PROGRAM 11

Implement Warshall’s algorithm using dynamic programming.

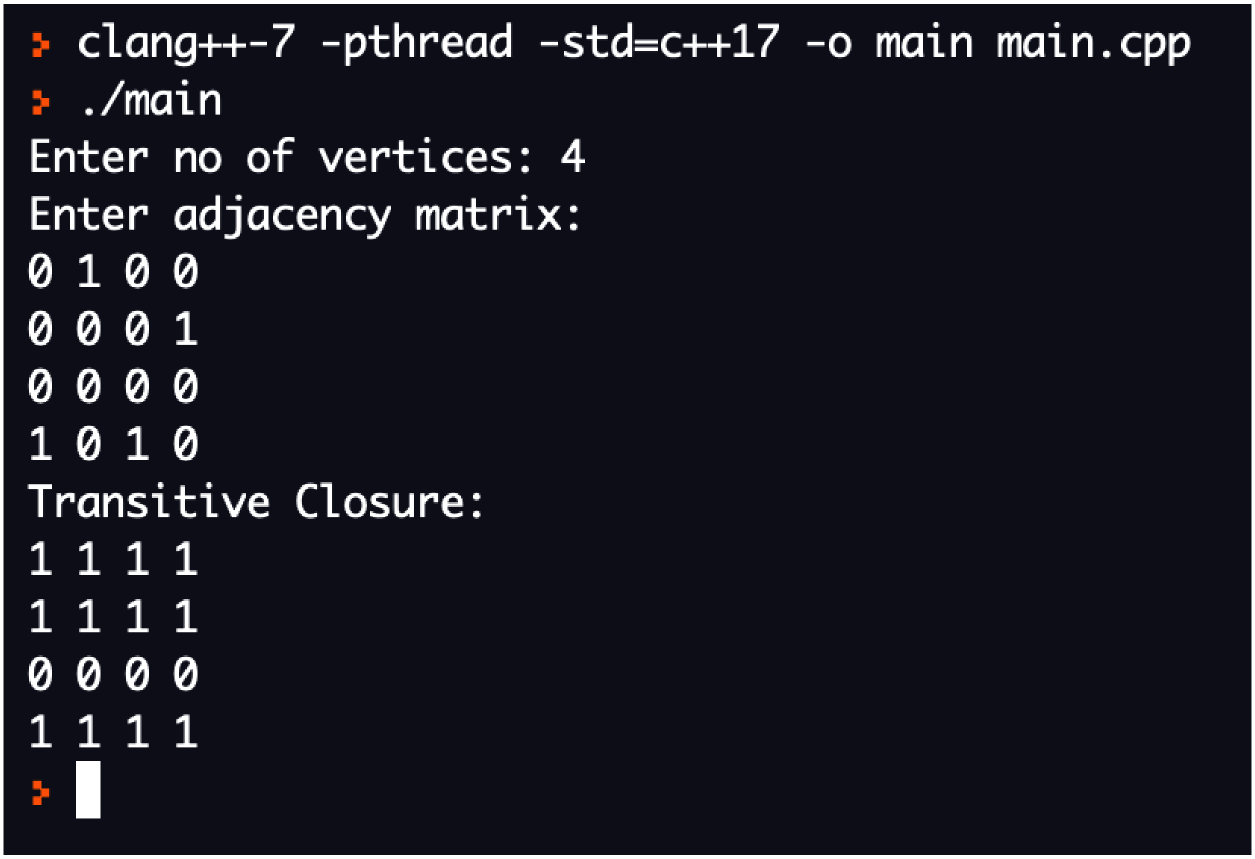
//Code

```
#include<iostream>
using namespace std;
int a[10][10],r[10][10][10];

void warshall(int n){
  int k=0;
  for(int i=1;i<=n;i++)
     for(int j=1;j<=n;j++)
        r[k][i][j]=a[i][j];
  for(k=1;k<=n;k++)
     for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
           r[k][i][j]=r[k-1][i][j] || (r[k-1][i][k] && r[k-1][k][j]);
}

int main(){
  int n;
  cout<<"Enter no of vertices: ";
  cin>>n;
  cout<<"Enter adjacency matrix: ";
  for(int i=1;i<=n;i++)
     for(int j=1;j<=n;j++)
        cin>>a[i][j];
  warshall(n);
  cout<<"Transitive Closure: "<<endl;
  for(int i=1;i<=n;i++){
     for(int j=1;j<=n;j++)
        cout<<r[n][i][j]<<" ";
     cout<<endl;
  }
}
```

//Output



```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
Enter no of vertices: 4
Enter adjacency matrix:
0 1 0 0
0 0 0 1
0 0 0 0
1 0 1 0
Transitive Closure:
1 1 1 1
1 1 1 1
0 0 0 0
1 1 1 1
> █
```