# A Sum Fun (`sumfun.{c,cc,java}`)

## A.1 Description

In mathematics, *n-nacci* numbers (for $n \geq 2$) refer to a family of integer sequences defined by the recurrence,

$$F_n(k) = \begin{cases} 0 & \text{if } k \in \{0, 1, \ldots, n-2\} \\ 1 & \text{if } k = n-1 \\ \sum_{i=1}^{n} F_n(k-i) & \text{if } n \geq k. \end{cases} \tag{1}$$

For example, the Fibonacci numbers correspond to a special case of $n$-nacci numbers in which $n = 2$. In this problem, your goal is to compute the last 9 digits of

$$\sum_{k=0}^{s-1} F_n(k). \tag{2}$$

## A.2 Input

The input file will contain multiple test cases. Each test case contains a single line with two integers $n$ and $s$ (where $2 \leq n \leq 20$ and $0 \leq s \leq 1{,}000{,}000{,}000$). The end-of-input will be denoted by a single line containing "0 0" and should not be processed.

```
3 2
2 5
3 8
0 0
```

## A.3 Output

For each input test case, print a single integer containing the last 9 digits of the sum of the first $s$ $n$-nacci integers (starting from $F_n(0)$). If the last 9 digits contain any leading zeroes, remove those before printing.

```
0
7
28
```