A look-and-say sequence is a sequence of integers, expressed in decimal notation, where each sucessive term is generated by *describing* the previous one.

For instance, if x_1 (the first term of the sequence) is 1, the next term is the description of this term, 11 (“one 1”), which is described by 21 (”two 1’s”), which is described by 1211 (“one 2 one 1”), etc.; the series continues 111221, 312211, 13112221, ...

Your problem is to build a program that, given the first term of a look-and-say sequence x_1, calculates the j-th digit of the i-th term, x_i.

Input

Each line in the input corresponds to a test case specified by 3 integer values: x_1, i and j, with $1 \leq x_1 \leq 1000$, $1 \leq i \leq 1000$ and $1 \leq j \leq \min(\lfloor \log_{10}(x_i) + 1 \rfloor, 1000)$. The end of the input is indicated by a line ‘`0 0 0`’.

Output

For each test case, the program must output a line with the j-th digit of the term x_i of the look-and-say sequence that starts with the term x_1.

Sample Input

```
1 3 1
1 3 2
1 7 2
123 3 1
0 0 0
```

Sample Output

```
2
1
3
3
```