Floyds Algorithm

```
#include <stdio.h>
int n;

void floydWarshall(int matrix[n][n]){
    int mat[n][n];
    for(int i = 0; i < n; i++){
        for(int j = 0; j < n; j++){
            mat[i][j] = matrix[i][j];
        }
    }
    for(int k = 0; k < n; k++){
        for(int i = 0; i < n; i++){
            for(int j = 0; j < n; j++){
                if(mat[i][j]>mat[i][k]+mat[k][j]){
                    mat[i][j]=mat[i][k]+mat[k][j];
                }
            }
        }
    }
    for(int i = 0; i < n; i++){
        for(int j = 0; j < n; j++){
            printf("%d\t",mat[i][j]);
        }
        printf("\n");
    }
}

int main(){
    printf("\nEnter the number of vertices: ");
    scanf("%d",&n);
    int m[n][n];
    printf("\nEnter matrix:\n");
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            scanf("%d",&m[i][j]);
```

```
        }
    }
    printf("Shortest path matrix:\n");
    floydWarshall(m);
    return 0;
}
```