**Experiment 14**

**Write a program to implement the Dynamic Floyd Warshwall Algorithm to solve all pair shortest path problem.**

**Program:-**

#include <stdio.h>

#include <conio.h>

#include <time.h>

int min(int a, int b);

void floyds(int p[10][10], int n) {

int i, j, k;

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

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

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

if (i == j) {

p[i][j] = 0;

} else {

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

}

}

}

}

}

int min(int a, int b) {

if (a < b) {

return (a);

} else {

return (b);

}

}

void main() {

double time;

clock\_t start, end;

int p[100][100],w, n, e, u, v,i,j;

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

scanf("%d", &n);

printf("\n Enter the number of edges: \n");

scanf("%d", &e);

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

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

p[i][j] = 999;

}

}

for (i = 1; i <= e; i++) {

printf("\n Enter the end vertices of edge %d with its weight \n", i);

scanf("%d%d%d", &u, &v, &w);

p[u][v] = w;

}

printf("\n Note: 999 = infinity\n");

printf("\n Matrix of input data:\n");

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

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

printf("%d \t", p[i][j]);

}

printf("\n");

}

start = clock();

floyds(p, n);

printf("\n Transitive closure:\n");

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

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

printf("%d \t", p[i][j]);

}

printf("\n");

}

printf("\n The shortest paths are:\n");

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

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

if (i != j) {

printf("\n <%d,%d> = %d", i, j, p[i][j]);

}

}

}

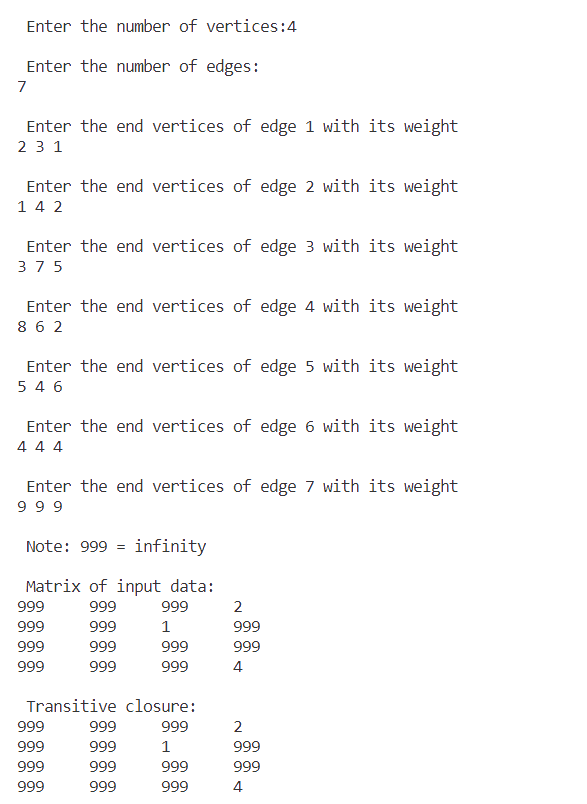
end = clock();

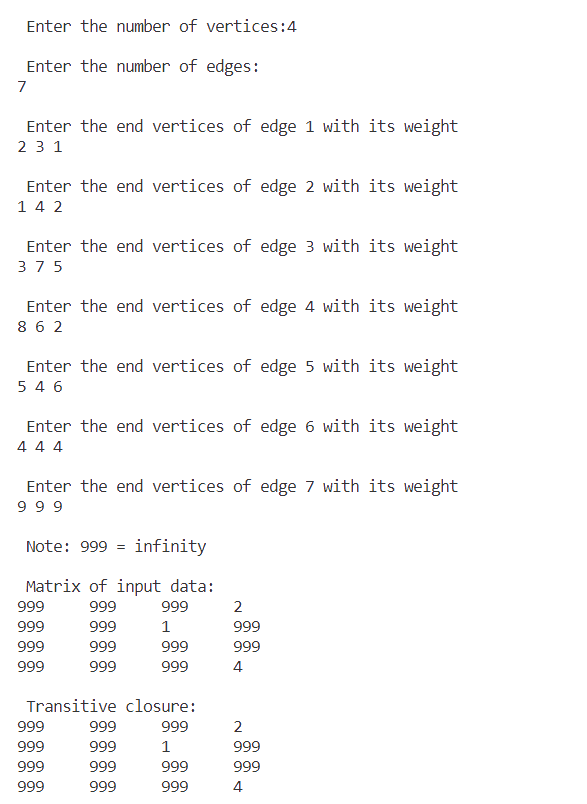
time = ((double) (end - start) \* 1000) / CLOCKS\_PER\_SEC;

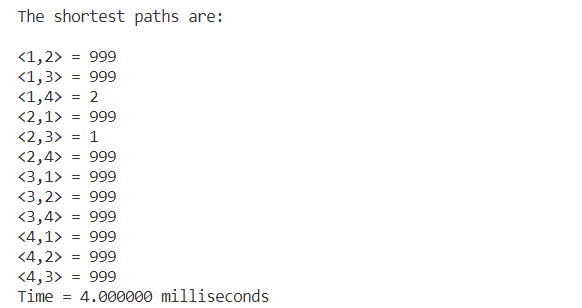
printf("\n Time = %lf milliseconds", time);

}

**Result Analysis and Discussion:**







This experiment has been conducted in a 64-bit system with 16 GB RAM and Processor 12th Gen Intel(R) Core (TM) i5-12500H 3.10 GHz. The algorithm is implemented in C programming language in Visual Studio Code 1.83.1 Code Editor. The time taken by this algorithm for 4 vertices and 7 edges is 4 milliseconds.

**Conclusion:**

The running time of Dynamic Floyd Warshwall Algorithm to solve all pair shortest path problem is analyzed as O(n3).