Numerological prediction (bloomsday)

Everybody knows that the world should end with the coming of *doomsday*, when all ills will be unleashed, throwing the earth in an hell of ice and fire.

What is not nearly as known is that (given the right astrological conditions) a day of heavenly peace and joy should also come!

Predicting this day, called *bloomsday*, is the ultimate quest of the most renowned numerologists, and Giorgio is no exception. After years of excruciating scientific research, he proved that bloomsday will come when the *pattern of god* will finally appear in the *holy sequence*.

For this quest, Giorgio printed out the first few bazillion numbers of the "holy sequence", which is the infinite sequence of numbers:

$$K \cdot 1^E,\ K \cdot 2^E,\ \ldots,\ K \cdot X^E,\ \ldots$$

where K and E are carefully chosen to maximise holiness. Even though the description of the "pattern of god" is too complex to fit into the margin of this page, Giorgio has already found it in the N-th digit of the printout! Thus, this digit belongs to the X-th element $K \cdot X^E$ where X is the advent of bloomsday, measured in days elapsed from the beginning of the universe.

Figure 1: Artist's impression of the advent of bloomsday.

Sadly, due to his limited research budget, Giorgio had to print the numbers with the lowest quality and smallest font, making impossible to recognise the spaces between subsequent numbers of the sequence. Thus, Giorgio has no clue on which X corresponds to the N he found. Help him find when bloomsday will come, by computing such X!

☞ Among the attachments of this task you may find a template file bloomsday.* with a sample incomplete implementation.

Input

The first and only line contains integers K, E, N.

Output

You need to write a single line with an integer: the number X such that the N-th digit of the sequence belongs to $K \cdot X^E$.

Constraints

- $1 \leq E, K \leq 10$.
- $1 \leq N, X < 10^{18}$.
- $K \cdot X^E < 10^{18}$ except in subtasks where otherwise stated.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

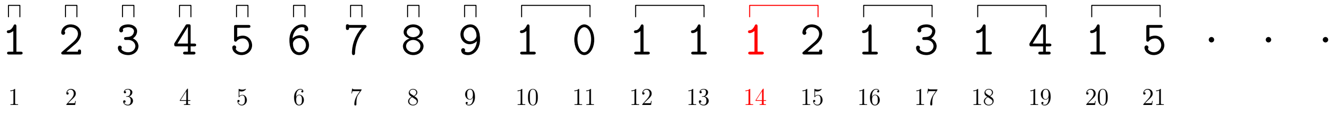
- **Subtask 1** [**0 points**]: Examples.
- **Subtask 2** [**30 points**]: $K = E = 1$, $X < 10\,000$.
- **Subtask 3** [**15 points**]: $K = E = 1$.
- **Subtask 4** [**15 points**]: $E = 1$.
- **Subtask 5** [**10 points**]: $X < 10\,000$.
- **Subtask 6** [**20 points**]: No additional limitations.
- **Subtask 7** [**10 points**]: $K \cdot X^E > 10^{18}$.

Examples

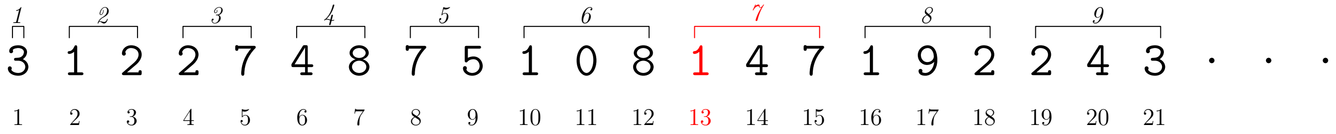
input.txt	output.txt
`1 1 14`	`12`
`3 2 13`	`7`

Explanation

In the **first sample case**, the holy sequence starts as follows (the N-th digit is highlighted in red):



In the **second sample case**, the holy sequence starts as follows (the N-th digit is highlighted in red):


1 2 3 4 5 6 7 8 9
3 1 2 2 7 4 8 7 5 1 0 8 1 4 7 1 9 2 2 4 3 . . .
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21