

[Hàm bổ sung]. Bài 1. Số ước nguyên tố

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Bạn hãy đếm xem số N có bao nhiêu ước là số nguyên tố khác nhau. Gọi ý : Hãy đếm N có bao nhiêu thừa số nguyên tố khác nhau.

Input Format

Dòng duy nhất chứa số nguyên dương N

Constraints

$$1 \leq N \leq 10^{12}$$

Output Format

In ra số lượng ước là số nguyên tố của N

Sample Input 0

96

Sample Output 0

2

[Hàm bổ sung]. Bài 2. Thừa số nguyên tố số mũ lớn nhất

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Bạn hãy tìm thừa số nguyên tố có số mũ lớn nhất của số nguyên N , nếu có nhiều thừa số nguyên tố có cùng số mũ lớn nhất hãy chọn số nhỏ hơn. Ví dụ $N = 60 = 2^2 * 3^1 * 5^1$ thì đáp án sẽ là 2.

Input Format

- Dòng duy nhất chứa số nguyên dương N

Constraints

$$2 \leq N \leq 10^{12}$$

Output Format

In ra thừa số nguyên tố kèm với số mũ

Sample Input 0

88

Sample Output 0

2 3

[Hàm bổ sung]. Bài 3. Phân tích thừa số nguyên tố

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho số nguyên dương N, bạn hãy phân tích thừa số nguyên tố theo các cách sau.

- Cách 1 : Chỉ in ra các thừa số nguyên tố khác nhau của N mỗi thừa số 1 lần
- Cách 2 : In ra các thừa số nguyên tố kèm theo số mũ
- Cách 3 : In ra các thừa số nguyên tố

Xem output mẫu để rõ hơn

Mã nguồn tham khảo :

```
#include <bits/stdc++.h>
using namespace std;

using ll = long long;
//n = 28 : 2 2 7
```

```

void pt1(int n){
    for(int i = 2; i <= sqrt(n); i++){
        //Neu ma n chia het cho i
        if(n % i == 0){
            cout << i << " ";
            while(n % i == 0){
                n /= i;
            }
        }
    }
    if(n > 1){
        cout << n;
    }
}

void pt2(int n){
    for(int i = 2; i <= sqrt(n); i++){
        //Neu ma n chia het cho i
        if(n % i == 0){
            // dem so mu cua i
            int mu = 0;
            while(n % i == 0){
                n /= i;
                ++mu;
            }
            cout << "(" << i << ", " << mu << ") ";
        }
    }
    if(n > 1){
        cout << "(" << n << ", 1)";
    }
}

void pt3(int n){
    for(int i = 2; i <= sqrt(n); i++){
        //Neu ma n chia het cho i
        if(n % i == 0){
            while(n % i == 0){
                cout << i;
                n /= i;
                if(n > 1){
                    cout << " x ";
                }
            }
        }
    }
    if(n > 1){
        cout << n << endl;
    }
}

int main(){
    int n; cin >> n;

```

```
pt1(n);  
cout << endl;  
pt2(n);  
cout << endl;  
pt3(n);  
}
```

Input Format

- Dòng duy nhất chứa số nguyên dương N

Constraints

$$2 \leq N \leq 10^6$$

Output Format

In ra 3 dòng theo yêu cầu

Sample Input 0

120

Sample Output 0

2 3 5
(2, 3) (3, 1) (5, 1)
2 x 2 x 2 x 3 x 5

[Hàm bổ sung]. Bài 4. GCD

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho 4 số a, b, c, d. Hãy tìm ước chung lớn nhất và bội chung nhỏ nhất của 4 số trên. Gợi ý $\gcd(\gcd(a, b), \gcd(c, d))$, $\text{lcm}(\text{lcm}(a, b), \text{lcm}(c, d))$.

Input Format

4 số nguyên dương a, b, c, d

Constraints

$$1 \leq a, b, c, d \leq 100$$

Output Format

In ra UCLN và BCNN của 4 số

Sample Input 0

```
76 81 1 53
```

Sample Output 0

```
1 326268
```

[Hàm bổ sung]. Bài 5. Tam giác pascal bản dễ

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho chiều cao của tam giác pascal, bạn hãy in ra tam giác pascal tương ứng. Ví dụ tam giác pascal có chiều cao là 4 :

```
1
1 1
1 2 1
1 3 3 1
```

Các bạn có thể tính toán các phần tử trên tam giác pascal bằng tổ hợp. Ví dụ tam giác pascal có chiều cao là 4 :

```
C(0, 0)
C(0, 1) C(1, 1)
C(0, 2) C(1, 2) C(2, 2)
C(0, 3) C(1, 3) C(2, 3) C(3, 3)
```

Trong đó $C(i, j)$ là tổ hợp chập j của i

Input Format

Chiều cao của tam giác

Constraints

Chiều cao không quá 12.

Output Format

In ra tam giác pascal

Sample Input 0

```
6
```

Sample Output 0

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```

[Hàm bổ sung]. Bài 6. Tổng chia dư

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho N số nguyên, bạn hãy tính tổng các số này và chia dư tổng cho $10^9 + 7$ (1000000007).

