

## Number Theory

### Phần 1. Các bài toán về số nguyên tố

#### Bài 1. Kiểm tra số nguyên tố

##### Input

Số nguyên  $n$  ( $0 \leq n \leq 10^9$ ).

##### Output

In YES nếu  $n$  là số nguyên tố, ngược lại in NO.

##### Ví dụ

Input	Output
4	NO
13	YES

#### Bài 2. Sàng số nguyên tố.

##### Input

Số nguyên  $n$  ( $0 \leq n \leq 10^6$ ).

##### Output

4

In ra trên một dòng các số nguyên tố không vượt quá  $n$ , mỗi số cách nhau một khoảng trắng.

##### Ví dụ

Input	Output
4	2 3

13	2 3 5 7 11 13
----	---------------

**Bài 3.** Sàng số nguyên tố trên đoạn.

**Input**

2 số nguyên không âm  $a, b (0 \leq a \leq b \leq 10^9, b - a \leq 10^5)$ .

**Output**

In ra các số nguyên tố trong đoạn từ  $a$  tới  $b$  (Chú ý lấy cả 2 cận).

Ví dụ

Input	Output
4 20	5 7 11 13 17 19
1 5	2 3 5

**Bài 4.** Kiểm tra số nguyên tố 2.

**Input**

Dòng đầu tiên là số lượng test case  $T. (1 \leq T \leq 1000)$ .

Mỗi test case là một số nguyên  $n (0 \leq n \leq 10^6)$ .

**Output**

In ra kết quả mỗi test case trên một dòng. In YES nếu  $n$  là số nguyên tố, ngược lại in NO. Ví

dụ

Input	Output
-------	--------

4	YES
2	YES
3	NO
20	NO
188	

### **Bài 5.** Số nguyên tố và chữ số nguyên tố

Viết chương trình đếm xem trong đoạn  $[a,b]$  có bao nhiêu số là số nguyên tố và tất cả các chữ số của nó cũng là số nguyên tố.

#### **Input**

Dòng đầu ghi số bộ test

Mỗi bộ test ghi 2 số  $a, b$  ( $1 < a < b < 10^6$ )

#### **Output**

Với mỗi bộ test, ghi ra số lượng số thỏa mãn trên một dòng.

#### **Ví dụ**

Input	Output
2	4
10 100	26
1234 5678	

### **Bài 6.** Liệt kê N số nguyên tố đầu tiên

Viết chương trình liệt kê N số nguyên tố đầu tiên với N là một số nguyên dương không quá  $10^5$ .

### Input

Dữ liệu vào chỉ có duy nhất một số N.

### Output

Kết quả ghi mỗi số nguyên tố trên một dòng, theo thứ tự từ nhỏ đến lớn.

### Ví dụ

Input	Output
5	2 3 5 7 11

### Bài 7. Cặp số nguyên tố.

Cho số nguyên dương chẵn  $N > 2$ . Hãy liệt kê các cặp số nguyên tố p, q có tổng đúng bằng N.

Ví dụ  $N = 6$  ta có 1 cặp số nguyên tố là  $3 + 3 = 6$ .

### Input

Dòng đầu tiên đưa vào số lượng bộ test T.

Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số chẵn N. T,

N thỏa mãn ràng buộc :  $1 \leq T \leq 100$ ;  $4 \leq N \leq 10000$ .

### Output

Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ**

Input	Output
2	2 2
4	3 3
6	

**Bài 8.** Số thuần nguyên tố.

Một số được coi là thuần nguyên tố nếu nó là số nguyên tố, tất cả các chữ số là nguyên tố và tổng chữ số của nó cũng là một số nguyên tố. Bài toán đặt ra là đếm xem trong một đoạn giữa hai số nguyên cho trước có bao nhiêu số thuần nguyên tố.

**Input**

Dòng đầu tiên ghi số bộ test. Mỗi bộ test viết trên một dòng hai số nguyên dương tương ứng, cách nhau một khoảng trống. Các số đều không vượt quá 9 chữ số.

**Output**

Mỗi bộ test viết ra số lượng các số thuần nguyên tố tương ứng

**Ví dụ**

Input	Output
2	1
23 199	15
2345 6789	

**Bài 9.** Nguyên tố cùng nhau.

Viết chương trình nhập hai số nguyên dương  $a, b$  thỏa mãn  $2 < a < b < 100$ .

Một cặp số  $(i,j)$  được gọi là nguyên tố cùng nhau nếu  $i \neq j$  và ước số chung lớn nhất của  $i$  với  $j$  là 1

Hãy liệt kê các cặp số nguyên tố cùng nhau trong đoạn  $[a,b]$  theo thứ tự từ nhỏ đến lớn.

### Input

Chỉ có một dòng ghi hai số  $a,b$

### Output

Các cặp số  $i,j$  thỏa mãn được viết lần lượt trên từng dòng theo định dạng  $(i,j)$ , theo thứ tự từ điển.

### Ví dụ

Input Output	
5 8	(5,6)
	(5,7)
	(5,8)
	(6,7)
	(7,8)

### Bài 10. T-prime

Cho số tự nhiên  $N$ . Nhiệm vụ của bạn là hãy liệt kê tất cả các số có đúng ba ước số. Ví dụ  $n=100$ , ta có các số 4, 9, 25, 49.

### Input

Dòng đầu tiên đưa vào số lượng test  $T$ .

Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số N.

T, N thỏa mãn ràng buộc  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^6$ .

### Output

Đưa ra kết quả mỗi test theo từng dòng.

### Ví dụ

Input: Output:	
2	4 9 25 49
50	4 9 25 49 121 169
200	

### Bài 11. T-Prime 2

Cho số tự nhiên N. Nhiệm vụ của bạn là hãy đếm tất cả các số có đúng ba ước số. Ví dụ  $n=100$ , ta có các số 4.

### Input

Dòng đầu tiên đưa vào số lượng test T.

Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số N.

T, N thỏa mãn ràng buộc  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^{12}$ .

### Output

Đưa ra kết quả mỗi test theo từng dòng.

### Ví dụ

Input: Output:	
2	4
50	6
200	

### Bài 12. T-prime 3

Cho hai số  $L, R$ . Nhiệm vụ của bạn là hãy đếm tất cả các số có đúng ba ước số trong khoảng  $[L, R]$ . Ví dụ  $L=1, R=10$ , ta có kết quả là 2 vì chỉ có số 3 và 9 là có đúng 3 ước số.

