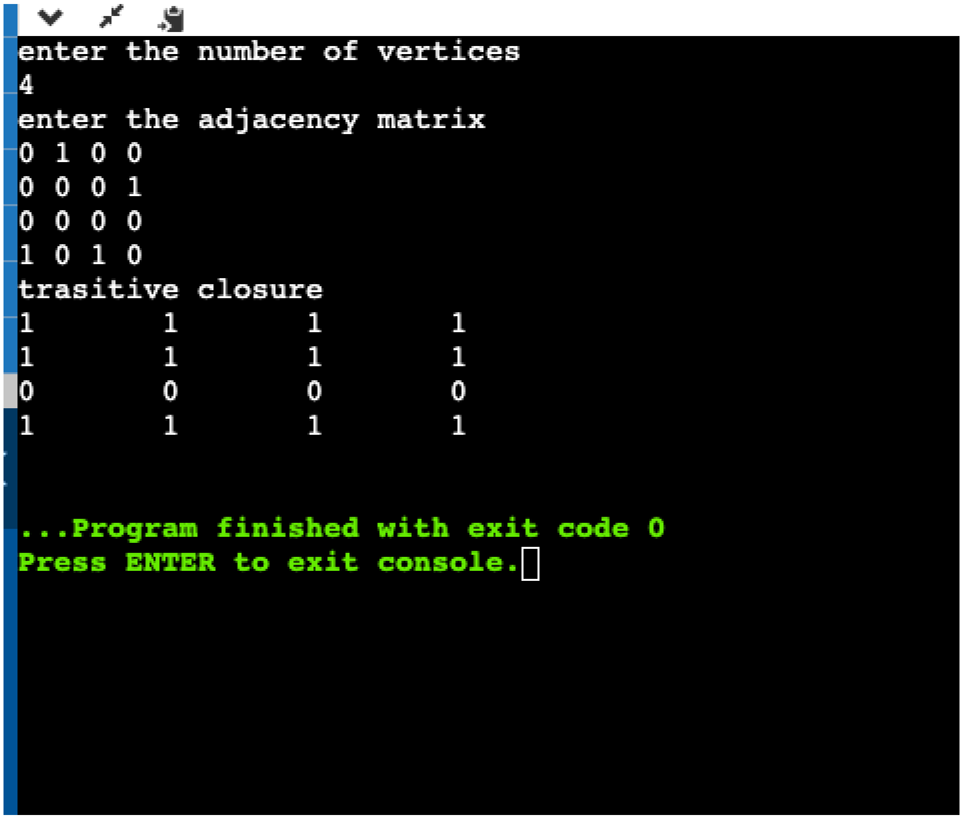
ADA LAB TEST 2

Implement Warshall’s algorithm using dynamic programming.

```
#include<stdio.h>
#include<conio.h>
int n,a[10][10],p[10][10];
void warshall(int n,int a[10][10],int p[10][10])
{
int i,j,k;
for(i=0;i<n;i++)
for(j=0;j<n;j++)
p[i][j]=a[i][j];
for(k=0;k<n;k++)
for(i=0;i<n;i++)
for(j=0;j<n;j++)
if((p[i][j]==0) && (p[i][k]==1 && p[k][j]==1))
p[i][j]=1;

}
void main()
{
int i,j;
printf("enter the 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",&a[i][j]);
}
}
warshall(n,a,p);
printf("trasitive closure\n");
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
printf("%d\t",p[i][j]);
}
printf("\n");
}
}
```



