Puzzle

You have a range of consecutive numbers, from 1 to n (inclusive). e.g. [1, 2, 3, 4... n].

We would like to calculate a sum of a function across the entire range, where the function returns the product of the *preceding* C elements.

If there are less than C previous elements, just use the available numbers. i.e. if you are processing the fourth number in the range, but C is greater than 3, then you will calculate the product using only the 3 available preceding numbers. In this situation as you move further along in this range more preceding numbers become available.

A worked example follows: if n=5 and C=2, the correct products and final sum for each element of the range are: 0 + 1 + 2 + 6 + 12 = 21

The solution should be implemented in Java. Initially calculate the value where n=100 and C=10, and let us know the answer

Please additionally tell us the values where n=1000000 and C=200, but only include the first 10 digits of the result in your answer.

What is the algorithmic complexity of your solution? What if you needed to calculate n=10,000,000 and C=200?