#### Input

Dòng đầu tiên đưa vào số lượng test  $T$ .

Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là cặp số  $L, R$ .

$T, N$  thỏa mãn ràng buộc  $1 \leq T \leq 100; 1 \leq L, R \leq 10^{12}$ .

#### Output

Đưa ra kết quả mỗi test theo từng dòng.

#### Ví dụ

Input: Output:	
2	2
1 10	78498



**Bài 13.** Ước số nguyên tố nhỏ nhất.( Sử dụng sàng biến đổi).

Cho số tự nhiên N. Nhiệm vụ của bạn là in ra ước số nguyên tố nhỏ nhất của các số từ 1 đến N. Ước số nguyên tố nhỏ nhất của 1 là 1. Ước số nguyên tố nhỏ nhất của các số chẵn là 2. Ước số nguyên tố nhỏ nhất của các số nguyên tố là chính nó.

**Input**

Dòng đầu tiên đưa vào số lượng test T.

Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số N được ghi trên một dòng. T,

N thỏa mãn ràng buộc:  $1 \leq T \leq 1000$ ;  $1 \leq N \leq 100000$ .

**Output:**

Đưa ra kết quả mỗi test theo từng dòng.

Input: Output:	
2	1 2 3 2 5 2
6	1 2 3 2 5 2 7 2 3 2
10	

**Phần 2. Phân tích thừa số nguyên tố.**

**Bài 1.** Phân tích 1

Hãy phân tích một số nguyên dương n thành thừa số nguyên tố

**Input**

Số nguyên dương n ( $1 \leq n \leq 10^9$ )

**Output**

Cách phân tích thừa số nguyên tố của n. Bạn hãy thử cài đặt với 5 cách in thừa số nguyên tố sau.

Ví dụ cách phân tích 1.

Input: Output:	
28	2 2 7

Ví dụ cách phân tích 2.(Mỗi thừa số nguyên tố chỉ liệt kê 1 lần).

Input: Output:	
28	2 7

Cách phân tích 3. Thừa số nguyên tố được liệt kê kèm theo số mũ.

Input: Output:	
28	2(2) 7(1)

Cách phân tích 4. Thêm dấu x vào giữa các thừa số nguyên tố

Input: Output:	
8	2x2x2
28	2x2x7

Cách phân tích 5.

Input: Output:	
60	$60 = 2^2 * 3^1 * 5^1$

**Bài 2.** Phân tích 2.(Sử dụng sàng số nguyên tố biến đổi).

Hãy phân tích một số nguyên dương thành tích các thừa số nguyên tố.

## Input

Dòng đầu tiên ghi số bộ test.

Mỗi bộ test viết trên một dòng số nguyên dương  $n$  không quá 5 chữ số.

## Output

Mỗi bộ test viết ra thứ tự bộ test, sau đó lần lượt là các số nguyên tố khác nhau có trong tích, với mỗi số viết thêm số lượng số đó. Xem ví dụ để hiểu rõ hơn về cách viết kết quả.

## Ví dụ

Input: Output:	
3	#TC1: 2(2) 3(1) 5(1)
60	#TC2: 2(7)
128	#TC3: 2(4) 5(4)
10000	

## Bài 3. Đếm thừa số nguyên tố.

Hãy đếm số lượng thừa số nguyên tố khác nhau trong phân tích thừa số nguyên tố của 1 số nguyên dương  $n$ .

## Input

Dòng đầu tiên là số lượng test case  $T(1 \leq n \leq 100)$ .

T dòng tiếp theo mỗi dòng là một số nguyên dương  $n (1 \leq n \leq 10^9)$

## Output

Số lượng thừa số nguyên tố khác nhau của  $n$ .

## Ví dụ

Input: Output:	
3	3
60	1
128	2
10000	

#### Bài 4. Lũy thừa và giai thừa.

Cho số tự nhiên  $N$  và số nguyên tố  $P$ . Nhiệm vụ của bạn là tìm số  $x$  lớn nhất để  $N!$  chia hết cho  $p^x$ . Ví dụ với  $N=7$ ,  $p=3$  thì  $x=2$  là số lớn nhất để  $7!$  Chia hết cho  $3^2$ .

#### Input

Dòng đầu tiên đưa vào số lượng bộ test  $T$ .

Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là cặp số  $N$ ,  $p$  được viết cách nhau một vài khoảng trống.

$T$ ,  $N$ ,  $p$  thỏa mãn ràng buộc :  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^5$ ;  $2 \leq p \leq 5000$ ;

#### Output

Đưa ra kết quả mỗi test theo từng dòng.

#### Ví dụ.

Input: Output:
----------------

3	76 2
62 7	3 5

**Bài 5.** Đếm số lượng chữ 9

số 0 của  $n!$ . **Input**            73 0

Dòng đầu tiên là số lượng test case  $T(1 \leq n \leq 100)$ .

T dòng tiếp theo mỗi dòng là một số nguyên dương  $n(1 \leq n \leq 10^6)$

**Output**

In ra số lượng chữ số 0 liên tiếp tính từ cuối của  $n!$ .

**Ví dụ**

Input: Output:	
2	2
10	4
20	

Giải thích :

$10! = 3628800$ . Có 2 chữ số 0 liên tiếp tính từ cuối.

Link submit : <https://cses.fi/problemset/task/1618>

Chú ý bạn phải code theo đúng yêu cầu của đầu bài.

**Bài 6.** Số Sphenic.

Số nguyên dương  $N$  được gọi là số Sphenic nếu  $N$  được phân tích duy nhất dưới dạng tích của ba số khác nhau. Ví dụ  $N=30$  là số Sphenic vì  $30 = 2 \times 3 \times 5$ ;  $N = 60$  không phải số

Sphenic vì  $60 = 2 \times 2 \times 3 \times 5$ . Cho số tự nhiên  $N$ , nhiệm vụ của bạn là kiểm tra xem  $N$  có phải số Sphenic hay không?

Một số số Sphenic đầu tiên : 30, 42, 66, 70, 78, 102

### Input

Dòng đầu tiên đưa vào số lượng bộ test  $T$ .

Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số nguyên dương  $N$ .  $T$ ,

$N$  thỏa mãn ràng buộc :  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10000$ .

