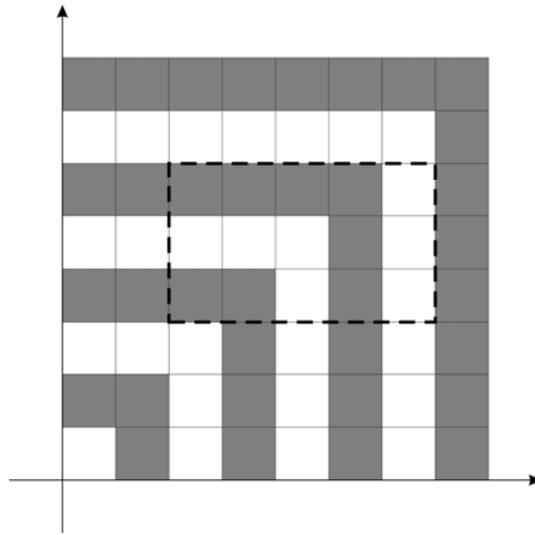

tasks

| task | pruge | petice | maraton |
|---|-----------------------------------|--------------------------------------|---|
| source file | pruge.pas pruge.c pruge.cpp | petice.pas petice.c petice.cpp | maraton.pas maraton.c maraton.cpp |
| input data | stdin | | |
| output data | stdout | | |
| time limit (Intel Celeron 2.66Ghz) | 1 sec | | |
| memory limit (heap) | 32 MB | | |
| memory limit (stack) | 8 MB | | |
| points | 40 | 50 | 60 |
| | 150 | | |

pruge

A pattern composed of dark stripes (as depicted in the figure below) spans the entire first quadrant in the rectangular coordinate system.



Write a program that finds the total number of dark squares inside the given rectangle.

input data

The first and only line of input contains four integers X_1 , Y_1 , X_2 i Y_2 – the coordinates of the rectangle with the lower left corner in (X_1, Y_1) and upper right corner in (X_2, Y_2) .

$0 \leq X_1 < X_2 \leq 1\,000\,000$, $0 \leq Y_1 < Y_2 \leq 1\,000\,000$.

output data

The first and only line of output should contain a single integer – the total number of dark squares inside the given rectangle.

Note: use the 64-bit signed integer type (int64 in Pascal, long long in C/C++).

examples

input

2 3 7 6

output

8

input

0 4 2 104

output

100

input

4 1 7 50

output

72

petice

Write a program that, given integers N and K , finds the **smallest integer greater than N** whose decimal representation **contains at least K occurrences of digit 5**.

input data

The first and only line of input contains two integers N and K , $1 \leq N \leq 10^{15}$, $1 \leq K \leq 15$.

output data

The first and only line of output should contain a single integer – the smallest number from the task description.

Note: use the 64-bit signed integer type (int64 in Pascal, long long in C/C++).

examples

input

99 1

output

105

input

595 2

output

655

input

123456789012345 3

output

123456789012355

maraton

N teams are participating in a football marathon. A total of K matches are to be played, one match at a time.

The first match is played between teams 1 and 2. After that match, the defeated team leaves the field while the winner stays and plays team 3. After that match, the winner again stays on the field and plays the team 4 etc.

After team N finishes playing its match, we start over with team 1, then team 2, and so on. If a team is supposed to enter the field, but it is already there (as the winner), or if it has just left the field (was defeated), the next team enters the field.

Because of the big differences between teams in the tournament, the results are very predictive, and for each two teams **we know who will be the winner** in a match between them.

Write a program that finds the **total number of matches** that each of the teams will play.

input data

The first line of input contains two integers N and K, $3 \leq N \leq 1000$, $0 \leq K \leq 10^{14}$.

Each of the following N lines contains sequence of N integers representing the NxN matrix consisting **only of zeros and ones**.

If A_{ij} is 1 in the matrix, then team **i** defeats team **j** every time they play and if $A_{ij}=0$ then team **j** defeats team **i** ($A_{ij} \neq A_{ji}$).

output data

The first and only line of output should contain N numbers – the i^{th} number is the total number of matches that team **i** will play in the tournament.

Note: use the 64-bit signed integer type (int64 in Pascal, long long in C/C++).

examples

| input | input | input |
|--------|-------------------|---------|
| 3 5 | 3 50000 | 4 5 |
| 000 | 000 | 0110 |
| 100 | 100 | 0010 |
| 110 | 110 | 0000 |
| | | 1110 |
| output | output | output |
| 3 3 4 | 25000 25001 49999 | 3 2 2 3 |