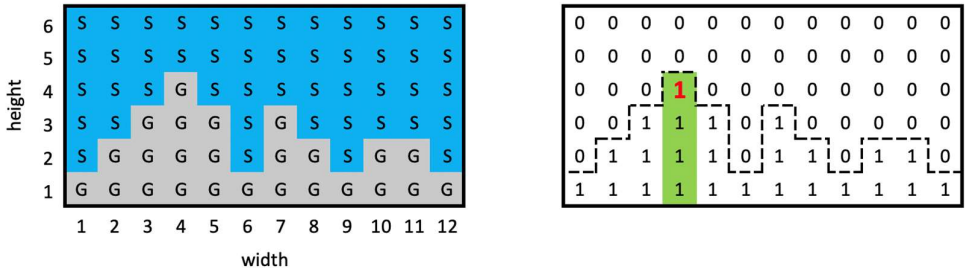
Orbital survey

Problem description

An automated probe is sent to conduct orbital surveys of nearby planets. The scientists are particularly interested in the shape and the configuration of the planets' surfaces. The probe flies over a surface and scans the local region. Each scan is a 2D image, which, in the raw format, is temporarily saved as a binary sequence of 1's and 0's. 1's represent solid ground (mountains), while 0's represent open sky.

For example, a 12 x 6 area given in the image below (left), will be stored as the binary sequence in the right. Note, dashed lines and colors are there to help you visualize the problem, and are not the part of the encoded data.

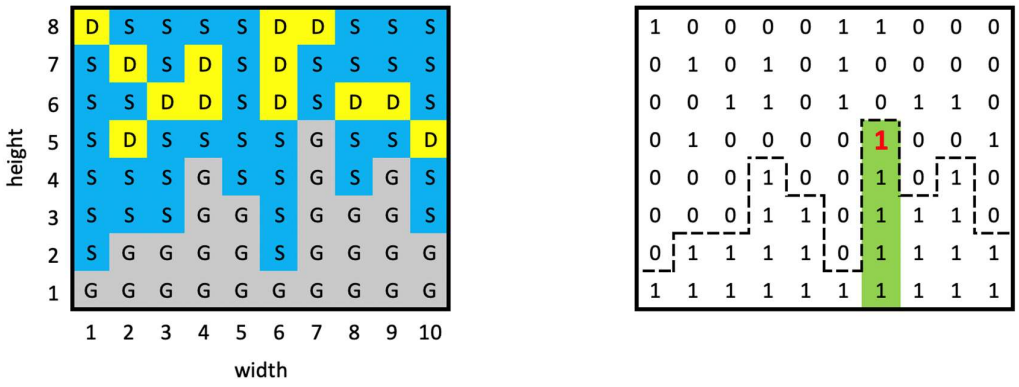


Your task is to write a program that will analyze the binary sequence and use it to determine the maximum height of the scanned area. In the example above, the answer is 4 (see the column in green).

However, there could be metallic dust above the ground, interfering with the probe's sensors. If dust is present, it will generate a false-positive and the binary value of that particular cell will be erroneously saved as 1, too. The scientists have looked at the problem and determined, that the interference **never occurs exactly on top of any peaks**, i.e. there will be

always at least one 0 between the peak and the metallic dust. Therefore, the algorithm should be able to filter out the noise.

