**Minimum Path Sum**

#include <bits/stdc++.h>

int func(int i, int j, vector<vector<int>> &grid,vector<vector<int>> &dp){

//Base Cases

if(i == 0 and j == 0)return grid[0][0];

//out of bound

if(i < 0 or j < 0) return 1e9;

if(dp[i][j]!=-1) return dp[i][j];

int up = abs(grid[i][j] + func(i-1,j,grid,dp));

int left = abs(grid[i][j] + func(i, j-1, grid,dp));

return dp[i][j] = min(up,left);

}

int minSumPath(vector<vector<int>> &grid) {

// Write your code here.

int n= grid.size();

int m = grid[0].size();

vector<vector<int>> dp(n+1,vector<int>(m+1,-1));

return func(n-1,m-1,grid,dp);

}