Input Format

- Dòng 1 là N : số lượng số nguyên
- Dòng 2 gồm N số nguyên cách nhau 1 khoảng trắng

Constraints

- $1 \leq N \leq 10^5$
- Các số là nguyên dương không quá 10^{16}

Output Format

In ra đáp án của bài toán

Sample Input 0

```
5
534 7 669 826 610
```

Sample Output 0

```
2646
```

[Hàm bổ sung]. Bài 7. Tích chia dư

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho N số nguyên, bạn hãy tính tích các số này và chia dư tổng cho $10^9 + 7$ (1000000007).

Input Format

- Dòng 1 là N : số lượng số nguyên
- Dòng 2 gồm N số nguyên cách nhau 1 khoảng trắng

Constraints

- $1 \leq N \leq 10^5$
- Các số là nguyên dương không quá 10^6

Output Format

In ra đáp án của bài toán

Sample Input 0

```
5
153 747 236 481 789
```

Sample Output 0

```
373224432
```

[Hàm bổ sung]. Bài 8. Giai thừa

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Hãy tính giai thừa các số từ 1 tới N và chia dư cho $10^9 + 7$

Input Format

Dòng duy nhất chứa số nguyên dương N

Constraints

$$1 \leq N \leq 10^6$$

Output Format

In ra N dòng tương ứng với giai thừa các số từ 1 tới N sau khi chia dư

Sample Input 0

5

Sample Output 0

1
2
6
24
120

[Hàm bổ sung]. Bài 9. Chữ số cuối cùng

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Tìm K chữ số cuối cùng của N^M , kết quả có thể không đủ K chữ số. Ví dụ $9^5 = 59049$ và $K = 3$ thì bạn chỉ cần in ra 49. Gợi ý : Tính N^M chia dư cho 10^K

Input Format

3 số N, M, K

Constraints

- $1 \leq N, M \leq 10^6$
- $1 \leq K \leq 9$

Output Format

In ra đáp án của bài toán

Sample Input 0

9 8 1

Sample Output 0

1

[Hàm bổ sung]. Bài 10. Dãy số 23

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Dãy số 23 được định nghĩa như sau : $F(1) = 1$, $F(2) = 1$, $F_n = 2 * F(n - 1) + 3 * F(n - 2)$. Một số đầu tiên trong dãy 28 : 1, 1, 5, 13, 41.... Hãy tính số thứ n trong dãy số 23 chia dư cho $10^9 + 7$.

Input Format

Dòng duy nhất chứa số nguyên dương n

Constraints

$1 \leq n \leq 10^6$

Output Format

In ra đáp án của bài toán

Sample Input 0

6

Sample Output 0

121

[Hàm, Lý Thuyết Số]. Fibonacci 1

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Nhiệm vụ của bạn là in ra N số Fibonacci đầu tiên, một số đầu tiên trong dãy Fibonacci là 1, 1, 2, 3, 5, 8....

Input Format

Dòng duy nhất chứa số nguyên dương N

Constraints

$1 \leq N \leq 92$

Output Format

In ra N số Fibonacci đầu tiên, mỗi số trên 1 dòng

Sample Input 0

6

Sample Output 0

1
1
2
3
5
8

[Hàm, Lý Thuyết Số]. Bài 64. Số Fibonacci 2

- [Problem](#)

- [Submissions](#)
- [Discussions](#)

Nhập vào một số và kiểm tra xem số vừa nhập có phải là số trong dãy fibonacci hay không? Biết rằng số fibonacci bắt đầu bằng 0 và 1.

Input Format

Số nguyên không âm n

Constraints

$$0 \leq n \leq 9 \cdot 10^{18}$$

Output Format

In ra YES nếu n là số Fibonacci, ngược lại in NO

Sample Input 0

0

Sample Output 0

YES

Sample Input 1

18636

Sample Output 1

NO

[Hàm, Lý Thuyết Số]. Fibonacci 3

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Nhiệm vụ của bạn là hãy tìm số thuộc dãy số Fibonacci nhỏ nhất lớn hơn hoặc bằng số N đã cho. Biết một số đầu tiên trong dãy Fibonacci là : 1, 1, 2, 3, 5, 8, 13....

Input Format

Dòng duy nhất chứa số nguyên dương N

Constraints

$$1 \leq N \leq 10^{18}$$

Output Format

In ra số Fibonacci nhỏ nhất lớn hơn hoặc bằng N

Sample Input 0

12

Sample Output 0

13

[Hàm, Lý Thuyết Số]. Fibonacci 4

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Liệt kê những số là số nguyên tố nhỏ hơn N và có tổng các chữ số của nó là một số trong dãy số Fibonacci.

Input Format

Dòng duy nhất chứa số nguyên dương N

Constraints

$$1 \leq N \leq 10000$$

Output Format

In ra các số nhỏ hơn N là số nguyên tố và thỏa mãn tổng chữ số của nó là một số trong dãy Fibonacci. Các số in cách nhau một khoảng trắng

Sample Input 0

114

Sample Output 0

```
2 3 5 11 17 23 41 53 67 71 101 107 113
```

[Hàm, Lý Thuyết Số]. Fibonacci 5

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Kiểm tra một số có phải là số Fibonacci hay không, bạn phải trả lời nhiều trường hợp trong bài toán này.

Input Format

Dòng đầu tiên là số lượng bộ test T; T dòng tiếp theo mỗi dòng là một số nguyên dương N;

Constraints

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

Output Format

Đối với mỗi test case in kết quả trên một dòng, nếu là số Fibonacci in YES, ngược lại in NO.

Sample Input 0

```
5
89
754
399
34
661
```

Sample Output 0

```
YES
NO
NO
YES
NO
```