# Problem I. Addition

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 5 seconds |
| Memory limit: | 256 mebibytes |

This is an interactive problem.

Polycarp is a software engineer in the «Abacus» company. Each day he must generate $N$ binary strings of length $M$.

Once Polycarp get bored and decided to play the following game: when he generates next binary string, he treats it as a number written in binary representation (it is possible that number contains leading zeroes). Then he adds that number to each of previously generated strings (also treated as numbers) and counts the number of times when the sum can't be written with M or less binary digits.

But Polycarp is not so good in the binary addition, so he asks you to help him.

## Interaction Protocol

At the start of interaction process, the jury program sends to your program two positive integers $N$ and $M$ ($1 \le N \times M \le 10^5$) — number of binary strings, generated by Polycarp, and requested length, appropriately.

Then next scenario repeats $N$ times: the jury program sends at the new line to your program next binary string of length $M$, and your program must print at the new line one integer — number of times when this string, treated as integer in binary representation, added to one of previously generated strings is too big to fit $M$ binary digits.

When all $N$ strings are processed, your program must exit with exit code 0, else interaction will cause an error (even if all answers will be correct).

## Examples

| standard input | standard output |
|---|---|
| 5 2<br>01<br>10<br>11<br>00<br>10 | 0<br>0<br>2<br>0<br>2 |
| 5 3<br>110<br>001<br>101<br>110<br>011 | 0<br>0<br>1<br>2<br>3 |

## Note

Please do not forget to flush output buffer after each printed answer, i.e. do next:

- For Pascal: **flush(output)**;
- For C/C++: **fflush(stdout)** or **cout.flush()**;
- For Java: **System.out.flush()**;
- For Python: **sys.stdout.flush()** from library **sys**;

- For C#: **Console.Out.Flush()**;

Also dont forget to print the newline character after each number you output.