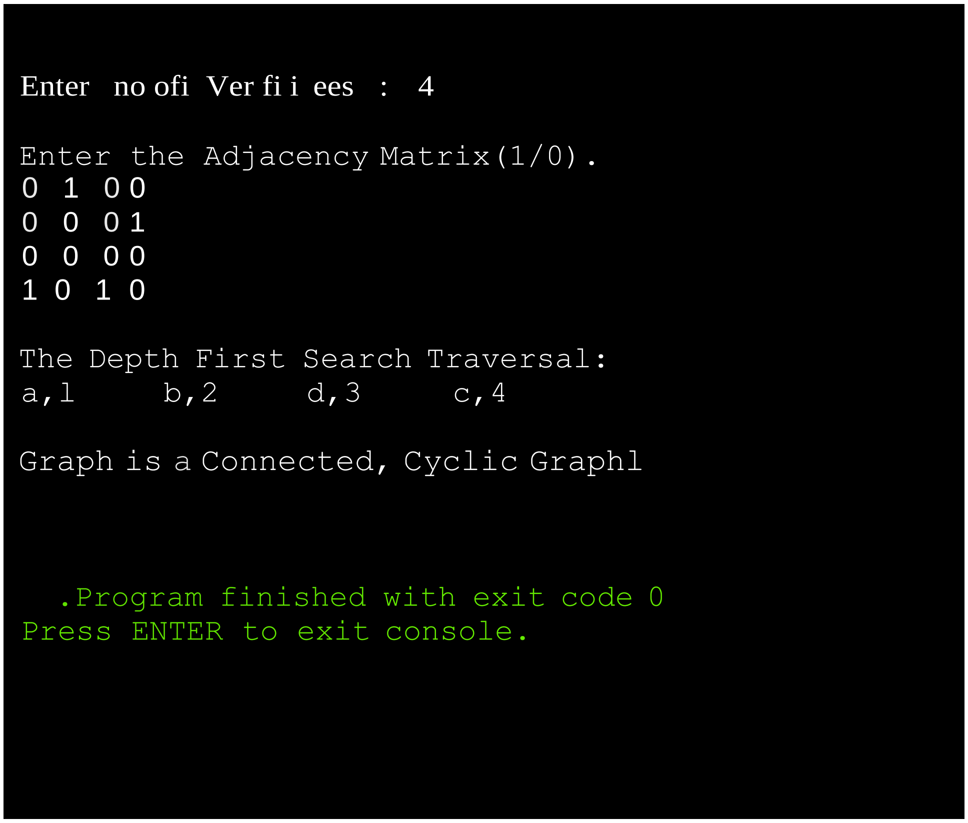
```
enter the number of vertices
4
enter the adjacency matrix
0 1 0 0
0 0 0 1
0 0 0 0
1 0 1 0
trasitive closure
1       1       1       1
1       1       1       1
0       0       0       0
1       1       1       1


...Program finished with exit code 0
Press ENTER to exit console.
```

Modification: By using path matrix obtained detect the cycle in the graph.

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

/****Global Variables****/
int A[20][20],visited[20],count=0,n;
int seq[20],connected=1,acyclic=1;

/****DFS Function Declaration****/
void DFS();

/****DFSearch Function Declaration****/
```

```
void DFSearch(int cur);

/****Main Function****/
int main()
   {
    int i,j;

    printf("\nEnter no of Vertices: ");
    scanf("%d",&n);

    printf("\nEnter the Adjacency Matrix(1/0):\n");
    for(i=1;i<=n;i++)
            for(j=1;j<=n;j++)
        scanf("%d",&A[i][j]);

    printf("\nThe Depth First Search Traversal:\n");

    DFS();

    for(i=1;i<=n;i++)
        printf("%c,%d\t",'a'+seq[i]-1,i);

    if(connected && acyclic)    printf("\n\nIt is a Connected, Acyclic Graph!");
    if(!connected && acyclic)   printf("\n\nIt is a Not-Connected,Acyclic Graph!");
    if(connected && !acyclic)   printf("\n\nGraph is a Connected, Cyclic Graph!");
    if(!connected && !acyclic) printf("\n\nIt is a Not-Connected, Cyclic Graph!");

    printf("\n\n");
    return 0;
   }

/****DFS Function Definition****/
void DFS()
    {
    int i;
    for(i=1;i<=n;i++)
        if(!visited[i])
```

```
          {
        if(i>1) connected=0;
        DFSearch(i);
              }
    }

/****DFSearch Function Definition****/
void DFSearch(int cur)
    {
    int i,j;
    visited[cur]=++count;

        seq[count]=cur;
        for(i=1;i<count-1;i++)
                if(A[cur][seq[i]])
                   acyclic=0;

    for(i=1;i<=n;i++)
        if(A[cur][i] && !visited[i])
           DFSearch(i);
}
```



```
Enter  no ofi  Ver fi i  ees  :  4

Enter the Adjacency Matrix(1/0).
0  1  0 0
0  0  0 1
0  0  0 0
1  0  1  0

The Depth First Search Traversal:
a,l     b,2     d,3     c,4

Graph is a Connected, Cyclic Graphl



  .Program finished with exit code 0
Press ENTER to exit console.
```