

P3612 [USACO17JAN]Secret Cow Code S

题目描述

奶牛正在试验秘密代码，并设计了一种方法来创建一个无限长的字符串作为其代码的一部分使用。

给定一个字符串，让后面的字符旋转一次（每一次正确的旋转，最后一个字符都会成为新的第一个字符）。也就是说，给定一个初始字符串，之后的每一步都会增加当前字符串的长度。

给定初始字符串和索引，请帮助奶牛计算无限字符串中位置N的字符。

输入格式

第一行输入一个字符串。该字符串包含最多30个大写字母，并 $N \leq 10^{18}$ 。

第二行输入N。请注意，数据可能很大，放进一个标准的32位整数可能不够，所以你可能要使用一个64位的整数类型（例如，在C / C++ 中是 long long）。

输出格式

请输出从初始字符串生成的无限字符串中的位置的字符。第一个字符是 N=1.。

输入样例

COW 8

输出样例

C

解析

我们观察到每一次的更新其实都是原有字符的一次翻倍，因此我们首先考虑其实至少一半的字符都是没有用处的，因为这两部分的字符是完全相同的。如下所示：

1	2	3	4	5	6
C	O	W	W	C	O

减半后发现n所处的字符比正确的字符向右偏移了一位，因此我们只需将其向左移动1位即可，如下图所示，红色箭头是减半后的位置，绿色箭头是向前移动的最终结果。

1	2	3	4	5	6	7	8	9	10	11	12
C	O	W	W	C	O	O	C	O	W	W	C

再来看这样一种情况，如果减半过后的字符串其实也是由原字符串的上一次翻倍得到的，该怎么处理呢？例如，我们现在需要求第11个字符，减半后指向了第5个字符C，同理，我们需要进行偏移，将其指向第4个字符。

1	2	3	4	5	6	7	8	9	10	11	12
C	O	W	W	C	O	O	C	O	W	W	C

此时，第4个字符并不在原始的字符范围内，我们需要继续对其进行减半，并偏移索引。特别注意，当求出的索引为0时，则代表最终结果为原始字符的最后一个，即为W。

0	1	2	3	4	5	6
	C	O	W	W	C	O

所以，我们得出了这样的结论：

对于给定的目标n，我们只需要不停的进行砍半，直到n已经在原有字符串范围内了，而具体对应的哪个字符，因为每次都发生了位移，因此想要确定在原有字符串的位置，那么每一次进行减半时，都要进行归位，即将 $n - 1$

编码

```
#include<bits/stdc++.h>

using namespace std;
string str;
long long n, temp;

int main() {
    cin >> str >> n;
    int len = str.size();
    //只要n还大于原有字符串就要不断进行砍半
```

```
while (len < n) {  
    temp = len; //将原有长度用临时变量记录，不然原有长度会被改变  
    while (temp * 2 < n) {  
        temp *= 2; //找到该砍半的长度  
    }  
    n -= (temp + 1); //将其砍半，并进行归位  
    if (!n) {  
        n = temp; //如果是最后一位直接输出最后一位  
    }  
}  
cout << str[n - 1];  
return 0;  
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

