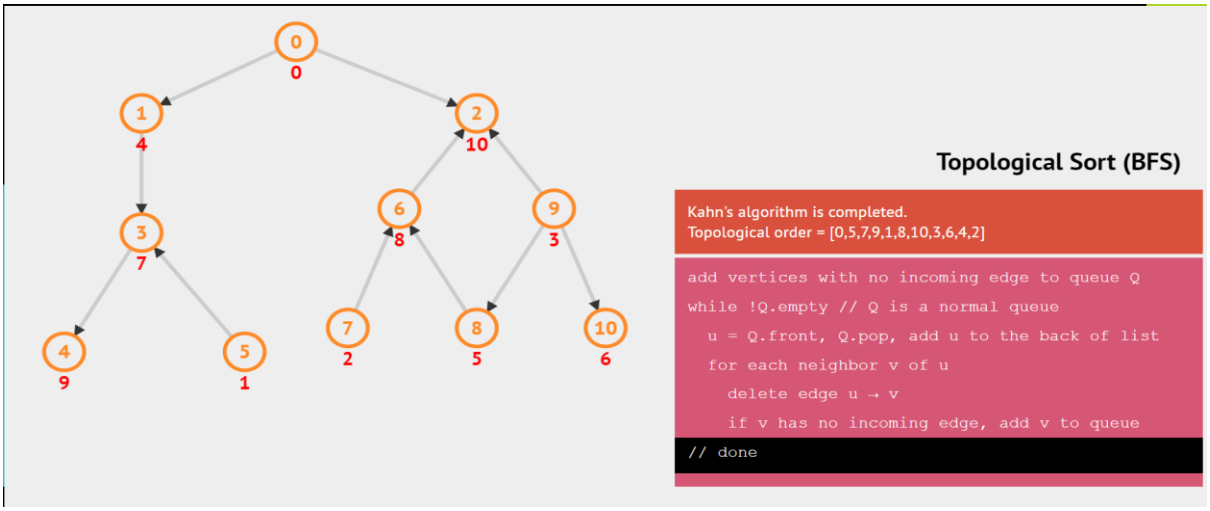





```
D:\codes\ADA LAb6.exe
enter the number of vertices:5

 enter the adjacency matrix
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
1 0 0 0 0
0 1 1 0 0
topological ordering is not possible

--------------------------------
Process exited after 34.59 seconds with return value 0
Press any key to continue . . .
```



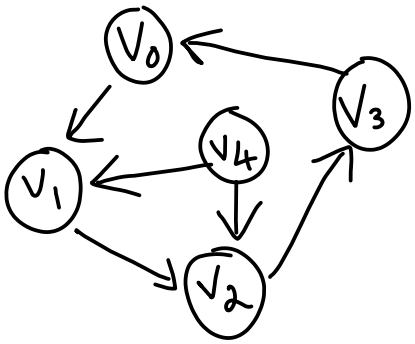

```
D:\codes\ADA LAb6.exe
enter the number of vertices:11

 enter the adjacency matrix
0 1 1 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0
0 0 1 0 0 0 0 0 1 0 1
0 0 0 0 0 0 0 0 0 0 0

 topological ordering is :
v0      v1      v5      v3      v4      v7      v9      v8      v6      v2      v10
--------------------------------
Process exited after 607.6 seconds with return value 0
Press any key to continue . . .
```

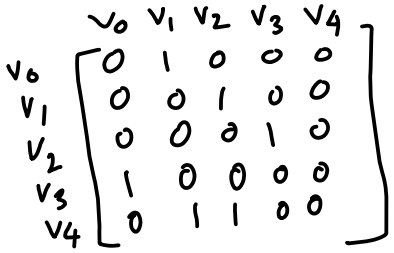


Using DFS Technique

```
#include<stdio.h>
  int res[10],top=0,s[10];
  void dfs(int v,int n,int a[10][10]){
    s[v]=1;
    for(int i=1;i<=n;i++)
    if(s[i]==0&&a[v][i]==1) dfs(i,n,a);
  top++;
  res[top]=v;
  }

void topo(int n,int a[10][10])
  {
  for(int i=1;i<=n;i++){
    s[i]=0;
  }
  top=0;
  for(int i=1;i<=n;i++){
    if(s[i]==0)dfs(i,n,a);
  }

  }

 void main()
  {
  int i,j,n,vertices[10],a[10][10];
  printf("enter the number of vertices:");
  scanf("%d",&n);
  for(i=1;i<=n;i++)
  vertices[i]=0;

  printf("\n enter the adjacency matrix\n");
  for(i=1;i<=n;i++)
  for(j=1;j<=n;j++)
  {
     scanf("%d",&a[i][j]);

  }

  topo(n,a);

  if(top!=n)
  printf("topological ordering is not possible\n");

else
  {
```

```
    printf("\n topological ordering is :\n");
    for(int i=n;i>0;i--){
      printf("v%d\t",res[i]);
    }
  }
}
```






```
enter the number of vertices:5

 enter the adjacency matrix
0 1 1 0 0
0 0 0 1 0
0 0 0 1 0
0 0 0 0 1
0 0 0 0 0

 topological ordering is :
v1      v3      v2      v4      v5
--------------------------------
Process exited after 23.07 seconds with return value 0
Press any key to continue . . .
```