

代码

【故事背景】

JSOI 最近流行一种新的程序设计语言 JS，JYY 现在想分析 JS 的一些性质。

【问题描述】

JS 语言非常简单。一个 JS 语言程序可以看成是一个字符串，它没有输入，只有一个初始为 0 但可以存储任意整数的内部寄存器。它支持 5 种不同的操作：

操作一 “+”：寄存器加 1

操作二 “-”：寄存器减 1

操作三 “!x”：调用库函数 x ，根据库的版本不同共有 K 种不同的库函数， x 是一个 1 到 K 之间的整数，不会对当前寄存器造成影响。注意 “!123” 是长度为 4 的字符串。

操作四和五 “[” 和 “]”：循环。在一个合法的程序当中，[和]必须符合括号匹配，当执行到[时如果寄存器为 0 则直接跳转到对应的]，否则继续执行。执行到]时如果寄存器为 0 则继续执行，否则跳转到对应的[。

一个 JS 语言程序就是由上述操作组成的字符串。除循环操作外，操作按照从左到右顺序执行。

在写了一些程序之后，JYY 发现，有些程序会死循环，永远无法终止，而另一些则不会。现在 JYY 想计算，给定库函数的个数 K 有多少个长度为 N 的 JS 程序不会死循环。

【输入格式】

从文件 `code.in` 中读入数据。

第一行包含两个整数，分别为 N 和 K 。

【输出格式】

输出到文件 `code.out` 中。

输出一行一个整数，表示当有 K 个库函数的时候不会死循环的 JS 程序个数模 10^9+7 。

【输入样例 1】

3 1

【输出样例 1】

16

【输入样例 2】

5 9

【输出样例 2】

1010

【样例说明】

第一个样例中，库函数的调用只有一种!1，不会死循环的程序有以下 16 种：

+++	+++	+++	+++	[]+	++-	+-	---
---	[]-	[+]	[-]	!1+	!1-	+!1	-!1

【数据规模】

对于 30%的数据满足 $N \leq 10$ ；

对于 50%的数据，满足 $N \leq 50$ ；

对于 100%的数据，满足 $1 \leq N \leq 100$, $1 \leq K < 10^4$ 。