```c
#include<stdio.h>
#include<stdlib.h>
#define CITY 5
#define INF 9999


int cost[CITY][CITY];
/* = {
   {0, 20, 42, 25, 30},
   {20, 0, 30, 34, 15},
   {42, 30, 0, 10, 10},
   {25, 34, 10, 0, 25},
   {30, 15, 10, 25, 0}
};*/
int min(int a,int b)
{
    if(a>=b)
    return b;
    else
    {
       return a;
    }

}

int VISIT_ALL = (1 << CITY) - 1;

int dp[16][4];    //make array of size (2^n, n)

int travellingSalesman(int mask, int pos) {
   if(mask == VISIT_ALL)    //when all cities are marked as visited
      return cost[pos][0];    //from current city to origin

   if(dp[mask][pos] != -1)    //when it is considered
      return dp[mask][pos];

   int finalCost = INF;

   for(int i = 0; i<CITY; i++) {
      if((mask & (1 << i)) == 0) {    //if the ith bit of the result is 0, then it is unvisited
         int tempCost = cost[pos][i] + travellingSalesman(mask | (1 << i), i);    //as ith city is visited
         finalCost = min(finalCost, tempCost);
      }
   }
   return dp[mask][pos] = finalCost;
}

int main() {
     printf("enter the cost matrix\n");


   int row = (1 << CITY), col = CITY;
   for(int i = 0; i<row; i++)
      for(int j = 0; j<col; j++)

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

        }
for(int i = 0; i<row; i++)
      for(int j = 0; j<col; j++)

        {         dp[i][j] = -1;    //initialize dp array to -1



        }

    printf( "Distance of Travelling Salesman: ");
    printf("%d",travellingSalesman(1, 0));    //initially mask is 0001, as 0th city already visited
}
```