

Bài 1. Song nguyên tố**Tên file: CPRIME.CPP**

Số nguyên tố là một số nguyên dương có 2 ước dương là 1 và chính nó.

Ví dụ: 7, 13, 17, ... là những số nguyên tố; còn các số 14, 8, 25, ... không phải là những số nguyên tố.

Số đảo ngược của một số là số được viết theo thứ tự ngược lại của số đó.

Ví dụ: 13 đảo ngược của nó là 31; 145 đảo ngược của nó là 541.

Một số n được gọi là song nguyên tố nếu như n là số nguyên tố và số đảo ngược của nó cũng là số nguyên tố.

Ví dụ: 7, 13 là các số song nguyên tố còn các số 8, 41 không phải là các số song nguyên tố.

Cho một số nguyên dương n . Em hãy kiểm tra xem số n có phải là song nguyên tố không?

INPUT

- Một dòng duy nhất chứa số nguyên dương n ($n \leq 2 \cdot 10^9$)

OUTPUT

- In ra số **1** nếu đó là số song nguyên tố, và in ra số **0** nếu đó không phải là số song nguyên tố.

Ví dụ:

Ví dụ 1		Ví dụ 2	
CPRIME.INP	CPRIME.OUT	CPRIME.INP	CPRIME.OUT
7	1	8	0

Bài 2. Đếm số 0 bên phải**Tên file: BZERO.CPP**

Cho một số nguyên n . Hãy đếm xem trong kết quả của số $n!$ (n giai thừa) có bao nhiêu chữ số 0 liên tiếp tính từ hàng đơn vị (hay bao nhiêu số 0 liên tiếp bên phải).

INPUT: Là số nguyên n ($1 \leq n \leq 1.000$)

OUTPUT: Là số lượng chữ số 0 liên tiếp tính từ hàng đơn vị của $n!$.

Ví dụ:

BZERO	BZERO
4	0
8	1
20	4

Bài 3. Số siêu nguyên tốTên file: **SPRIME.CPP**

Một số tự nhiên N được gọi là siêu nguyên tố nếu bản thân nó là một số nguyên tố và tất cả các số thu được bằng cách xóa lần lượt các chữ số bên phải của nó đều là số nguyên tố.

Ví dụ:

Số 317 là một số siêu nguyên tố vì: 317 là 1 số nguyên tố

Xóa 1 chữ số bên phải: 31 là 1 số nguyên tố

Xóa 2 chữ số bên phải: 3 là 1 số nguyên tố

Cho 2 số nguyên a, b . Hãy liệt kê tất cả các số siêu nguyên tố thuộc đoạn $[a, b]$.

Input: Tập SPRIME.INP gồm một dòng ghi 2 số nguyên dương a, b ($0 < a, b < 10^7$)

Output: Tập SPRIME.OUT liệt kê theo thứ tự tăng các số siêu nguyên tố thuộc đoạn $[a, b]$, mỗi số trên một dòng, hoặc ghi “NO” trong trường hợp không có số nào thuộc đoạn đó.

Ví dụ

SPRIME.INP	SPRIME.OUT
3 57	3
	5
	7
	23
	29
	31
	37
	53

Bài 4. Biểu diễn $N!$ Tên file: **analyse.py hoặc analyse.cpp**

Giai thừa N , ký hiệu $N!$ là tích tất cả các số nguyên từ 1 đến N . Giai thừa N tăng rất nhanh, ví dụ $5! = 120$, $10! = 3628800$. Một cách để xác định số lớn như vậy, người ta chỉ ra số lần xuất hiện các số nguyên tố trong phân tích của nó ra thừa số nguyên tố. Ví dụ, số 825 có thể xác định bởi dãy (0 1 2 0 1) có nghĩa là $825 = 2^0 \cdot 3^1 \cdot 5^2 \cdot 7^0 \cdot 11^1$.

Cho một số nguyên dương $N \leq 1000$. Hãy tìm cách biểu diễn số $N!$ dưới dạng số lần xuất hiện các số nguyên tố trong phân tích $n!$ ra các thừa số nguyên tố.

Dữ liệu vào từ file ANALYSE.INP gồm nhiều dòng, mỗi dòng chứa 1 số nguyên N ($2 \leq N \leq 1000$).

Kết quả Ghi ra file ANALYSE.OUT mỗi dòng tương ứng với 1 dòng trong file dữ liệu vào là dãy số thể hiện biểu diễn dưới dạng phân tích thành số nguyên tố của $n!$ (phần tử cuối cùng của dãy phải là số dương).

Ví dụ:

ANALYSE.INP	ANALYSE.OUT
10	8 4 2 1

Bài 5. Ước số

Tên file: uocso.cpp hoặc uocso.py

Cho hai số nguyên dương a và b . Hãy đếm số lượng các ước của tích $a * b$

Input: uocso.inp

- Dòng 1 chứa số nguyên dương T ($T \leq 100$)
- T dòng tiếp theo, mỗi dòng chứa một cặp số a, b ($1 \leq a, b \leq 10^9$)

Output: uocso.out

- Gồm T dòng, mỗi dòng chứa kết quả của tích tương ứng của input

Ví dụ:

uocso.inp	uocso.out
2	4
2 4	8
6 7	

Ràng buộc:

- Subtask1: có 60% số test có $T = 1$
- Subtask2: có 40% số test còn lại không có ràng buộc gì