

Table

Input file: **standard input**
Output file: **standard output**
Time limit: 1.5 seconds
Memory limit: 128 megabytes

As you may already know, accountants keep their data in form of tables and they calculate all sorts of sums on lines and columns. Atnoc, our accountant has organized his values in the form of a table with N rows (numbered from 0 to $N - 1$) and M columns (numbered from 0 to $M - 1$).

Elements on the last column are the sums of their row (more precisely, the element on row i and column $M - 1$ is the sum of elements on row i situated on the columns $0, 1, 2, \dots, M - 2$), and the elements on the last line are the sums of their column (more precisely, the element on column i and row $N - 1$ is the sum of elements on column i situated on the rows $0, 1, 2, \dots, N - 2$). An example of such a table is shown below.

2	5	7	14
11	6	6	23
13	11	13	37

Unfortunately Atnoc spilled water on his beloved table and in so doing some of the table's elements became unreadable. In order to recover the value on cell (i, j) he will need to pay a cost of B_{ij} .

Determine the minimum cost Atnoc has to pay in order to be able to recover the table.

To solve this problem you will have to implement one or more functions. For more details, make sure to check out `sample.cpp` located in Statement Attachments.

Input

You won't read any number from the input. You will receive the input as parameters in the function whose header will be found in `sample.cpp`.

$(1 \leq n, m \leq 3000)$, $(1 \leq a_{ij} \leq 100 \text{ or } a_{ij} = -1)$, $(1 \leq b_{ij} \leq 5 * 10^8)$.

For tests worth 10 points, $(1 \leq n, m \leq 4)$.

For tests worth 25 more points, $(1 \leq n, m \leq 15)$.

For tests worth 40 more points, $(1 \leq n, m \leq 750)$.

Output

You will have to return the answer asked in the statement.

Note

Sample input:

```
3 3
-1 -1 -1
-1 2 -1
3 4 7
3 3 18
4 1 4
2 2 2
```

By finding the cell $(1, 1)$, paying 3 we will have enough information to reconstruct the entire table.