### Output

Đưa ra 1 hoặc 0 tương ứng với  $N$  là số Sphenic hoặc không của mỗi test theo từng dòng. Ví

dụ

Input: Output:	
2	1
30	0
60	

### Bài 7. Số Smith.

Cho số tự nhiên  $N$ . Nhiệm vụ của bạn là hãy kiểm tra  $N$  có phải là số Smith hay không. Một số được gọi là số Smith nếu  $N$  không phải là số nguyên tố và có tổng các chữ số của  $N$  bằng tổng các chữ số của các ước số nguyên tố của  $N$ . Ví dụ  $N = 666$  có các ước số nguyên tố là 2, 3, 3, 37 có tổng các chữ số là 18.

Input:

Dòng đầu tiên đưa vào số lượng test  $T$ .

Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số nguyên  $N$ .  $T$ ,

$N$  thỏa mãn ràng buộc  $1 \leq T \leq 100$ ;  $1 \leq N \leq 100000$ .

**Output**

Đưa ra kết quả mỗi test theo từng dòng.

Input: Output:	
2	YES
4	YES
666	

**Bài 8.** Ước số nguyên tố lớn nhất của số nguyên dương.

**Input**

Dòng đầu tiên là số lượng test case  $T (1 \leq n \leq 100)$ .

$T$  dòng tiếp theo mỗi dòng là một số nguyên dương  $n (1 \leq n \leq 10^6)$

**Output**

Ước số nguyên tố lớn nhất của  $n$  in ra mỗi test case trên 1 dòng.

**Ví dụ**

Input	Output
2	5
10	17
17	

**Bài 9 .** Bình phương nguyên tố.

Một số được coi là số đẹp khi nó đồng thời vừa chia hết cho một số nguyên tố và chia hết cho bình phương của số nguyên tố đó. Viết chương trình liệt kê các số đẹp

như vậy trong đoạn giữa hai số nguyên dương cho trước.

### Input

2 số nguyên dương  $a, b$  ( $1 \leq a \leq b \leq 10^6$ ).

### Output

In ra các số đẹp trong đoạn từ  $a$  tới  $b$ .

Ví dụ

Input	Output
1 50	4 8 9 12 16 18 20 24 25 27 28 32 36 40 44 45 48 49 50

## Bài 10. Bình phương nguyên tố 2.

Một số được coi là số đẹp khi nếu nó chia hết cho một số nguyên tố nào đó thì cũng chia hết cho bình phương của số nguyên tố đó. Viết chương trình liệt kê các số đẹp như vậy trong đoạn giữa hai số nguyên dương cho trước

### Input

2 số nguyên dương  $a, b$  ( $1 \leq a \leq b \leq 10^6$ ).

### Output

In ra các số đẹp trong đoạn từ  $a$  tới  $b$ .

Ví dụ

Input	Output
1 50	4 8 9 16 25 27 32 36 49

## Phần 3. Tổng hợp (Số thuận nghịch, số chính phương, số fibonacci...).

### Bài 1. Số thuận nghịch.

Kiểm tra số thuận nghịch.



**Input**

Dòng đầu tiên là số lượng test case  $T(1 \leq n \leq 100)$ .

T dòng tiếp theo mỗi dòng là một số nguyên dương  $n(1 \leq n \leq 10^{18})$

**Output**

Mỗi test case in trên 1 dòng, in YES nếu  $n$  là số thuận nghịch, NO trong trường hợp ngược lại.

**Ví dụ**

Input	Output
2	NO
10019	YES
9999999999999999	

**Bài 2. Số chính phương.**

Kiểm tra số chính phương.

**Input**

Dòng đầu tiên là số lượng test case  $T(1 \leq n \leq 100)$ .

T dòng tiếp theo mỗi dòng là một số nguyên dương  $n(1 \leq n \leq 10^{18})$

**Output**

Mỗi test case in trên 1 dòng, in YES nếu  $n$  là số chính phương, NO trong trường hợp ngược lại.

**Ví dụ.**

Input	Output
2	NO
24	

1000000000000000000	YES
---------------------	-----

**Bài 3.** Số chính phương trong đoạn.

In ra các số chính phương trong đoạn từ a tới b.

**Input**

Dòng đầu tiên là số lượng test case  $T(1 \leq n \leq 100)$ .

T dòng tiếp theo mỗi dòng là 2 số nguyên dương a, b ( $1 \leq a \leq b \leq 10^6$ )

**Output**

Mỗi test case in ra trên 1 dòng.

**Ví dụ**

Input	Output
2	1 4 9 16 25 36 49
1 50	16
10 20	

**Bài 4.** Đếm số lượng chính phương trong đoạn.

Đếm số lượng các số chính phương trong đoạn từ a tới b.

**Input**

Dòng đầu tiên là số lượng test case  $T(1 \leq n \leq 100)$ .

T dòng tiếp theo mỗi dòng là 2 số nguyên dương a, b ( $1 \leq a \leq b \leq 10^6$ )

**Output**

Mỗi test case in ra trên 1 dòng.

### Ví dụ

21

Input	Output
2	7
1 50	1
10 20	

**Bài 5.** Tính tổng ước của 1 số nguyên dương  $n$ .

### Input

Dòng đầu tiên là số lượng test case  $T(1 \leq n \leq 100)$ .

T dòng tiếp theo mỗi dòng là 1 số nguyên dương  $n(1 \leq n \leq 10^9)$

### Output

Mỗi test case in ra trên 1 dòng.

### Ví dụ

Input	Output
2	18
10	56
28	

**Bài 6.** Đếm số lượng ước của 1 số nguyên dương  $n$ .

### Input

Dòng đầu tiên là số lượng test case  $T(1 \leq n \leq 100)$ .

T dòng tiếp theo mỗi dòng là 1 số nguyên dương  $n(1 \leq n \leq 10^9)$

### Output

Mỗi test case in ra trên 1 dòng.

### Ví dụ

22

Input	Output
2	4
10	6
28	

### Bài 7. Số có ước lẻ.

Kiểm tra xem một số có số lượng ước số của nó là số lẻ.

#### Input

Dòng đầu tiên là số lượng test case  $T(1 \leq n \leq 100)$ .

T dòng tiếp theo mỗi dòng là 1 số nguyên dương  $n(1 \leq n \leq 10^{18})$

#### Output

Mỗi test case in ra trên 1 dòng. YES nếu  $n$  có số lượng ước lẻ, ngược lại in NO. **Ví**

#### dụ

Input	Output
1	YES
1000000000000000000	

### Bài 8. Số hoàn hảo.

Số hoàn hảo là số có tổng các ước thực sự (Không tính chính nó) bằng chính số đó.

