



Martian Mining

LA 3530, uva 1366

Time: 3 seconds

Problem Descriptions(1/6)

- ❖ **The NASA Space Center, Houston, is less than 200 miles from San Antonio, Texas (the site of the ACM Finals this year).**
- ❖ **This is the place where the astronauts are trained for Mission Seven Dwarfs, the next giant leap in space exploration. The Mars Odyssey program revealed that the surface of Mars is very rich in yeyenum and bloggium.**
- ❖ **These minerals are important ingredients for certain revolutionary new medicines, but they are extremely rare on Earth.**

Problem Descriptions(2/6)

- ❖ The aim of Mission Seven Dwarfs is to mine these minerals on Mars and bring them back to Earth.
- ❖ The Mars Odyssey orbiter identified a rectangular area on the surface of Mars that is rich in minerals.
- ❖ The area is divided into cells that form a matrix of n rows and m columns, where the rows go from east to west and the columns go from north to south. The orbiter determined the amount of yeyenum and bloggium in each cell.

Problem Descriptions(3/6)

- ◆ The astronauts will build a yeyenum renement factory west of the rectangular area and a bloggium factory to the north. Your task is to design the conveyor belt system that will allow them to mine the largest amount of minerals.
- ◆ There are two types of conveyor belts: the rst moves minerals from east to west, the second moves minerals from south to north.
- ◆ In each cell you can build either type of conveyor belt, but you cannot build both of them in the same cell.

Problem Descriptions(4/6)

- ◆ If two conveyor belts of the same type are next to each other, then they can be connected. For example, the bloggium mined at a cell can be transported to the bloggium renement factory via a series of south-north conveyor belts.

Problem Descriptions(5/6)

- ❖ **The minerals are very unstable, thus they have to be brought to the factories on a straight path without any turns.**
- ❖ **This means that if there is a south-north conveyor belt in a cell, but the cell north of it contains an east-west conveyor belt, then any mineral transported on the south-north conveyor belt will be lost.**

Problem Descriptions(6/6)

- ◆ The minerals mined in a particular cell have to be put on a conveyor belt immediately, in the same cell (thus they cannot start the transportation in an adjacent cell).
- ◆ Furthermore, any bloggium transported to the yeyenum renement factory will be lost, and vice versa.
- ◆ Your program has to design a conveyor belt system that maximizes the total amount of minerals mined,i.e., the sum of the amount of yeyenum transported to the yeyenum renery and the amount of bloggium transported to the bloggium renery.

Input

- ❖ The input contains several blocks of test cases. Each case begins with a line containing two integers: the number $1 \leq n \leq 500$ of rows, and the number $1 \leq m \leq 500$ of columns.
- ❖ The next n lines describe the amount of yeyenum that can be found in the cells. Each of these n lines contains m integers. The first line corresponds to the northernmost row; the first integer of each line corresponds to the westernmost cell of the row. The integers are between 0 and 1000.
- ❖ The next n lines describe in a similar fashion the amount of bloggium found in the cells.
- ❖ The input is terminated by a block with $n = m = 0$.

Output

- ❖ For each test case, you have to output a single integer on a separate line: the maximum amount of mineralsthat can be mined.

Sample Input / Output

4 4 ← nxm rectangle

0 0 10 9

1 3 10 0

4 2 1 3

1 1 20 0

10 0 0 0

1 1 1 30

0 0 5 5

5 10 10 10

0 0

End of input

yeyenum in each cell

bloggium in each cell

98

	10		45
	0,10	0,0	10,0
14	1,1	3,1	10,1
7	4,0	2,0	1,5
22	1,5	1,10	20,1
		0	0,10

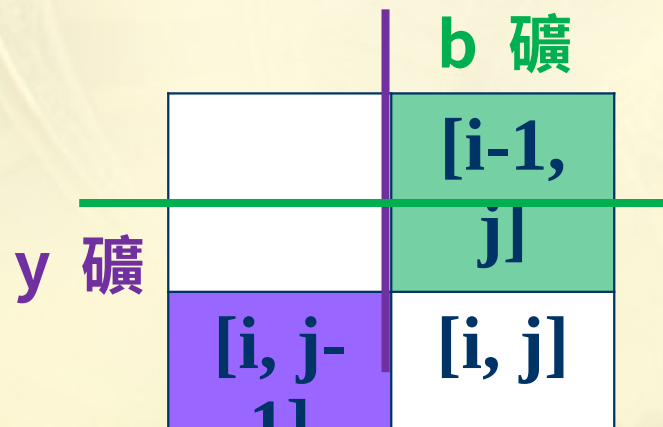
$$14+7+22+10+45=98$$

State Definition

State Transition:

讓 b 礦

讓 y 礦



y

0	0	10	19
1	4	14	14
4	6	7	10
1	2	22	22

b

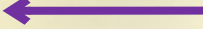
10	0	0	0
11	1	1	30
11	1	6	35
16	11	16	45

b

	1	2	3	4
1	10	0	0	0
2	11	1	1	30
3	11	1	6	35
4	16	11	16	45



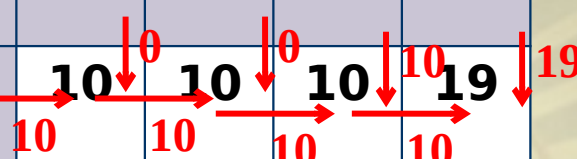
y



	1	2	3	4
1	0	0	10	19
2	1	4	14	14
3	4	6	7	10
4	1	2	22	22


dp[i][j]

	0	1	2	3	4
0	0	0	0	0	0
1	0	10	10	10	19
2	0				
3	0				
4	0				




b

	1	2	3	4
1	10	0	0	0
2	11	1	1	30
3	11	1	6	35
4	16	11	16	45



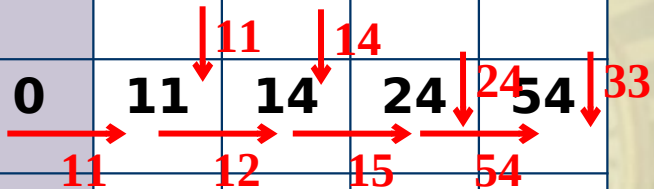
y



	1	2	3	4
1	0	0	10	19
2	1	4	14	14
3	4	6	7	10
4	1	2	22	22

dp[i][j]

	0	1	2	3	4
0	0	0	0	0	0
1	0	10	10	10	19
2	0	11	14	24	54
3	0				
4	0				



b

	1	2	3	4
1	10	0	0	0
2	11	1	1	30
3	11	1	6	35
4	16	11	16	45

↑

y

←

	1	2	3	4
1	0	0	10	19
2	1	4	14	14
3	4	6	7	10
4	1	2	22	22

dp[i][j]

	0	1	2	3	4
0	0	0	0	0	0
1	0	10	10	10	19
2	0	11	14	24	54
3	0	15	20	31	66
4	0	16	27	53	98

```
1  #include <iostream>
2  #include <cstring>
3  #include <cstdio>
4  #include <algorithm>
5
6  using namespace std;
7
8  const int maxn = 500 + 10;
9  int n, m, dp[maxn][maxn], y[maxn][maxn], b[maxn][maxn];
10
11 int main()
12 {
13     freopen("d:\\l366_in.txt", "r", stdin);
14     int i, j, x;
15     while (~scanf("%d%d", &n, &m))
16     {
17         if (n == 0 && m == 0) break;
18
19         memset(dp, 0, sizeof(dp));
20         memset(y, 0, sizeof(y));
21         memset(b, 0, sizeof(b));
22
23         for (i = 1; i <= n; i++)
24         {
25             for (j = 1; j <= m; j++)
26             {
27                 scanf("%d", &x);
28                 y[i][j] = y[i][j - 1] + x;
29             }
30         }
31     }
```



```
32     for(i=1; i<=n; i++)
33     {
34         for(j=1; j<=m; j++)
35         {
36             scanf("%d",&x);
37             b[i][j]=b[i-1][j]+x;
38         }
39     }
40
41     for(i=1; i<=n; i++)
42     {
43         for(j=1; j<=m; j++)
44             dp[i][j]=max(dp[i-1][j]+y[i][j], dp[i][j-1]+b[i][j]);
45     }
46     printf("%d\n", dp[n][m]);
47 }
48 return 0;
49 }
```