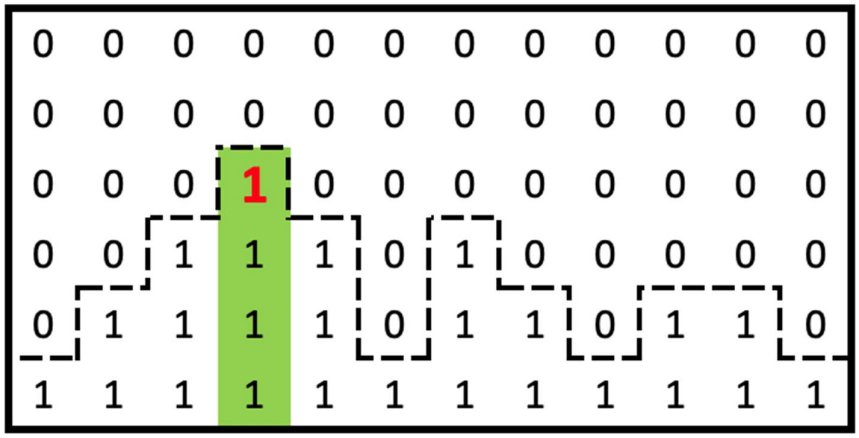
For example, below is the 10 x 8 image of an area with the noise generated by metallic dust (cells marked as D). Please note that both G cells and D cells register as 1s. As stated in the paragraph above, there is assurance that a D cell can never be on top of a G cell. For this particular case, the highest peak measurement, after filtering the noise, should be 5.



Data format

The probe scans the area from left to right, row by row, starting at the top-left cell, and ending with the bottom right cell. Then, 2D image is encoded in 1D array (list) in the same order as the cells have been scanned.

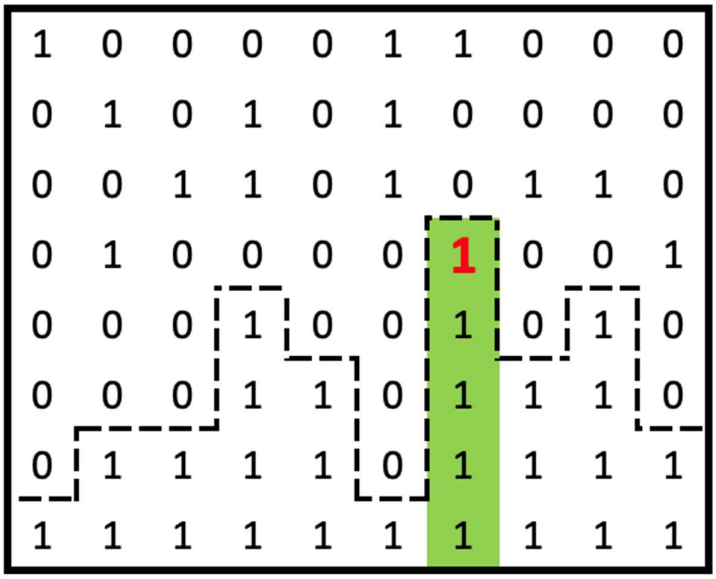
For example, the first 12 x 6 image:



as 1D binary list will look like:

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

The second 10 x 8 image:



as 1D binary list will look like:

[1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

The probe then sends the data back to Earth, consisting of **three integer values**. The **first** value is the **number of rows** in the image (height). The **second** value is the **number of columns** in the image (width). Finally, the **binary sequence is converted to its decimal equivalent**, and **that number is sent as the third value**.

Important reminder: when converting a binary sequence to a decimal value, leading zeroes will be discarded. For example: $0010_2 \rightarrow 2_{10} \rightarrow 10_2$, $00001001_2 \rightarrow 9_{10} \rightarrow 1001_2$.

Problem statement

Read three numeral inputs from the standard input – the user will input one number at a time. **Do not print any text to the user**. For the given values, find the maximal height of the terrain, according to the process and the format described above. Then, print that number as an integer value. Again, **please do not add any additional text**.

Note: sequence
$00000000000000000000000000010000000000111010000001111011011011111111111_2$
(the first example) is 17607763390463_{10}.
Also,
$10000110000101010000001101011001000010010001001010000110111001111011111111111111_2$ (the second example) is $633184731564110577647615_{10}$.

Example input and output:
Input is given in BLUE. Required output is given in RED.

Case 1:

6
12
17607763390463
4

Case 2:
8
10
633184731564110577647615
5

Case 3:
8
8
2887099909389580922
6

Case 4:
2
2
7
2

Case 5:
2
2
12
0