6766 Money For Nothing

Every Wednesday and Saturday night, thousands of Ohioans watch Karen Harris oversee the selection of that evening's winning Super Lotto numbers. Forty-seven numbered balls are released into a mixing chamber, and six are randomly extracted to establish the winning combination. Cash prizes await those who match four, five, or all six of the winning numbers. The cash prizes increase significantly in magnitude between these three prize levels — that's the *good* news for lottery players.

According to the rules of combinatorics, the number or unique winning combinations of these balls is given by the formula 47!/(6!*(47-6)!). This means that the odds of a single wet of six numbers beting selected is 1 is 10,737,573–and that's the *bad* news for lottery players.

The Ohio Lottery Commission must be capable of processing millions of tickets for each drawing. Its staff must be able to record the numbers on each ticket sold, recall these numbers, and quickly verify them whenever winning tickets (if any) are presented. In order to do this effectively, a *sequence number* must be assigned to each winning combination. This naturally entails the development of an ordering scheme over the set of possible combinations. Note that when viewed from this perspective, the Ohio Lottery is really picking only one sequence number from the 10,737,573 possible combinations which exist — the use of the individually numbered balls, while providing an entertaining and suspense-filled method of selecting the winning combination, also serves to somewhat conceal the true nature of the game.

Suppose that there are to be n balls selected from a pool of m. Consider numbering the combinations according to the following scheme: two different combinations of balls are represented by A_1, A_2, ..., A_n and B_1, B_2, ..., B_n, where $A_i < A_{(i+1)}$ and $B_i < B_{(i+1)}$. To determine which combination appears first in the ordering, each corresponding ball number is examined beginning with A_1 and B_1 until a difference is found. The combination with the lower differing number comes earlier in the ordering. Because this produces a deterministic ordering, sequence numbers may be applied directly.

Consider the following example: suppose that 3 balls are to be chosen from a pool of 8 to determine a winning combination. Two possible combinations are 2-3-8 and 2-4-7. Both combinations begin with a 2, so the second element of each triple must be considered. Since 3 comes before 4, the combination 2-3-8 comes earlier in the ordering. 1-2-3 is the first combination in the ordering, so it has the sequence number of 1. 6-7-8 is the last combination in this case, with a sequence number of 56.

Write a program that will accept a lottery description as outlined above (total number of balls and number of balls drawn to determine a winner) along with a sequence number for that lottery, and generate the set of winning numbers corresponding to the supplied sequence number.

Input

The input file will consist of an unknown number of lottery records. Each lottery record will consist of three integers on a single line separated by at least one space. The first integer will give the number of balls in the pool, the second will indicate the number of balls drawn from the pool, and the third will be the sequence number of a particular combination. You may assume that when there are m balls in the pool that these balls will be numbered from 1 to m and that $3 \leq m \leq 50$ in every case. You may also assume that when n balls are drawn from the pool, that $2 \leq n \leq 15$ in every case. Furthermore, m will always be strictly greater than n. Note that there is no guarantee, however, that the sequence number requested will necessarily correspond to a combination which actually exists in the lottery described.

All data values which appear in the input file will be positive integer values.

Output

Your program will output a sentence for each lottery record in the input file. Each sentence will be of the form:

`In` n `of` m `lotteries,` *combination* `has sequence number` s`.`

where n is the number of balls selected to determine the winning combination, m is the total number of balls in the pool of available numbers, and *combination* is the list of numbers (in the format detailed below) which correspond to sequence number s.

In each output sentence, requested lottery number combinations must be separated by hyphen characters (hyphern characters which lead off or trail behind the requested lottery combination are not acceptable). All of the characters in the output sentence must appear as given, including the comma (with its trailing space) and the period.

Should a requested sequence number not exist for an input lottery description, the output sentence should read:

`In` n `of` m `lotteries, no combination corresponds to sequence number` s`.`

Sample Input

```
8 3 4
8 3 1
8 3 56
8 3 57
8 5 10
5 2 7
47 6 2034
```

Sample Output

```
In 3 of 8 lotteries, 1-2-6 has sequence number 4.
In 3 of 8 lotteries, 1-2-3 has sequence number 1.
In 3 of 8 lotteries, 6-7-8 has sequence number 56.
In 3 of 8 lotteries, no combination corresponds to sequence number 57.
In 5 of 8 lotteries, 1-2-3-7-8 has sequence number 10.
In 2 of 5 lotteries, 2-5 has sequence number 7.
In 6 of 47 lotteries, 1-2-3-6-14-25 has sequence number 2034.
```