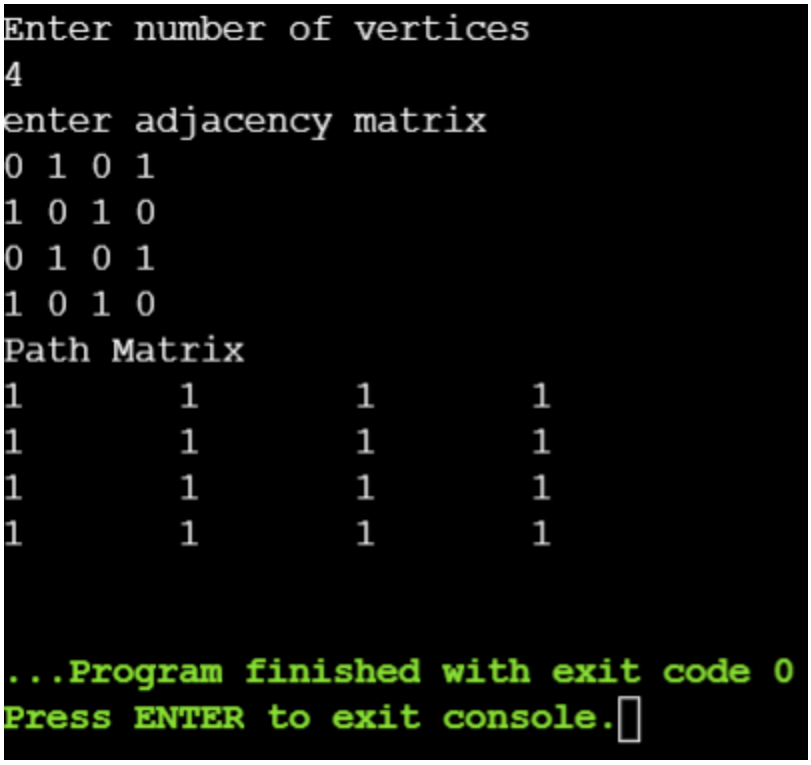
ADA LAB TEST 2 WARSHALLS PROGRAM

PROGRAM :-

```c
#include<stdio.h>

void warshalls();

int a[10][10], p[10][10], i,j,k,n;

void main()
{
printf("Enter number of vertices\n");
scanf("%d",&n);
printf("enter adjacency matrix\n");
for(i=1;i<=n;i++)
  {
    for(j=1;j<=n;j++)
   {
    scanf("%d",&a[i][j]);
  }
}
warshalls();
printf("Path Matrix\n");
for(i=1;i<=n;i++)
   {
    for(j=1;j<=n;j++)
    {
      printf("%d\t",p[i][j]);
     }
     printf("\n");
  }
getch();
}
```

```
void warshalls()
{
for(i=1;i<=n;i++)
  {
   for(j=1;j<=n;j++)
   {
    p[i][j]=a[i][j];
   }
}
    for(k=1;k<=n;k++)
    {
      for(i=1;i<=n;i++)
       {
         for(j=1;j<=n;j++)
           {
             if((p[i][j]!=1) && (p[i][k]==1 && p[k][j]==1))
            p[i][j]=1;
         }
      }
   }
}
```

OUTPUT:-



MODIFICATION PROGRAM :-

```c
#include<stdio.h>
#include<conio.h>
void warshalls();
int a[10][10], p[10][10], i,j,k,n,count=0;

void main()
{
    printf("Enter number of vertices\n");
    scanf("%d",&n);
    printf("Enter adjacency matrix\n");
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=n;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    warshalls();
    printf("\nPath Matrix\n");
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=n;j++)
        {
            printf("%d ",p[i][j]);
        }
        printf("\n");
    }
    for(i=1;i<=n;i++)
    {
        if(p[i][i]==1)
```

```
        {
            count++;
        }
    }
    if(count==0)
    {
        printf("\nGraph does not contain any cycle");
    }
    else
    {
        printf("\nGraph contains cycle with vertices:\n");
        for(i=1;i<=n;i++)
        {
            if(p[i][i]==1)
            {
               printf("%d ",i);
            }
        }
    }

    getch();
}
void warshalls()
{
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=n;j++)
        {
            p[i][j]=a[i][j];
        }
```

```
    }
    for(k=1;k<=n;k++)
    {
        for(i=1;i<=n;i++)
        {
            for(j=1;j<=n;j++)
            {
                if((p[i][j]!=1) && (p[i][k]==1 && p[k][j]==1))
                {
                    p[i][j]=1;
                }
            }
        }
    }
}
```

```
Enter number of vertices
4
Enter adjacency matrix
0 1 0 1
1 0 1 0
0 1 0 1
1 0 1 0

Path Matrix
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1

Graph contains cycle with vertices:
1 2 3 4

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

----------x----------