Cho một số nguyên dương  $n$ , kiểm tra xem  $n$  có phải là số hoàn hảo hay không. **Input**

Dòng đầu tiên là số lượng test case  $T(1 \leq n \leq 100)$ .

T dòng tiếp theo mỗi dòng là 1 số nguyên dương  $n$  ( $1 \leq n \leq 10^{18}$ )

### Output

23

Mỗi test case in ra trên 1 dòng. YES nếu  $n$  là số hoàn hảo, ngược lại in NO.

### Ví dụ

Input	Output
2	YES
28	YES
2305843008139952128	

### Bài 9. Tổng chữ số.

Tính tổng chữ số của 1 số nguyên dương  $n$ .

### Input

Dòng đầu tiên là số lượng test case  $T$  ( $1 \leq T \leq 100$ ).

T dòng tiếp theo mỗi dòng là 1 số nguyên dương  $n$  ( $1 \leq n \leq 10^{18}$ )

### Output

Mỗi test case in ra trên 1 dòng tổng các chữ số của  $n$ .

### Ví dụ

Input	Output
2	1
1000000000000000000	7
124	

### Bài 10. Số tăng giảm.

Một số được gọi là số tăng giảm nếu số đó có các chữ số thỏa mãn hoặc tăng

dần, hoặc giảm dần từ trái qua phải.

Hãy đếm các số nguyên tố là số tăng giảm với  $n$  là số chữ số cho trước.

### Input

24

Số nguyên dương duy nhất  $n$  ( $1 \leq n \leq 6$ ).

### Output

In ra số lượng số tăng giảm là số nguyên tố có  $n$  chữ số.

### Ví dụ

Input	Output
2	20

### Bài 11. Fibonacci

Dãy số Fibonacci được định nghĩa như sau:  $F_0 = 0$ ,  $F_1 = 1$ ;  $F_i = F_{i-1} + F_{i-2}$ . Cho số nguyên dương  $n$ , với  $2 \leq n \leq 92$ . Hãy viết chương trình in ra  $n$  số Fibonacci đầu tiên.

### Input

Số nguyên dương  $n$  ( $2 \leq n \leq 92$ )

### Output

$n$  số fibonacci đầu tiên, mỗi số được in cách nhau một dấu cách.

### Ví dụ

Input	Output
5	0 1 1 2 3

### Bài 12. Kiểm tra số fibonacci.

Nhập vào một số và kiểm tra xem số vừa nhập có phải là số trong dãy fibonacci hay không?

### Input

Dòng đầu tiên là số lượng test case  $T(1 \leq n \leq 100)$ .

25

T dòng tiếp theo mỗi dòng là 1 số nguyên dương  $n(1 \leq n \leq 10^{18})$

### Output

Mỗi test case in trên 1 dòng, in YES nếu  $n$  là số fibonacci, ngược lại in NO.

### Ví dụ

Input	Output
3	YES
2	NO
4	YES
420196140727489673	

### Bài 13. Số đẹp

Một số được coi là đẹp nếu nó là số nguyên tố và tổng chữ số là một số trong dãy Fibonacci. Viết chương trình liệt kê trong một đoạn giữa hai số nguyên cho trước có bao nhiêu số đẹp như vậy

### Input

Dòng duy nhất là 2 số nguyên dương  $a, b(1 \leq a \leq b \leq 10^9)$ .

### Output

In ra các số đẹp trong đoạn từ  $a$  tới  $b$ . Trong trường hợp không tồn tại số đẹp trong đoạn từ  $a$  tới  $b$  thì in ra -1.

### Ví dụ

Input	Output
-------	--------

2 50	2 3 5 11 17 23 41
24 30	-1

26

**Bài 14.** Thuận nghịch và có 3 ước số nguyên tố.

Một số được coi là đẹp nếu nó là số thuận nghịch và có ít nhất 3 ước số nguyên tố khác nhau. Viết chương trình in ra các số đẹp như vậy trong một đoạn giữa hai số nguyên cho trước

#### Input

Dòng duy nhất là 2 số nguyên dương  $a, b (1 \leq a \leq b \leq 10^9)$ .

#### Output

In ra các số đẹp trong đoạn từ  $a$  tới  $b$ . Trong trường hợp không tồn tại số đẹp trong đoạn từ  $a$  tới  $b$  thì in ra -1.

Input	Output
1 1000	66 222 252 282 414 434 444 474 494 525 555 585 595 606 616 636 646 666 696 777 828 858 868 888 969

**Bài 15.** Đếm chữ số chẵn, lẻ.

Nhập một số nguyên dương  $N$  không quá 9 chữ số. Hãy đếm xem  $N$  có bao nhiêu chữ số lẻ và bao nhiêu chữ số chẵn. Nếu không tồn tại số lẻ hoặc số chẵn thì in ra kết quả là 0 cho loại số tương ứng

Input	Output
12345678	4 4



**Bài 16. Số Strong.**

Viết chương trình cho phép nhập vào hai số nguyên dương và tìm tất cả các số Strong (là số có tổng giai thừa các chữ số bằng chính nó) nằm trong khoảng đó (nếu không tồn tại số nào thì in ra 0)

27

Input	Output
1 1000	1 2 145

**Bài 17. Số Lộc phát.**

Một số được gọi là “lộc phát” nếu chỉ có các chữ số 0,6,8. Nhập vào một số nguyên dương không quá 9 chữ số, hãy kiểm tra xem đó có phải số lộc phát hay không. Nếu đúng in ra 1, sai in ra 0.

Input	Output
6808	1
16808	0

**Bài 18. Thuận nghịch và lộc phát.**

Một số được coi là số đẹp nếu nó là số thuận nghịch, có chứa ít nhất một chữ số 6, và tổng các chữ số của nó có chữ số cuối cùng là 8. Viết chương trình liệt kê trong một đoạn giữa hai số nguyên cho trước có bao nhiêu số đẹp như vậy.

Input	Output
1 500	161

**Bài 19. Tính giai thừa.**

Viết chương trình C cho phép nhập một số tự nhiên n và tính giai thừa của n.

Input	Output
10	3628800

28

### Bài 20. Số armstrong

Số armstrong là số A có n chữ số và thỏa mãn tổng của lũy thừa bậc n của từng chữ số trong A bằng chính nó.

