Legendary Game

Someone who plays the peak of mobile games gets very angry at the game they have to play while trying to sign a contract one day. The game goes as follows:

- There is a card pool consisting of numbers from 1 to $2N + K$.
- The game draws N cards one by one from this card pool (without replacement, meaning **a number cannot be drawn twice**).
- After the drawn cards, the player makes a guess as to whether the next card will be higher or lower.

The point that angers the player is that, due to their bad luck, a smaller number constantly comes after a small number, and a constantly larger number comes after a large number. Upon this, the player aims to prevent this by modifying the randomness mechanism of the game. The change they make adds the following two rules to the card drawing:

- If a card numbered from 1 to N is drawn, the number on the next drawn card must be strictly greater than N ($> N$).
- If a card numbered from $N + K + 1$ to $2N + K$ is drawn, the number on the next drawn card must be strictly less than $N + K + 1$ ($< N + K + 1$).

The player wonders how much they broke the game after making these modifications. The measure of how much they broke it is the ratio of the number of all possible different card drawing sequences in the modified version of the game to the number of all possible card drawing sequences in the normal version of the game. They ask you for this ratio modulo $10^9 + 7$.

Implementation Details

You are expected to implement the following function:

```
int oyun(int N, int K)
```

- N and K are integers determining the card pool and the number of cards to be drawn as mentioned in the question.
- This function should return the requested ratio.
- This function will be called exactly once in each test.

Constraints

- $1 \leq N \leq 10^6$
- $1 \leq K \leq 100$

Subtasks & Scoring

Subtasks	Points	Additional Constraints	Dependencies
1	10	$1 \leq N, K \leq 6$	—
2	18	$1 \leq N \leq 100$	1
3	6	$1 \leq N \leq 1000$	1, 2
4	47	$1 \leq N \leq 10^4$	1, 2, 3
5	19	No additional constraints	1, 2, 3, 4

Examples

```
oyun(2, 1)
```

The function call above is expected to return `600000005`.

Our numbers: 1, 2, 3, 4, 5. We will draw 2 cards. In a normal situation, we can draw the cards in 20 different ways. In the modified game, if our first card is 1, 2, 4, or 5, there are 3 different possibilities for the second card; if it is 3, there are 4 different possibilities for the second card. So, there are a total of 16 possibilities for the modified game. $\frac{16}{20} = \frac{4}{5}$. This probability is equivalent to 600000005 in our modulo value.

Sample Grader

The sample grader expects each case in the following format:

```
N K
```

If the requested function successfully returns an integer, it will print this result R in the following format:

```
R
```