# Task 3: RADIOACTIVE

In the Cow Village, every cow owns exactly one house. These houses are arranged in a straight line, ending at the inaptly named Village Centre. The $N$ houses are labelled from 1 to $N$ in order of their distance from the Village Centre, starting from the house nearest to the Village Centre.

Now Kraw the Krow has lots and lots of bananas and is feeling very generous today. He decides to give the cows some of his bananas. He has so many bananas, anyway, that he will not run out of bananas no matter how many he gives the cows.

However, as we all know, bananas are radioactive. What's worse, Kraw's bananas are particularly radioactive because he lives close to the Equator where the average cosmic ray flux is the highest. If too many of Kraw's bananas were to be placed in close proximity of one another, they could trigger a nuclear explosion which would certainly tar Kraw's environmentally friendly corporate image. Kraw stores his bananas in small lead boxes, but he'll have to take them out if he wants to give them to the cows. To prevent a nuclear explosion, Kraw must distribute his bananas according to the following rules:

- No cow gets more than one banana.
- No more than $C$ cows with consecutive house numbers can all get bananas.

So, for example, if $N = 4$ and $C = 2$, Kraw can give the cows in houses 1, 2 and 4 a banana each, but he can't give the cows in houses 2, 3 and 4 a banana each because houses 2, 3 and 4 are all next to each other and a nuclear explosion would occur.

Kraw wants to find the number of ways he can distribute his bananas to the cows, mod 1,000,000,007. Note that there is no restriction on the number of bananas that Kraw can distribute to the cows – in particular, Kraw can give the cows no bananas at all.

## Input format

Your program must read from standard input. The first and only line of the input are the two positive numbers $N$, the number of houses, and $C$, the number of consecutive houses that can all get bananas without causing a nuclear explosion. For the above example, the input is as follow:

```
4 2
```

## Output format

You program must write to the standard output a number, which is the number of ways Kraw can distribute his bananas to the cows, mod 1,000,000,007. For the input above, the output is:

```
13
```

## Explanation

When $N$ = 4 and $C$ = 2, there are 13 sets of house numbers that satisfy the above rules: {1, 2, 4}, {1, 3, 4}, {1, 2}, {1, 3}, {1, 4}, {2, 3}, {2, 4}, {3, 4}, {1}, {2}, {3}, {4}, {} (the empty set).

## More samples

Here are a few more samples.

### Input

```
3 1
```

### Output

```
5
```

### Remark

When $N$ = 3 and $C$ = 1, there are 5 sets of house numbers that satisfy the above rules: {1, 3}, {1}, {2}, {3}, {} (the empty set). Note that {1, 3} is allowed because houses 1 and 3 are not next to each other, so a nuclear explosion would not occur.

### Input

```
3 5
```

### Output

```
8
```

### Remark

When $N$ = 3 and $C$ = 5, Kraw doesn't need to worry about the second rule because there are no $C$ = 5 consecutive house numbers to begin with. So all 8 sets of house numbers satisfy the above rules: {1, 2, 3}, {1, 2}, {1, 3}, {2, 3}, {1}, {2}, {3}, {} (the empty set).

## Subtasks

The maximum execution time on each instance is 1.0s. Your program will be tested on sets of input instances as follows:

1. (11 marks) Each instance satisfies $0 < N < C < 200$;
2. (4 marks) Each instance satisfies $0 < N < 10^6, C = 1$;
3. (7 marks) Each instance satisfies $0 < N < 16, 0 < C < 16$;
4. (9 marks) Each instance satisfies $0 < N < 10^5, 0 < C < 20$;
5. (27 marks) Each instance satisfies $0 < N < 10^6, 0 < C < 200$;
6. (14 marks) Each instance satisfies $0 < N < 10^{18}, C = 1$;
7. (15 marks) Each instance satisfies $0 < N < 10^{18}, 0 < C < 50$;
8. (13 marks) Each instance satisfies $0 < N < 10^{18}, 0 < C < 200$;