

题目描述

- 3020年，空间通信集团的员工人数突破20亿人，即将遇到现有工号不够用的窘境。
- 现在，请你负责调研新工号系统。继承历史传统，新的工号系统由小写英文字母（a-z）和数字（0-9）两部分构成。
- 新工号由一段英文字母开头，之后跟随一段数字，比如"aaahw0001","a12345","abcd1","a00"。
- 注意新工号不能全为字母或者数字,允许数字部分有前导0或者全为0。
- 但是过长的工号会增加同事们的记忆成本，现在给出新工号至少需要分配的人数X和新工号中字母的长度Y，求新工号中数字的最短长度Z。

输入描述

- 一行两个非负整数 X Y，用数字用单个空格分隔。
- $0 < X \leq 2^{50} - 1$
- $0 < Y \leq 5$

输出描述

- 输出新工号中数字的最短长度Z

用例

输入	260 1
输出	1
说明	无

输入	26 1
输出	1
说明	数字长度不能为0

输入	2600 1
输出	2
说明	无

题目解析

这题应该就是一道数学问题。

小写字母有26种，数字有10种，因此如果工号组合选择一个字母，一个数字的话，则有 $26 * 10 = 260$ 种。

如果选择两个字母，两个数字的话，则会产生 $26^2 * 10^2$ 种工号。

现在确定了需要的工号总个数x，以及字母个数y，也就是说

$$x = 26^y * 10^z$$

求最小的z，且 $z \geq 1$ 。

因此z的求解公式：

$$z = \log(x / 26^y)$$

这里我们要保证z 向上取整，且保证z最小取1

JavaScript算法源码

```
1  /* JavaScript Node ACP模式 控制台输入获取 */
2  const readline = require("readline");
3
4  const rl = readline.createInterface({
5    input: process.stdin,
6    output: process.stdout,
7  });
8
9  rl.on("line", (line) => {
10    const [x, y] = line.split(" ").map(Number);
11
12    console.log(Math.max(1, Math.ceil(Math.log10(x / Math.pow(26, y)))));
13  });
```

Java算法源码

```
1  import java.util.Scanner;
2
3  public class Main {
4    public static void main(String[] args) {
5      Scanner sc = new Scanner(System.in);
6
7      int x = sc.nextInt();
8      int y = sc.nextInt();
9
10     System.out.println((int) Math.max(1, Math.ceil(Math.log10(x / Math.pow(26, y)))));
11   }
12 }
```

Python算法源码

```
1  # 输入获取
2  import math
3
4  x, y = map(int, input().split())
5
6
7  # 算法入口
8  def getResult(x, y):
9    print(max(1, math.ceil(math.log10(x / math.pow(26, y))))
10
11
12 # 算法调用
13 getResult(x, y)
```