

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. **Leat 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. Prime 7.

11. Prime 8.

12. Prime 9.

13. Prime 10.

B1. Prime 1. <https://www.geeksforgeeks.org/print-all-prime-factors-of-a-given-number/>

B3. <https://www.geeksforgeeks.org/sieve-of-eratosthenes/>

B4.

B5. <https://practice.geeksforgeeks.org/problems/find-prime-numbers-in-a-range/0/?ref=self>

B6. <https://practice.geeksforgeeks.org/problems/return-two-prime-numbers/0/?ref=self>

<https://www.geeksforgeeks.org/smith-number/>

<https://www.geeksforgeeks.org/sphenic-number/>

<https://www.geeksforgeeks.org/k-th-prime-factor-given-number/>

<https://www.geeksforgeeks.org/rsa-algorithm-cryptography/>

Mark 8. <https://practice.geeksforgeeks.org/problems/generalised-fibonacci-numbers/0>

<https://practice.geeksforgeeks.org/problems/total-decoding-messages/0>

<https://www.geeksforgeeks.org/count-ways-reach-nth-stair/>

<https://practice.geeksforgeeks.org/problems/fibonacci-sum/0>

<https://www.geeksforgeeks.org/nth-non-fibonacci-number/>

<https://practice.geeksforgeeks.org/problems/nth-even-fibonacci-number/0>

<https://practice.geeksforgeeks.org/problems/even-fibonacci-numbers-sum/0>

Mark 8. <https://practice.geeksforgeeks.org/problems/return-two-prime-numbers/0/?ref=self>

<https://www.geeksforgeeks.org/print-all-prime-factors-and-their-powers/>

<https://www.geeksforgeeks.org/no-factors-n/>