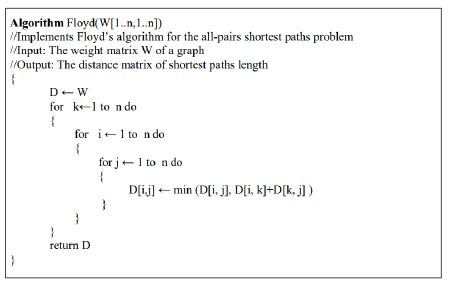
**Program 3:**

1. Design and implement C/C++ Program to solve All-Pairs Shortest Paths problem using Floyd’s algorithm.
2. Design and implement C/C++ Program to find the transitive closure using Warshal’s algorithm.

**Algorithm:**



**Code:**

#include<stdio.h>

int min(int a, int b)

{

return(a<b?a:b);

}

void floyd(int D[][10],int n)

{

for(int k=1;k<=n;k++)

for(int i=1;i<=n;i++)

for(int j=1;j<=n;j++)

D[i][j]=min(D[i][j],D[i][k]+D[k][j]);

}

int main()

{

int n, cost[10][10];

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

scanf("%d",&n);

printf("Enter the cost matrix \n");

for(int i=1;i<=n;i++)

for(int j=1;j<=n;j++)

scanf("%d",&cost[i][j]);

floyd(cost,n);

printf("All pair shortest path \n");

for(int i=1;i<=n;i++)

{

for(int j=1;j<=n;j++)

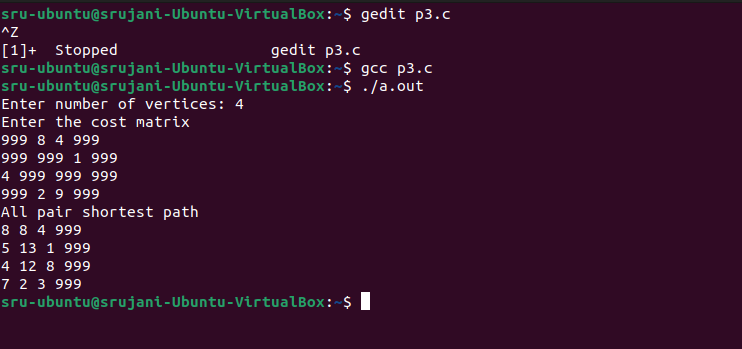
printf("%d ",cost[i][j]);

printf("\n");

}

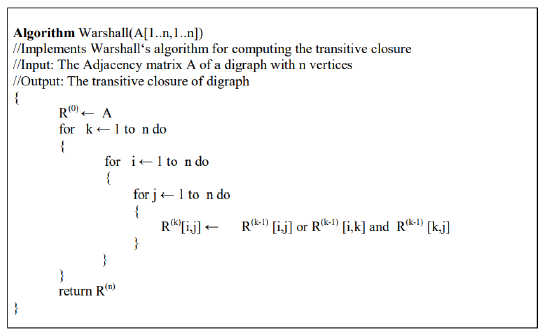
}

**Output:**

****

Program 3b

**Algorithm:**

****

**Code:**

#include<stdio.h>

void warshal(int A[][10], int n)

{

for(int k=1;k<=n;k++)

for(int i=1;i<=n;i++)

for(int j=1;j<=n;j++)

A[i][j]=A[i][j] || (A[i][k]&&A[k][j]);

}

void main()

{

int n,adj[10][10];

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

scanf("%d",&n);

printf("Enter the adjacency matrix: \n");

for(int i=1;i<=n;i++)

for(int j=1;j<=n;j++)

scanf("%d",&adj[i][j]);

warshal(adj,n);

printf("Transitive closure of the given graph is \n");

for(int i=1;i<=n;i++)

{

for(int j=1;j<=n;j++)

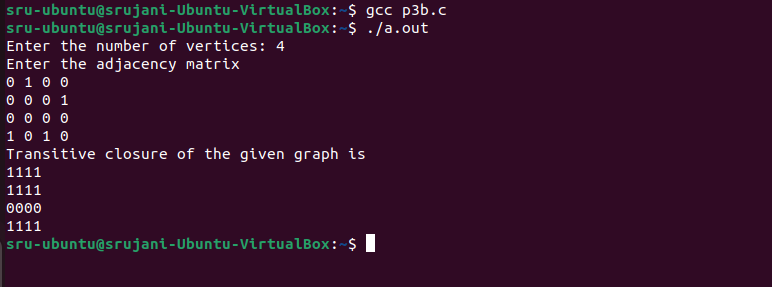
printf("%d",adj[i][j]);

printf("\n");

}

}

**Output:**

****