

Tipos Primitivos de Dados

Problemas resolvidos

Prof. Edson Alves

Faculdade UnB Gama

1. AtCoder Beginner Contest 106 – Problem C: To Infinity
2. OJ 11879 – Multiple of 17

AtCoder Beginner Contest 106 – Problem C: To Infinity

Problema

Mr. Infinity has a string S consisting of digits from 1 to 9. Each time the date changes, this string changes as follows:

- Each occurrence of '2' in S is replaced with '22'. Similarly, each '3' becomes '333', '4' becomes '4444', '5' becomes '55555', '6' becomes '666666', '7' becomes '7777777', '8' becomes '88888888' and '9' becomes '999999999'. '1' remains as '1'.

For example, if S is '1324', it becomes '1333224444' the next day, and it becomes '133333333322224444444444444444' the day after next. You are interested in what the string looks like after 5×10^{15} days. What is the K -th character from the left in the string after 5×10^{15} days?

Constraints

- S is a string of length between 1 and 100 (inclusive).
- K is an integer between 1 and 10^{18} (inclusive).
- The length of the string after 5×10^{15} days is at least K .

Input

Input is given from Standard Input in the following format:

$$S$$
$$K$$

Output

Print the K -th character from the left in Mr. Infinity's string after 5×10^{15} days.

Exemplo de entradas e saídas

Exemplo de Entrada

1214

4

3

157

299792458

9460730472580800

Exemplo de Saída

2

3

2

- O primeiro fato a ser observado é, se o número inicial x da string for diferente de 1, ele será replicado $x^{5 \times 10^{15}}$ vezes, de modo que a resposta será o próprio x
- O caso especial ocorre quando a string é prefixada por uma sequência de 1s
- Se a quantidade de uns for maior ou igual a K , a resposta será igual a 1
- Caso contrário, a resposta será igual ao primeiro caractere da string diferente de 1

Solução AC

```
1 #include <iostream>
2
3 using namespace std;
4 using ll = long long;
5
6 char solve(const string& S, ll K)
7 {
8     for (const auto& c : S)
9     {
10         if (c != '1')
11             return c;
12
13         if (not (--K))
14             break;
15     }
16
17     return '1';
18 }
19
```



```
20 int main()
21 {
22     ios::sync_with_stdio(false);
23
24     string S;
25     ll K;
26
27     cin >> S >> K;
28
29     cout << solve(S, K) << '\n';
30
31     return 0;
32 }
```

OJ 11879 – Multiple of 17

Theorem: *If you drop the last digit d of an integer n ($n \geq 10$), subtract $5d$ from the remaining integer, then the difference is a multiple of 17 if and only if n is a multiple of 17.*

For example, 34 is a multiple of 17, because $3-20=-17$ is a multiple of 17; 201 is not a multiple of 17, because $20-5=15$ is not a multiple of 17.

Given a positive integer n , your task is to determine whether it is a multiple of 17.

Input

There will be at most 10 test cases, each containing a single line with an integer n ($1 \leq n \leq 10^{100}$).

The input terminates with $n = 0$, which should not be processed.

Output

For each case, print 1 if the corresponding integer is a multiple of 17, print 0 otherwise.

Exemplo de entradas e saídas

Exemplo de Entrada

34

201

2098765413

1718

0

Exemplo de Saída

1

0

1

0

- O limite máximo da entrada não pode ser armazenada em tipos primitivos do C/C++
- Uma alternativa é ler a entrada como strings, e aplicar o algoritmo descrito, acumulando no total S , para cada dígito d , o valor $-5d$
- Em seguida, basta verificar se S é ou não múltiplo de 17
- Outra alternativa é utilizar uma linguagem que tenha suporte nativo para aritmética estendida
- Uma opção é a linguagem Java e sua classe BigInteger
- Outra opção é usar a linguagem Python

```
1 import sys
2
3 xs = [int(x) for x in sys.stdin.readlines()][:-1]
4 ans = ['1' if x % 17 == 0 else '0' for x in xs]
5
6 print('\n'.join(ans))
```

1. [AtCoder Beginner Contest 103 – Problema C: To Infinity](#)
2. [OJ 11879 – Multiple of 17](#)