Define the k-bonacci sequence as follows. The first kterms are all 1. The nth term, for n >= k, is the sum of kprevious terms in the sequence. For example, the first five terms of the 3-bonacci sequence are 1, 1, 1, 3, 5.

Given n and k, write a function which returns the nth term (0-indexed) of the k-bonacci sequence. Since the answer may be quite large, return it as a string.

**Example**  
For k = 3 and n = 4, the output should be kbonacci(k, n) = "5".

**Input/Output**

* **[execution time limit] 3 seconds (java)**
* **[input] integer k**

The value k defining the k-bonacci sequence; i.e., the degree of the recurrence.

*Guaranteed constraints*:  
1 ≤ k ≤ 103.

* **[input] integer n**

The term of the k-bonacci sequence to return.

*Guaranteed constraints*:  
1 ≤ n ≤ 105.

* **[output] string**

The nth number of the k-bonacci sequence, returned as a string.

**[Java] Syntax Tips**

// Prints help message to the console

// Returns a string

//

// Globals declared here will cause a compilation error,

// declare variables inside the function instead!

String **helloWorld**(String name) {

System.out.println("This prints to the console when you Run Tests");

**return** "Hello, " + name;

}

<https://codefights.com/challenge/sbbmjCsPhn77Qkqtm/solutions>

*/\**

*\* To change this license header, choose License Headers in Project Properties.*

*\* To change this template file, choose Tools | Templates*

*\* and open the template in the editor.*

*\*/*

**package** javaapplication4;

**import** java.math.BigInteger;

**import** java.util.ArrayList;

***/\*\****

***\****

***\* @author Usuario***

***\*/***

**public** **class** JavaApplication4 {

***/\*\****

***\* @param args the command line arguments***

***\*/***

**static** String kbonacci(**int** k, **int** n)

        {

            ArrayList<BigInteger> lista = **new** ArrayList();

**for** (**int** i = 1; i <= k; i++)

            {

                lista.add(BigInteger.ONE);

            }

            BigInteger total = BigInteger.valueOf(k);  *// k;*

            BigInteger s = total;

**for** (**int** i = k; i < n; i++)

            {

                lista.add(s);

*//s -= lista[i - k];*

*//s += lista.Last();*

                s= s.subtract(lista.get(i-k));

                s = s.add(lista.get(lista.size()-1));

            }

**return** s.toString();

        }

**public** **static** **void** main(String[] args) {

*// TODO code application logic here*

**int** k = 15;

**int** n = 50;

*//"480889537633"*

         System.out.println(kbonacci(k,n));

    }

}