Ví dụ:  $371 = 3^3 + 7^3 + 1^3$

Viết chương trình C kiểm tra một số xem có phải là số armstrong hay không. Nếu đúng in ra 1, sai in ra 0.

Input	Output
371	1
24	0

### Bài 21. Thuận nghịch và không chứa số 9.

Viết chương trình C cho phép nhập vào số N, thực hiện liệt kê các số thuận nghịch lớn hơn 1 và nhỏ hơn N thỏa mãn không chứa chữ số 9. Có bao nhiêu số như vậy

Input	Output
100	2 3 4 5 6 7 8 11 22 33 44 55 66 77 88 15

### Bài 22. Chữ số cuối cùng lớn nhất

Viết chương trình C cho phép nhập vào n và liệt kê các số nguyên tố thỏa mãn nhỏ hơn n và có chữ số cuối cùng lớn nhất. Có bao nhiêu số như vậy

Input	Output
-------	--------

200	2 3 5 7 11 13 17 19 23 29 37 47 59 67 79 89 101 103 107 109 113 127 137 139 149 157 167 179 199 29
-----	---

29

### Bài 23. Nguyên tố cùng nhau.

Nhập 2 số nguyên dương a,b. Xác định xem 2 số vừa nhập có phải là 2 số nguyên tố cùng nhau.

#### Input

2 số nguyên dương a,b ( $1 \leq a, b \leq 10^{12}$ ).

#### Output

In YES nếu 2 số a,b nguyên tố cùng nhau, ngược lại in NO.

#### Ví dụ

Input	Output
20 17	YES
14 15	YES
8 128	NO

### Bài 24. Phi hàm Euler.

Đếm số lượng các số nguyên tố cùng nhau với n không vượt quá n.

#### Input

Số nguyên duy nhất n ( $1 \leq n \leq 10^{16}$ ).

#### Output

Kết quả của bài toán.

#### Ví dụ

Đếm số lượng các số nguyên tố cùng nhau với n không vượt quá n.

#### Input

Số nguyên duy nhất  $n$  ( $1 \leq n \leq 10^{16}$ ).

### Output

Kết quả của bài toán.

### Ví dụ

Input	Output
9	6

30

10000000000000000	4000000000000000
-------------------	------------------

### Bài 25. Thừa số nguyên tố thứ $k$

Đưa ra số nguyên tố thứ  $k$  trong phân tích thừa số nguyên tố của một số nguyên dương  $n$ .

Ví dụ  $n=28$ ,  $k=3$  ta có kết quả là 7 vì  $28=2 \times 2 \times 7$ .

### Input

2 số  $n, k$  ( $1 \leq n, k \leq 10^9$ ).

### Output

In ra số nguyên tố thứ  $k$  trong phân tích thừa số nguyên tố của  $n$ , trường hợp không tồn tại in -1.

### Ví dụ

Input	Output
28 3	7
8 5	-1
60 3	3

### Bài 26. Chữ số nguyên tố

Liệt kê số lần xuất hiện của chữ số nguyên tố của 1 số theo thứ tự từ nhỏ đến lớn.

**Input**

Số nguyên dương  $n$  ( $1 \leq n \leq 10^{18}$ ).

**Output**

Chữ số nguyên tố xuất hiện trong số ban đầu cùng với số lần xuất hiện của nó.

**Ví dụ**

Input	Output
722334123232277	2 6 3 4 7 3

31

**Bài 27.** Chữ số nguyên tố 2

Liệt kê số lần xuất hiện của chữ số nguyên tố của 1 số theo thứ tự xuất hiện các chữ số.

**Input**

Số nguyên dương  $n$  ( $1 \leq n \leq 10^{18}$ ).

**Output**

Chữ số nguyên tố xuất hiện trong số ban đầu cùng với số lần xuất hiện của nó theo thứ tự xuất hiện.

**Ví dụ**

Input	Output
722334123232277	7 3 2 6 3 4

**Bài 28.** Số nguyên dương nhỏ nhất.

Cho 4 số nguyên dương  $x, y, z, n$ .

Tìm số nguyên dương nhỏ nhất có  $n$  chữ số chia hết cho cả  $x, y$ , và  $z$ .

**Input**

4 số nguyên dương  $x, y, z, n$ . ( $1 \leq x, y, z \leq 10^4$ ).  $n \leq 16$ .

### Output

Kết quả của bài toán, trường hợp không tìm được số thỏa mãn in -1.

### Ví dụ

Input	Output
2 3 5 4	1020
3 5 7 2	-1

32

**Bài 29.** Tam giác Pascal.

In ra tam giác pascal với chiều cao là  $n$ .

### Input

Chiều cao của tam giác  $n$ . ( $1 \leq n \leq 10$ ).

### Output

Tam giác pascal tương ứng.

### Ví dụ

Input	Output
5	1 1 1 1 2 1 1 3 3 1 1 4 6 4 1

**Bài 30.** Ước chung lớn nhất, bội chung nhỏ nhất.

Tìm ước chung lớn nhất của 2 số nguyên không âm  $a$  và  $b$ .

### Input

2 số nguyên không âm  $a$  và  $b$  ( $0 \leq a, b \leq 10^9$ ).

### Output

In ra ước chung lớn nhất của 2 số và bội chung nhỏ nhất của 2 số.

### Ví dụ

Input	Output
100 20	20 100
17 29	1 493

33

### Bài 31. Phi hàm Euler 2.

Cho số nguyên dương  $n$ , nhiệm vụ của bạn là in ra  $\phi(i)$  với  $1 \leq i \leq n$ . Trong đó  $\phi(i)$  là phi hàm Euler của  $i$ .

### Input

Dòng đầu tiên là số lượng bộ test  $T$ . ( $1 \leq T \leq 100$ ).

Mỗi test case là một số nguyên dương  $n$  ( $1 \leq n \leq 10^6$ ).

### Output

In kết quả mỗi test case trên 1 dòng.

### Ví dụ

Input	Output
1 10	1 1 2 2 4 2 6 4 6 4

Link submit :

<https://www.spoj.com/problems/ETF/>

**Bài 32.** Lũy thừa nhị phân.

Tính  $a^b$  với  $a, b$  nguyên không âm.

**Input**

Dòng đầu tiên là số lượng bộ test  $T$ . ( $1 \leq T \leq 100$ ).

Mỗi test case là một số nguyên dương  $a, b$ .

**Output**

In kết quả mỗi test case trên 1 dòng.

