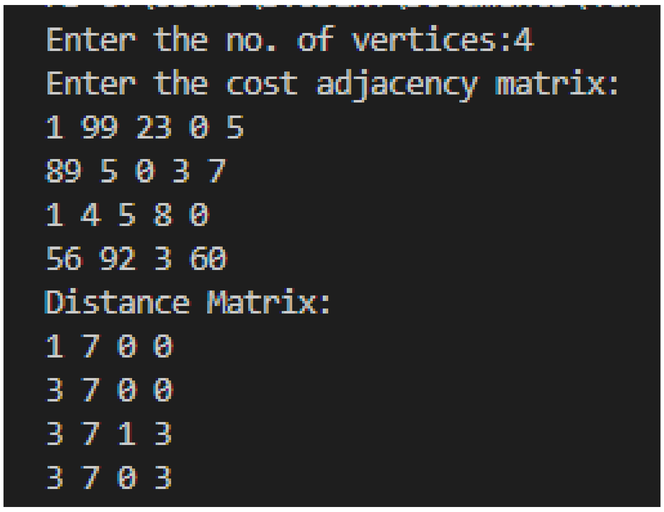
9/04/2025 – LAB 3

1. Implement all pair shortest paths problem using Floyd's algorithm

```c
#include <stdio.h>
int a[10][10],D[10][10],n;
void floyd(int [][10],int);
int min(int,int);
int main()
{
printf("Enter the no. of vertices:");
scanf("%d",&n);
printf("Enter the cost adjacency matrix:\n");
int i,j;
for(i=0;i<n;i++){
for(j=0;j<n;j++){
scanf("%d",&a[i][j]);
}
}
floyd(a,n);
printf("Distance Matrix:\n");
for(i=0;i<n;i++){
for(j=0;j<n;j++){
printf("%d ",D[i][j]);
}
printf("\n");
}
return 0;
}

void floyd(int a[][10],int n){
int i,j,k;
for(i=0;i<n;i++){
for(j=0;j<n;j++){
D[i][j]=a[i][j];
}
}

for(k=0;k<n;k++){
for(i=0;i<n;i++){
for(j=0;j<n;j++){
D[i][j]=min(D[i][j],(D[i][k]+D[k][j]));
}
}
}
}
```

```
int min(int a,int b){
if(a<b){
return a;
}else{
return b;
}
}
```

OUTPUT:



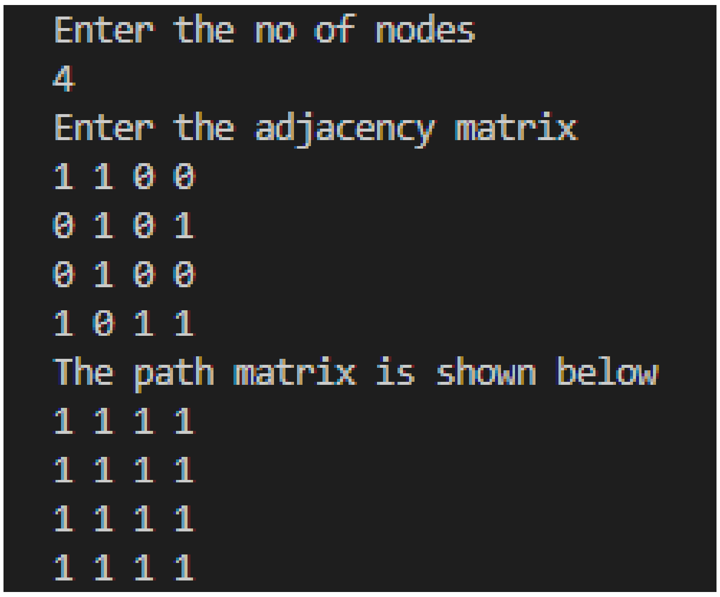
```
Enter the no. of vertices:4
Enter the cost adjacency matrix:
1 99 23 0 5
89 5 0 3 7
1 4 5 8 0
56 92 3 60
Distance Matrix:
1 7 0 0
3 7 0 0
3 7 1 3
3 7 0 3
```

2. Compute the transitive closure of a given directed graph using Warshall’s algorithm

```
#include<stdio.h>
int n;
int a[10][10];
int p[10][10];

void write_data()
{
int i,j;
printf("The path matrix is shown below\n");
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
printf("%d", p[i][j]);
printf(" ");
}
printf("\n");
}
}

void read_data()
{
int i,j;
printf("Enter the no of nodes\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]);
}
}
}

void path_matrix()
{
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][k]==1 & p[k][j]==1))
p[i][j]=1;
}
}
}
}

void main( )
{
read_data();
path_matrix();
write_data();
}
```

OUTPUT:



```
Enter the no of nodes
4
Enter the adjacency matrix
1 1 0 0
0 1 0 1
0 1 0 0
1 0 1 1
The path matrix is shown below
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
```