

THE 13TH LATVIAN OLYMPIAD IN INFORMATICS
3RD STAGE PROBLEMS
Day 2 (march 28, 2000)



1. "Swimming pool"

A long time ago the construction of swimming pools was done thus: first, a rectangular area of length m meters and width n meters was designated. This area was then divided to $1*1$ meter sized squares, thus obtaining $m*n$ cells. Then either a perfectly cut granite cube with side length equal to 1, or a tower of these perfect cubes was placed in each cell.

The cubes were so perfectly made that it was sufficient for two cubes to have their edges next to each other so that this place would be hermetically sealed (water would not flow through).

When water was flowing, such a "set of cubes" could only hold a limited amount of water.

Your task is to write a program, which would determine the maximum possible volume of water in cubic meters that can be gathered in this pool.

Input data

The first line of the text file `BASEINS.DAT` contains two positive integers m and n , which are the lengths of sides of the rectangular area in meters ($0 < m, n \leq 100$).

The next of the m lines in the file contain each n natural numbers.

The j -th number in the $i+1$ -st line of the file shows the height of the cube tower located in the i -th row of the j -th column (or number of cubes positioned there). It is given that there is at least one cube in each cell and that the height of the tower does not exceed 10000 in any of the cells.

Output data

The only line of the text file `BASEINS.REZ` must contain one positive integer – the maximum possible volume of water in cubic meters that can be stored in the given swimming pool.

Example

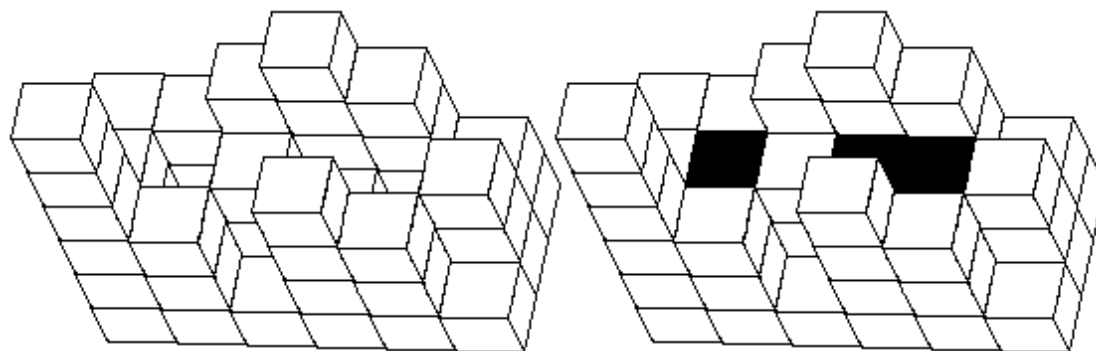
Input data (file `BASEINS.DAT`)

```
3 6
3 3 4 5 4 2
3 1 3 2 1 3
6 3 1 4 3 1
```

Output data (file `BASEINS.REZ`)

5

A view from the top of an empty pool and of a filled pool given in the pictures below.



2. "House construction"

A construction company on the Misfortune island has received an order to build L identical houses. There are X wooden planks and a unit area required to build each house on the island. For the completion of the order an area of precisely L unit areas has been allocated for the construction works.

The company already has 10 wooden plank factories located in another region on the island. For faster completion of the order, there may be a necessity for additional factories. There are Y wooden planks and a unit area required to build a factory. As the government of the island is not willing to provide additional land for factory construction, the only possibility is to use the land given for completion of the order for the construction of temporary factories and later raze these factories as necessary.

Each factory produces 10 planks per Day. The planks created by all the factories are identical.

The planks, which are not used in any specific Day for construction, are destroyed – eaten by little green plank eaters during the night. Due to this reason, also the planks from the razed factories cannot be recycled.

During one Day the workers of the company can carry out only one of the following actions:

- Build any number of factories, as long as the working factories produce sufficient number of planks that Day and the non-utilized area is sufficient.
- Build any number of houses, as long as the working factories produce sufficient number of planks that Day and the non-utilized area is sufficient.
- Raze any number of factories.

The newly-built factories can start producing planks (10 per Day), beginning with the following Day after being constructed.

Your task is to write a program that determines the minimum possible number of Day required for fulfillment of the order.

Input data

The first line of the text file BUVE.DAT contains a positive integer L ($L \leq 2500$), which is the ordered number of houses. The second line contains a positive integer X ($X \leq 255$) – the number of planks required to build one house. The third line contains a positive integer Y ($Y \leq 255$) – the number of planks required to build one factory.

Output data

The only line of the text file BUVE.REZ must contain a positive integer – the minimum number of Days, during which the order could be fulfilled. If it is not possible to fulfill the order with the given input data, the number 0 must be output to the only line of the file.

Example

Input data (file BUVE.DAT)

50
30
10

Output data (file BUVE.REZ)

12

One of the possible ways to fulfill the order within 12 Days is the following:

At the beginning [houses:0, factories:10, non-utilized area:50]

Day 1: Construct 10 factories	[0, 20, 40]	Day 7: Construct 5 houses	[35, 15, 10]
Day 2: Construct 10 factories	[0, 30, 30]	Day 8: Construct 5 houses	[40, 15, 5]
Day 3: Construct 10 houses	[10, 30, 20]	Day 9: Construct 5 houses	[45, 15, 0]
Day 4: Construct 10 houses	[20, 30, 10]	Day 10: Raze 5 factories	[45, 10, 5]
Day 5: Construct 10 houses	[30, 30, 0]	Day 11: Construct 3 houses	[48, 10, 2]
Day 6: Raze 15 factories	[30, 15, 15]	Day 12: Construct 2 houses	[50, 10, 0]