34

Ví dụ

Input	Output
2	1024
2 10	27
3 3	

**Bài 33.** Đếm ước của  $n!$ 

Đếm số lượng ước của  $n!$ .

**Input**

Dòng đầu tiên là số lượng test case  $T$  ( $1 \leq T \leq 100$ ).

Mỗi test case là một số nguyên không âm  $n$  ( $1 \leq T \leq 100$ ).

**Output**

In ra kết quả mỗi test case trên 1 dòng.

Ví dụ

Input	Output
-------	--------



2	270
10	26494182162432000
97	

### Bài 34. LCM Sum.

Cho số nguyên dương  $n$ , tính tổng  $\text{lcm}(1,n) + \text{lcm}(2,n) + \dots + \text{lcm}(n,n)$ .

Trong đó  $\text{lcm}(a,b)$  là bội chung nhỏ nhất của  $a$  và  $b$ .

#### Input

Dòng đầu tiên là số lượng test case  $T$  ( $1 \leq T \leq 300000$ ).

35

Mỗi test case là một số nguyên dương  $n$  ( $1 \leq n \leq 1000000$ ).

#### Output

In kết quả mỗi test case trên 1 dòng.

#### Ví dụ

Input	Output
2	55
5	2933779482000000
1000000	

Link submit :

<https://www.spoj.com/problems/LCMSUM/>

### Bài 35. 486A.

Đối với số nguyên dương  $n$  hãy xác định hàm  $f$ :

$$f(n) = -1 + 2 - 3 + \dots + (-1)^n n$$

Nhiệm vụ của bạn là tính  $f(n)$  cho một số nguyên  $n$  đã cho.

### Input

Dòng đơn chứa số nguyên dương  $n$  ( $1 \leq n \leq 10^{15}$ ).

### Output

In  $f(n)$  trong một dòng duy nhất.

### Ví dụ

Input	Output
4	2
5	-3

Link submit:

<https://codeforces.com/problemset/problem/486/A>

36

### Bài 36. 1350A.

Orac đang nghiên cứu lý thuyết số, và ông quan tâm đến các tính chất của ước số. Đối với hai số nguyên dương  $a$  và  $b$ ,  $a$  là ước của  $b$  khi và chỉ khi tồn tại số nguyên  $c$ , sao cho  $a \cdot c = b$ .

Với  $n \geq 2$ , chúng ta sẽ biểu thị là  $f(n)$  ước số dương nhỏ nhất của  $n$ , ngoại trừ 1. Ví dụ:  $f(7) = 7$ ,  $f(10) = 2$ ,  $f(35) = 5$ .

Đối với số nguyên  $n$  cố định, Orac quyết định thêm  $f(n)$  vào  $n$ .

Ví dụ: nếu anh ta có số nguyên  $n = 5$ , giá trị mới của  $n$  sẽ bằng 10. Và nếu anh ta có số nguyên  $n = 6$ ,  $n$  sẽ được thay đổi thành 8.

Orac yêu nó rất nhiều, vì vậy anh quyết định lặp lại thao tác này nhiều lần.

Bây giờ, với hai số nguyên dương  $n$  và  $k$ , Orac đã yêu cầu bạn thêm  $f(n)$  vào  $n$  chính xác  $k$  lần (lưu ý rằng  $n$  sẽ thay đổi sau mỗi thao tác, vì vậy  $f(n)$  cũng có thể thay đổi) và cho anh ta biết giá trị cuối cùng của  $n$ .

Ví dụ: nếu Orac cho bạn  $n = 5$  và  $k = 2$ , lúc đầu, bạn nên thêm  $f(5) = 5$  thành  $n = 5$ , vì vậy giá trị mới của  $n$  sẽ bằng  $n = 10$ , sau đó, bạn nên thêm  $f(10) = 2$  đến 10, vì vậy giá trị mới (và cuối cùng!) của bạn sẽ bằng 12.

Orac có thể hỏi bạn những truy vấn này nhiều lần.

### Input

Dòng đầu tiên của đầu vào là một số nguyên  $t$  ( $1 \leq t \leq 100$ ): số lần mà Orac sẽ hỏi bạn.

Mỗi dòng  $t$  tiếp theo chứa hai số nguyên dương  $n, k$  ( $2 \leq n \leq 10^6, 1 \leq k \leq 10^9$ ), tương ứng với truy vấn của Orac.

Nó được đảm bảo rằng tổng số của  $n$  tối đa là  $10^6$ .

### Output

In các dòng  $t, i$ -th của chúng nên chứa giá trị cuối cùng của  $n$  trong truy vấn thứ  $i$  của Orac.

### Ví dụ

Input	Output
-------	--------

3	10
5 1	12
8 2	12
3 4	

37

Link submit:

<https://codeforces.com/problemset/problem/1350/A>

### Bài 37. 1238A : Prime subtraction.

Bạn được cung cấp hai số nguyên  $x$  và  $y$  (đảm bảo rằng  $x > y$ ). Bạn có thể chọn bất kỳ số nguyên tố  $p$  nào và trừ nó bất kỳ số lần nào từ  $x$ . Có thể làm  $x$  bằng  $y$ ? Hãy nhớ rằng một số nguyên tố là một số nguyên dương có chính xác hai ước số dương: 1 và chính nó. Chuỗi các số nguyên tố bắt đầu bằng 2, 3, 5, 7, 11. Chương trình của bạn nên giải quyết các trường hợp kiểm tra độc lập. **Input**

Dòng đầu tiên chứa một số nguyên  $t$  ( $1 \leq t \leq 1000$ ) - số lượng trường hợp kiểm tra. Sau đó  $t$  dòng tiếp theo, mỗi dòng mô tả một trường hợp thử nghiệm. Mỗi dòng chứa hai số nguyên  $x$  và  $y$  ( $1 \leq y < x \leq 10^8$ ).

### Output

Đối với mỗi trường hợp kiểm tra, hãy in YES nếu có thể chọn số nguyên tố  $p$  và trừ nó bất kỳ số lần nào từ  $x$  để  $x$  trở thành bằng  $y$ . Nếu không, in NO.

Bạn có thể in mọi chữ cái trong mọi trường hợp bạn muốn (ví dụ: các chuỗi yEs, yes, Yes, và YES đều sẽ được công nhận là câu trả lời hợp lệ).

