```
#include <bits/stdc++.h>
using namespace std;
#define F(i,a,b) for(int i = (int)(a); i <= (int)(b); i++)
#define RF(i,a,b) for(int i = (int)(a); i >= (int)(b); i--)
int main()
{
   int X,Y; //X:number of rows, Y: number of columns
   X = Y = 10; //assuming 10X10 matrix
   int Cost[X][Y];

   F(i,0,X-1){
      F(j,0,Y-1){
         //Take input the cost of visiting cell (i,j)
         cin>>Cost[i][j];
      }
   }

   int MinCost[X][Y]; //declare the minCost matrix

   MinCost[0][0] = Cost[0][0];

   // initialize first row of MinCost matrix
   F(j,1,Y-1)
      MinCost[0][j] = MinCost[0][j-1] + Cost[0][j];

   //Initialize first column of MinCost Matrix
   F(i,1,X-1)
      MinCost[i][0] = MinCost[i-1][0] + Cost[i][0];

   //This bottom-up approach ensures that all the sub-problems needed
   // have already been calculated.
   F(i,1,X-1){
      F(j,1,Y-1){
         //Calculate cost of visiting (i,j) using the
         //recurrence relation discussed above
         MinCost[i][j] = min(MinCost[i-1][j],MinCost[i][j-1]) + Cost[i][j];
      }
   }

   cout<<"Minimum cost from (0,0) to (X,Y) is "<<MinCost[X-1][Y-1];
   return 0;
}
```