**Experiment 15**

**Write a program to implement the dynamic algorithm to solve the Travelling Salesman Problem.**

**Program:-**

#include <stdio.h>

#include <conio.h>

#include <time.h>

#include <stdlib.h>

void mincost(int city);

int least(int c);

int ary[10][10], completed[10],n,cost=0;

void takeInput()

{

int i,j;

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

scanf("%d",&n);

printf("\nEnter the Cost Matrix\n");

for(i=0;i<n;i++)

{

printf("\nEnter Elements of Row: %d\n",i+1);

for(j=0;j<n;j++)

{scanf("%d",&ary[i][j]);}

completed[i]=0;

}

printf("\n\nThe cost list is:\n\n");

for( i=0;i < n;i++)

{

printf("\n\n");

for(j=0;j < n;j++)

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

}

}

void mincost(int city)

{

int i,ncity;

completed[city]=1;

printf("%d--->", city+1);

ncity=least(city);

if(ncity==999)

{

ncity=0;

printf("%d",ncity+1);

cost+=ary[city][ncity];

return;

}

mincost(ncity);

}

int least(int c)

{

int i,nc=999;

int min=999, kmin;

for(i=0;i<n;i++)

{

if((ary[c][i]!=0)&&(completed[i]==0))

if(ary[c][i]+ary[i][c] < min)

{

min=ary[i][0]+ary[c][i];

kmin=ary[c][i];

nc=i;

}

}

if(min!=999)

cost+=kmin;

return nc;

}

int main()

{

double t;

clock\_t start, end;

start=clock();

takeInput();

printf("\n\nThe Path is: \n");

mincost(0); //passing e because starting vertex

printf("\n\nMinimum cost is %d\n",cost);

end=clock();

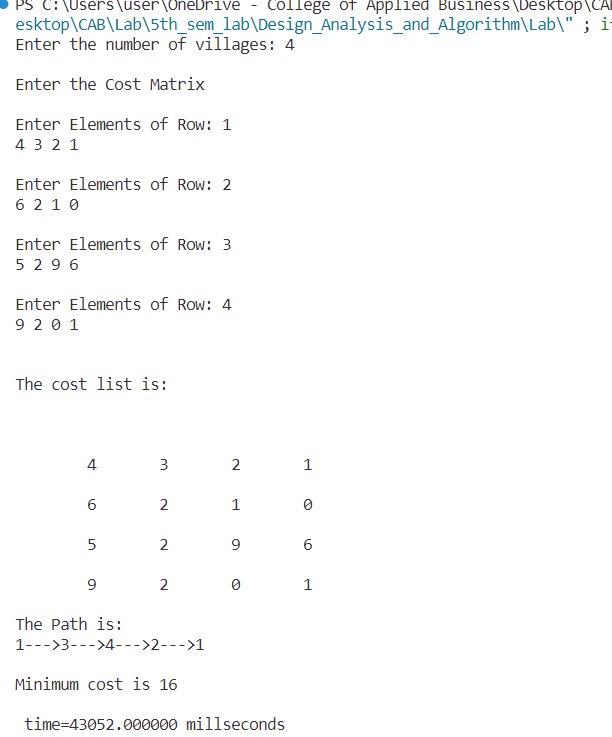
t=((double) (end-start)\*1000)/CLOCKS\_PER\_SEC;

printf("\n time=%lf millseconds", t);

return 0;

}

**Output:**



**Conclusion:**

Travelling Salesman Problem using dynamic algorithm was implemented in C programming language.