Ex2. Ma3moul



**Overview**

It is Eid Alfitr and you can't help but think of your two major fears:

* Feeling sick because of eating lots of ma3moul.
* Going bankrupt because of giving away too much (عيدية) money.

Therefore, you made a map for the houses on the road from your house to your grandparents house, which shows all the houses on the way that you have to enter if you pass by. On the map, you indicated how many ma3moul pieces you expect to be forced to eat in each house by your generous relatives and also the number of kids in each house that you will have to give (عيدية) money to.

Your goal is to get to your grandparents house minimizing the number ma3moul pieces that you eat, and also without paying more than the money you dedicated for this Eid.

**Task Description**

You are given the following:

* The **map** represented as two 2D grids of integers, each of size �×�*N*×*M*, where each cell corresponds to a house.
  + ma3moul[i][j] = the number of ma3moul pieces you are expected to eat at house [i][j].
  + money[i][j] = the amount of money you expect to pay at house [i][j].
* An integer �>0*K*>0 representing the maximum amount of money you are ready to pay during eid.

Assuming that your house is at [0][0] and your grandparents house is at [N-1][M-1], your task is to find the minimum amount of ma3moul pieces you must eat going from your house to your grandparents house.

**Constraints**

* You can move only in the following directions: **DOWN**, **RIGHT**, **DOWN\_RIGHT.**
* If the grids have a negative number at position [i][j], then the house at that position is not a house of one of your relatives and you can't pass through it. I.e., negative numbers represent **blocked cells** on the map.

**Requirements**

Modify **ma3moul.cpp** to implement the following function:

* **int** min\_ma3moul(**int** \*\*ma3moul, **int** \*\*money, **int** N, **int** M, **int** K)

Return the constant **INT\_MAX** in each of the following cases:

* If there is no path from [0][0] to [N-1][M-1] .
* If there is no path from [0][0] to [N-1][M-1] whose sum of money values is less than K.

**Examples**

**Example 1**

**Input:**

ma3moul[][] = {{**1\***, **2\***, 1},

{1, 1, **1\***},

{1, 3, **1\***}}

money[][] = {{**2\***, **2\***, 2},

{1, 10, **2\***},

{1, 1, **4\***}}

K = 10

**Result**: 5 (path shown with astrisks on the grid)

- The diagonal path is invalid: Eidieh cost is 14 > K=10.

- All other paths require eating more ma3moul pieces than 5.

**Example 2**

**Input:**

ma3moul[][] = {{1, 1, 1},

{1, -1, -1},

{1, -1, 1}}

money[][] = {{2, 2, 2},

{2, -2, -2},

{2, -1, 2}}

K = 10

**Result**: INT\_MAX (no path from [0][0] to [N-1][M-1])

**Example 3**

**Input**:

ma3moul[][] = {{1, 1, 1},

{1, 1, 1},

{1, 1, 1}}

money[][] = {{2, 2, 2},

{2, 2, 2},

{2, 2, 2}}

K = 5

**Result**: INT\_MAX (All paths cost > K)