### Ví dụ

Input	Output
4 100 98	

38

42 32	YES
1000000000000000000 1	YES
41 40	YES
	NO

Link Submit:

<https://codeforces.com/problemset/problem/1238/A>

### Bài 38. Số cắt đôi

Với một vài số nguyên dương có 1 chữ số, khi cắt đôi số đó theo chiều ngang và lấy nửa phía trên thì ta vẫn có một số nguyên. Cụ thể:

- Số 0 cắt đôi vẫn ra số 0
- Số 1 cắt đôi vẫn ra số 1
- Số 8 cắt đôi ra số 0
- Số 9 cắt đôi ra số 0
- Các số khác cắt đôi sẽ không hợp lệ.

Cho một số nguyên dương không quá 18 chữ số. Hãy in ra kết quả “cắt đôi” của số đó.

Nếu không hợp lệ thì ghi ra INVALID. **Chú ý:** nếu cắt đôi ra một dãy toàn 0 thì cũng được coi là không hợp lệ. Kết quả cắt đôi thì không tính chữ số 0 ở đầu.

### Input

Dòng đầu ghi số bộ test. Mỗi bộ test ghi một số nguyên dương không quá 18 chữ số.

### Output

Ghi ra kết quả tính toán

### Ví dụ

Input	Output
3	1000
1890	INVALID
3681	10

39

8919

### Bài 39. Mã hàng hóa

Trong mã hàng hóa người ta thường ghi kèm theo mã số quốc gia sản xuất. Nếu sản xuất tại Việt Nam thì mã tương ứng là 084. Bài toán đặt ra là cho một dãy mã dạng số nguyên không quá 18 chữ số. Hãy loại bỏ đoạn mã 084 ra khỏi mã ban đầu.

Dữ liệu đảm bảo dãy mã luôn có duy nhất một lần cụm 084.

### Input

Dòng đầu ghi số bộ test. Mỗi test là một số nguyên có ít nhất 4 chữ số nhưng không quá 18 chữ số.

### Output

Ghi ra kết quả sau khi loại bỏ 084

### Ví dụ

Input	Output
-------	--------

3	123567
123084567	3300478845
3300478808445	1
1084	

#### Bài 40. Số đẹp

Một số được coi là đẹp nếu chữ số đầu gấp đôi chữ số cuối hoặc ngược lại; đồng thời các chữ số từ vị trí thứ 2 đến gần cuối thỏa mãn là một số thuận nghịch.

Ví dụ: các số 36788766; 12345654322 là các số đẹp.

Viết chương trình kiểm tra số đẹp theo tiêu chí trên.

#### Input

- Dòng đầu ghi số bộ test
- Mỗi test là một số nguyên dương không quá 18 chữ số

#### Output

- Ghi ra YES tương ứng với số đẹp, NO trong trường hợp ngược lại Ví

40

dụ

Input	Output
3	YES
36788766	YES
22345654321	NO
12345654321	

### Phần 4. Phép toán Modulo

#### Bài 1. Pow mod

Cho ba số nguyên dương  $x$ ,  $y$ ,  $p$ . Nhiệm vụ của bạn là tính  $(x^y) \% p$ . Ví dụ với  $x = 2$ ,  $y = 3$ ,  $p = 5$  thì  $(2^3) \% 5 = 3$ .

**Input:**

Dòng đầu tiên đưa vào số lượng test T.

Những dòng kế tiếp mỗi dòng đưa vào một test. Mỗi test là bộ ba x, y, p được viết cách nhau một vài khoảng trống.

T, x, y, p thỏa mãn ràng buộc :  $1 \leq T \leq 100$ ;  $1 \leq x, y \leq 10^6$ ;  $1 \leq p \leq 10^9 + 7$ .

**Output:**

Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ**

Input: Output:	
2	3
2 3 5	1
3 2 4	

41

**Bài 2. Tổng modulo 1.**

Cho hai số nguyên không âm N và K. Nhiệm vụ của bạn là tìm  $S = 1\%K + 2\%K + \dots + N\%K$ . Ví dụ với N = 10, K=2 ta có  $S = 1\%2 + 2\%2 + 3\%2 + 4\%2 + 5\%2 + 6\%2 + 7\%2 + 8\%2 + 9\%2 + 10\%2 = 5$ .

**Input:**

Dòng đầu tiên đưa vào số lượng test T.

Những dòng kế tiếp mỗi dòng đưa vào một test. Mỗi test là bộ đôi N, K được viết cách nhau một vài khoảng trống.

T, N, K thỏa mãn ràng buộc :  $1 \leq T \leq 100$ ;  $0 \leq N \leq 1000$ ;  $0 \leq K \leq 10^{12}$ .

**Output:**

Đưa ra kết quả theo mỗi test.

Đưa ra kết quả mỗi test theo từng dòng.

Input: Output:
----------------

2 10 55 1 11	55 1
--------------------	---------

### Bài 3. Tổng modulo2

Cho hai số nguyên không âm  $N$  và  $K$ . Nhiệm vụ của bạn là kiểm tra xem  $K = 1\%K + 2\%K + \dots + N\%K$  hay không. Đưa ra 1 hoặc 0 nếu cặp  $N, K$  thỏa mãn hoặc không thỏa mãn yêu cầu bài toán. Ví dụ với  $N = 10, K=55$  ta có kết quả là 1 vì  $55 = 1\%55 + 2\%55 + 3\%55 + \dots + 10\%55$ . Ngược lại,  $N=4, K=11$  có kết quả là 0 vì  $11 \neq 1\%11 + 2\%11 + 3\%11 + 4\%11$ .

Input:

Dòng đầu tiên đưa vào số lượng test  $T$ .

Những dòng kế tiếp mỗi dòng đưa vào một test. Mỗi test là bộ đôi  $N, K$  được viết cách nhau một vài khoảng trống.

$T, N, K$  thỏa mãn ràng buộc :  $1 \leq T \leq 100; 0 \leq N \leq 1000; 0 \leq K \leq 10^{12}$ .

Output:

42

Đưa ra kết quả mỗi test theo từng dòng.

Input: Output:	
2 10 55 1 11	1 0

### Bài 4. Fibonacci modulo.

Dãy số Fibonacci được định nghĩa  $F_n = F_{n-1} + F_{n-2}$ ,  $n > 1$  và  $F_0 = 0, F_1 = 1$ . Dưới đây là một số Fibonacci : 0, 1, 1, 2, 3, 5, 8, 13, 21...

Nhiệm vụ của bạn là tìm số Fibonacci thứ  $n$ .

