

THỬA SỐ NGUYÊN TỐ

1. **Prime Facstors.** Cho số nguyên dương N . Hãy đưa ra tất cả các ước số nguyên tố của N .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một số nguyên dương N được ghi trên một dòng.
- T, N thỏa mãn ràng buộc: $1 \leq T \leq 100$; $2 \leq N \leq 10^{10}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	3 3 5 7
315	31
31	

2. **Max Prime Facstors.** Cho số nguyên dương N . Hãy đưa ra ước số nguyên tố lớn nhất của N .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một số nguyên dương N được ghi trên một dòng.
- T, N thỏa mãn ràng buộc: $1 \leq T \leq 100$; $2 \leq N \leq 10^{10}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	7
315	31
31	

3. **Prime Eratosthenes.** Cho số nguyên dương N . Hãy đưa ra tất cả các số nguyên tố nhỏ hơn hoặc bằng N .

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một số nguyên dương N được ghi trên một dòng.
- T, N thỏa mãn ràng buộc: $1 \leq T \leq 100$; $2 \leq N \leq 10^4$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2 3 5 7
10	2 3 5 7 11 13 17 19 23 29 31
35	

4. **Least Prime Factor.** 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: $1 \leq T \leq 100$; $1 \leq N \leq 10000$.

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	

5. **Prime in Range.** Hãy sinh ra tất cả các số nguyên tố trong khoảng $[M, N]$. Ví dụ $M=1, N=10$ ta có kết quả 2 3 5 7.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp mỗi dòng đưa vào một bộ test. Mỗi bộ test là bộ đôi M, N được viết cách nhau một vài khoảng trống.
- T, M, N thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq M \leq N \leq 10000$; $N-M \leq 10000$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

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

6. **Pair Primes.** Cho số nguyên dương chẵn $N > 2$. Hãy đưa ra cặp số nguyên tố p, q đầu tiên có tổng đúng bằng N. Ví dụ $N = 6$ ta có cặp số nguyên tố đầu tiên 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.

Input:	Output:
2	2 71
74	3 1021
1024	

7. **Sphenic Number.** 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?

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.

Input:	Output:
2	1
30	0
60	

8. 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.

Input:	Output:
3	9
62 7	73
76 2	0
3 5	

9. Cho số tự nhiên N. Nhiệm vụ của bạn là hãy đưa ra tất cả các ước số nguyên tố của N cùng lũy thừa của nó. Ví dụ $N=100 = 2^2 \times 5^2$. $N = 35 = 5^1 \times 7^1$.

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 $1 \leq T \leq 100$; $1 \leq N \leq 10000$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2 2 5 2
100	5 1 7 1
35	

10. **Smith Number**. 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	No
666	

11. **Perfect Number.** Cho số tự nhiên N . Nhiệm vụ của bạn là hãy kiểm tra N có phải là số hoàn hảo hay không. Một số N được gọi là số hoàn hảo nếu tổng các ước số của nó bằng chính nó. Ví dụ $N = 6 = 1 + 2 + 3$ là số hoàn hảo.

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 $1 \leq T \leq 100; 1 \leq N \leq 10^{18}$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
6	0
21	

12. **Kth Prime Divisors.** Cho số tự nhiên N . Nhiệm vụ của bạn là hãy đưa ra ước số nguyên tố thứ k của N . Đưa ra -1 nếu không tồn tại ước số thứ k của N . Ví dụ $N = 255, k = 2$ ta có kết quả là 3 vì $255 = 3 \times 3 \times 5 \times 5$. Với $N = 81, k = 5$ ta có kết quả -1 vì $81 = 3 \times 3 \times 3 \times 3$.

Input:

- Dòng đầu tiên đưa vào số lượng test T .
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một bộ đôi N và k .
- T, N thỏa mãn ràng buộc $1 \leq T \leq 100; 1 \leq N, k \leq 10^4$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	3
25 5	-1
81 5	

13. **Số các số có ước số lẻ.** Cho số tự nhiên N . Nhiệm vụ của bạn là hãy đếm số các số có ước số lẻ nhỏ hơn hoặc bằng N .

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 $1 \leq T \leq 100; 1 \leq N \leq 10^9$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1
1	2
5	

14. **Number 3-Divisors.** 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.

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

15. **Smallest Divisors.** Cho số tự nhiên N . Nhiệm vụ của bạn là hãy đưa ra ước số nguyên tố nhỏ nhất của các số từ 1 đến N . Ví dụ $n=10$, ta có được kết quả : 1 2 3 2 5 2 7 2 3 2 .

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.

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

16. **Number 3-Divisors.** 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.

Input:	Output:
2	4
50	6
200	

17. **Number 3-Divisors.** 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.

Input:	Output:
2	4
1 10	78498
1 1000000000000	

18. **Number 3-Divisors.** Cho bốn số M, N, A, B. Nhiệm vụ của bạn là hãy đếm tất cả các số trong khoảng [M, N] chia hết cho các số A, B.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm 4 số M, N, A, B.
- T, M, N, A, B thỏa mãn ràng buộc $1 \leq T \leq 100$; $1 \leq M, N \leq 10^5$; $1 \leq A, B \leq 500$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	2
5 11 4 6	289
3 1000 5 9	