

E. Maximum Element

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day Petya was solving a very interesting problem. But although he used many optimization techniques, his solution still got Time limit exceeded verdict. Petya conducted a thorough analysis of his program and found out that his function for finding maximum element in an array of n positive integers was too slow. Desperate, Petya decided to use a somewhat unexpected optimization using parameter k , so now his function contains the following code:

```
int fast_max(int n, int a[]) {
    int ans = 0;
    int offset = 0;
    for (int i = 0; i < n; ++i)
        if (ans < a[i]) {
            ans = a[i];
            offset = 0;
        } else {
            offset = offset + 1;
            if (offset == k)
                return ans;
        }
    return ans;
}
```

That way the function iteratively checks array elements, storing the intermediate maximum, and if after k consecutive iterations that maximum has not changed, it is returned as the answer.

Now Petya is interested in fault rate of his function. He asked you to find the number of permutations of integers from 1 to n such that the return value of his function on those permutations is not equal to n . Since this number could be very big, output the answer modulo $10^9 + 7$.

Input

The only line contains two integers n and k ($1 \leq n, k \leq 10^6$), separated by a space — the length of the permutations and the parameter k .

Output

Output the answer to the problem modulo $10^9 + 7$.

Examples

input
5 2
output
22

input
5 3
output
6

input
6 3
output
84

Note

Permutations from second example:

[4, 1, 2, 3, 5], [4, 1, 3, 2, 5], [4, 2, 1, 3, 5], [4, 2, 3, 1, 5], [4, 3, 1, 2, 5], [4, 3, 2, 1, 5].