***Giải thuật:*** *Xét một giá trị k nguyên tố, 1< k <109.*

*Nhận xét:*

* Nếu ***k*** < 10 – có nghiệm ***m*** = 1,
* Trường hợp ***k***>10:
* Nếu một số chỉ chứa các chữ số ***x*** (1 < ***x*** ≤ 9) chia hết cho ***k*** thì số nhỏ hơn, với các chữ số tương ứng là 1 cũng chia hết cho ***k***, như vậy nếu tồn tại ***m*** thì ***k***×***m*** là một số chỉ chứa chữ số 1,
* Gọi ***zi*** là số dư của 10***i*** chia cho ***k***, ***i***=0, 1, 2, . . ., ***k***-2,
* Dễ dàng chứng minh (bằng phản chứng): ***zi*** ≠ ***zj*** với ***i*** ≠ ***j***, 0 ≤ ***i***, ***j*** ≤ ***k***-2,
* Như vậy có số dư chia cho ***k*** bằng số dư của

**111…11**

*k số 1*

=  = 

chia cho ***k*** và bằng 0,

**111…11**

*k số 1*

Nói một cách khác số chia hết cho ***k***, tức là tồn tại ***m*** để ***k***×***m*** cho kết quả là một số chỉ chứa các chữ số 1.

* Như vậy, bài toán luôn có nghiệm.
* Tuy vậy, có thể số nhỏ hơn, chứa ***q*** chữ số 1(***q*** < k-1) và chia hết cho ***k***,
* Cũng bằng phương pháp phản chứng có thể chứng minh được rằng ***q*** phải là một ước của ***k***-1.
* Như vậy ta chỉ cần duyệt các số có ***u*** chữ số 1, trong đó ***u*** là ước của ***k***-1 và tìm ***u*** nhỏ nhất và giải phương trình:

***k×m =***

**111…11**

*u số 1*

* tìm ***m*** theo mô đun 109+7.
* Số ở vế phải của phương trình nêu trên phải lớn hơn hoặc bằng ***k***, vì vậy ta chỉ cần xét các giá trị ***u*** lớn hơn hoặc bằng số chữ số của ***k*** trong hệ cơ số 10.

*Tổ chức dữ liệu:* ***vector<int> a,b,c;***

* ***ai*** – thừa số nguyên tố thứ ***i*** khi phân tích ***k***-1 ra thừa số nguyên tố,
* ***bi*** – bậc của ***ai***,
* ***cj*** – ước số thứ ***j*** của ***k***-1.
* ***ai***, ***cj*** – sắp xếp tăng dần.

*Xử lý:*

Gán ***x*** = ***k***-1; tính ***lk*** – số chữ số của ***x***:

**t=x; lk=0;**

**while(t>0){++lk;t/=10;}**

Phân tích ***x*** ra thừa số nguyên tố

Tìm tất cả các ước của ***x*** , gọi ***md*** – số ước tính được,

Với mọi ***i*** = 1 ÷ ***md***-1: nếu ***ci*** ≥ ***lk*** → kiểm tra số có ***ci*** chữ số 1 chia hết cho ***k*** hay không, nếu chia hết thì ***ci*** là số ***u*** cần tìm.

**111…11**

*ci số 1*

=



Do ***k*** > 9 nên ta chỉ cần kiểm tra tử số có chia hết cho ***k*** hay không.

Việc tính số dư của ***xy*** chia cho ***v*** được thực hiện theo sơ đồ tính nhanh lũy thừa (tương tự sơ đồ nhân Ai Cập).

Để giải phương trình:

***k×m =***

**111…11**

*u số 1*

Trong đó m là ẩn số và ***u*** = ***ci***, trước hết cần tính giá trị vế phải theo mô đun 109+7.

Để tránh phải làm việc với số vượt quá khả năng lưu trữ kiểu ***uint64\_t***, cần cải tiến sơ đồ tính toán:

**int64\_t calc\_pw1(int u)**

**{int64\_t tx,ty,tr;**

**tx=1,ty=10; if(u&1)tr=1;else tr=0; u>>=1;**

**while(u)**

**{**

**tx=(tx\*ty+tx)%pp;**

**ty=ty\*ty%pp;**

**if(u&1)tr=(tr\*ty+tx)%pp;**

**u>>=1;**

**}**

**return tr;**

**}**

Ta có phương trình

***k×m = z***, (\*)

trong đó ***z = calc\_pw1(u);***

Với ***p*** là một số nguyên tố, ta có ***f(i) = ki%p*** là một hàm tuần hoàn với chu kỳ tuần hoàn là ***p***-1 và ***f***(0) = 1.

Nhân cả 2 vế của (\*) với ***kp***-2 và lấy mô đun ***p*** ta có:

***(kp-1×m)%p*** ***=*** ***z×kp-2%p***

***1×m%p = z×kp-2%p***

Việc tính ***kp-2%p*** được thực hiện theo sơ đồ tính nhanh lũy thừa đã nêu ở trên.

*Độ phức tạp của giải thuật:* ≈O(*n×k0.5*).