Input:

Dòng đầu tiên đưa vào số lượng bộ test  $T$ .

Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số nguyên dương  $n, T$ ,

$n$  thỏa mãn ràng buộc :  $1 \leq T \leq 100; 1 \leq n \leq 1000$ .



**Output:**

Đưa ra kết quả mỗi test theo modulo  $10^9 + 7$  theo từng dòng.

**Ví dụ**

Input	Output
2	1
2	5
5	

**Bài 5.** Giá trị của đa thức

Tính toán giá trị đa thức  $P(n, x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_0$ . Kết quả lấy dư với  $m$ . Trong đó,  $m = 10^9 + 7$ .

**Input:**

Dòng đầu tiên đưa vào số lượng test  $T$ .

43

Những dòng kế tiếp đưa vào các bộ test. Mỗi test gồm hai dòng: dòng thứ nhất đưa vào hai số  $n, x$ ; dòng tiếp theo đưa vào  $n$  số  $a_0, a_1, a_2, \dots, a_{n-2}, a_{n-1}$  là hệ số của đa thức  $P$ . Các số được viết cách nhau một vài khoảng trống.

$T, n, x, P[i]$  thỏa mãn ràng buộc :  $1 \leq T \leq 100; 0 \leq n \leq 2000; 0 \leq x, P[i]$

$\leq 1000$ . **Output:**

Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ**

Input:	Output:
1	20
4 2	
4 0 2 1	

## Bài 6. 1177A

Bạn được cho hai số nguyên dương  $a$  và  $b$ . Trong một lần di chuyển, bạn có thể tăng  $a$  thêm 1 (thay thế bằng  $+ 1$ ). Nhiệm vụ của bạn là tìm ra số lần di chuyển tối thiểu bạn cần thực hiện để chia hết cho  $b$ . Có thể, bạn phải thực hiện 0 di chuyển, vì  $a$  đã chia hết cho  $b$ . Bạn phải trả lời  $t$  trường hợp kiểm tra độc lập.

### Input

Dòng đầu tiên của đầu vào chứa một số nguyên  $t$  ( $1 \leq t \leq 10^4$ ) - số lượng trường hợp kiểm tra. Sau đó  $t$  kiểm tra trường hợp theo.

Dòng duy nhất của trường hợp thử nghiệm chứa hai số nguyên  $a$  và  $b$  ( $1 \leq a$ ,

$b \leq 10^9$ ). **Output**

Đối với mỗi trường hợp kiểm tra, hãy in câu trả lời - số lần di chuyển tối thiểu bạn cần thực hiện để chia hết cho  $b$ .

### Ví dụ

44

Input	Output
5	2
10 4	5
13 9	4
100 13	333
123 456	0
92 46	

## Bài 7. Số nguyên dương thứ $k$ không chia hết cho $n$

Bạn được cho hai số nguyên dương  $n$  và  $k$ . In số nguyên dương thứ  $k$  không chia hết cho  $n$ .

Ví dụ: nếu  $n = 3$  và  $k = 7$ , thì tất cả các số không chia hết cho 3 là: 1,2,4,5,7,8,10,11,13. Số thứ 7 trong số đó là 10.

### Input

Dòng đầu tiên chứa số nguyên  $t$  ( $1 \leq t \leq 1000$ ) - số lượng trường hợp kiểm tra trong đầu vào.

Tiếp theo, t trường hợp thử nghiệm được đưa ra, một trường hợp trên mỗi dòng.

Mỗi trường hợp thử nghiệm là hai số nguyên dương  $n$  ( $2 \leq n \leq 10^9$ ) và  $k$  ( $1 \leq k \leq 10^9$ ).

**Output** Đối với mỗi trường hợp thử nghiệm, in số nguyên dương thứ  $k$  không chia hết cho  $n$ .

**Ví dụ**

Input	Output
6	10
37	15
4 12	1999999999
2 1000000000	113
7 97	1000000001
1000000000 1000000000 21	1

**Bài 8. Last digit**

Cho  $n$ , in ra chữ số cuối cùng của  $1378^n$ .

**Input**

45

Dòng đầu vào chứa một số nguyên  $n$  ( $0 \leq n \leq 10^9$ ).

**Output**

In số nguyên đơn - chữ số cuối cùng của  $1378^n$ .

**Ví dụ**

Input	Output
2	4

**Bài 9. Mod 5**

Fedya học trong một phòng tập thể dục. Quê hương toán học của Fedya là để tính biểu thức sau:

$$(1^n + 2^n + 3^n + 4^n) \bmod 5$$

cho giá trị đã cho của n. Fedya quản lý để hoàn thành nhiệm vụ. Bạn có thể? Lưu ý rằng số n đã cho có thể cực kỳ lớn (ví dụ: nó có thể vượt quá mọi loại số nguyên của ngôn ngữ lập trình của bạn).

### Input

Dòng đơn chứa một số nguyên n ( $0 \leq n \leq 10^{10^5}$ ). Số này không chứa bất kỳ số 0 hàng đầu nào.

### Output

In giá trị của biểu thức mà không có số 0 đứng đầu.

### Ví dụ

Input	Output
4	4
124356983594583453458888889	0

## Bài 10. Last digit2

46

### Input

1 dòng duy nhất gồm cơ số và số mũ và cơ số a, b ( $0 \leq a, b \leq 10^9$ ).

### Output

In ra chữ số cuối cùng của  $a^b$

### Ví dụ

Input	Output
2 10	4
5 1000000000	5

Link Submit : <https://www.spoj.com/problems/LASTDIG/>

### Bài 11. Last digit 3.

Tìm chữ số tận cùng của  $a^n$ . Trong đó  $a, N$  có giá trị từ 1 trở lên và nhỏ hơn

1010.000.000

#### Input

Dòng đầu tiên là  $a$

Dòng thứ 2 là  $n$ .

#### Output

Số tận cùng của  $a^n$ . Ví dụ

Input	Output
2 5	2
2 10	4

### Bài 12. Number of binary string

47

Your task is to calculate the number of bit strings of length  $n$ .

For example, if  $n=3$ , the correct answer is 8, because the possible bit strings are 000, 001, 010, 011, 100, 101, 110, and 111.

#### Input

The only input line has an integer  $n$ .

#### Output

Print the result modulo  $10^9+7$ .

## Constraints

$1 \leq n \leq 10^6$ .

## Example

Input:

3

Output:

8

Link submit : <https://cses.fi/problemset/task/1617>