

## **[Hàm, Lý Thuyết Số]. Bài 1. Luyện tập viết hàm (Quan trọng)**

Cho số nguyên  $n$  không âm. Viết hàm xử lý các yêu cầu sau

- 1. Kiểm tra  $n$  là số nguyên tố, nếu đúng in 1, sai in 0.
- 1. In tổng chữ số của  $n$ .
- 1. In tổng chữ số chẵn của  $n$ .
- 1. In tổng chữ số của  $n$  là số nguyên tố.
- 1. In số lật ngược của  $n$ . Ví dụ 123 in 321.
- 1. In số lượng ước của  $n$  là số nguyên tố (làm tương tự như phân tích thừa số nguyên tố).
- 1. In ước nguyên tố lớn nhất của  $n$  (làm tương tự như phân tích thừa số nguyên tố).
- 1. Kiểm tra nếu  $n$  tồn tại ít nhất 1 số 6, nếu đúng in 1, sai in 0.
- 1. Kiểm tra nếu tổng chữ số của  $n$  chia hết cho 8, nếu đúng in 1, sai in 0.
-

1. Tính tổng giai thừa các chữ số của n và in ra. Ví dụ n = 123, tổng = 1! + 2! + 3!

•

1. Kiểm tra n có phải chỉ được tạo bởi 1 số hay không? Ví dụ 222, 333, 99999. Đúng in ra 1, sai in ra 0.

•

1. Kiểm tra n có phải có chữ số tận cùng là lớn nhất hay không, tức là không có chữ số nào của n lớn hơn chữ số hàng đơn vị của nó. nếu đúng in 1, sai in 0.

•

1. In tổng lũy thừa chữ số của n với số mũ là số chữ số. ví dụ 123 thì tính  $1^3 + 2^3 + 3^3$ .

## Input Format

Số duy nhất n

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

using ll = long long;

bool ham1(int n){
    if(n < 2) return false;
    for(int i = 2; i <= sqrt(n); i++){
        if(n % i == 0){
            return false;
        }
    }
    return true;
}

int ham2(int n){
    int sum = 0;
    while(n != 0){
        sum += n % 10;
        n /= 10;
    }
}
```

```

    }
    return sum;
}

int ham3(int n){
    int sum = 0;
    while(n != 0){
        if(n % 10 % 2 == 0){
            sum += n % 10;
        }
        n /= 10;
    }
    return sum;
}

int ham4(int n){
    int sum = 0;
    while(n != 0){
        int r = n % 10;
        if(r == 2 || r == 3 || r == 5 || r == 7)
            sum += r;
        n /= 10;
    }
    return sum;
}

int ham5(int n){
    int rev = 0;
    while(n){
        rev = rev * 10 + n % 10;
        n /= 10;
    }
    return rev;
}

int ham6(int n){
    int dem = 0;
    for(int i = 2; i <= sqrt(n); i++){
        if(n % i == 0){
            ++dem;
        }
    }
}

```

```

        while(n % i == 0){
            n /= i;
        }
    }
}
if(n > 1)
    ++dem;
return dem;
}
//60 = 2 2 3 5
//28 = 2 2 7
//13 = 13
//100 = 2 2 5 5

int ham7(int n){
    int ans;
    for(int i = 2; i <= sqrt(n); i++){
        if(n % i == 0){
            ans = i;
            while(n % i == 0){
                n /= i;
            }
        }
    }
    if(n > 1)
        ans = n;
    return ans;
}

int ham8(int n){
    while(n){
        if(n % 10 == 6)
            return 1;
        n /= 10;
    }
    return 0;
}

int ham9(int n){
    int sum = 0;

```

```
while(n){
    sum += n % 10;
    n /= 10;
}
if(sum % 8 == 0) return 1;
else return 0;
}
```

```
int gt(int n){
    int res = 1;
    for(int i = 1; i <= n; i++)
        res *= i;
    return res;
}
```

```
int ham10(int n){
    int sum = 0;
    while(n){
        sum += gt(n % 10);
        n /= 10;
    }
    return sum;
}
```

```
bool ham11(int n){
    int r = n % 10;
    while(n){
        if(n % 10 != r)
            return false;
        n /= 10;
    }
    return true;
}
```

```
bool ham12(int n){
    int r = n % 10;
    while(n){
        if(n % 10 > r)
            return false;
        n /= 10;
    }
}
```

```
    }  
    return true;  
}  
  
int count(int n){  
    int dem = 0;  
    while(n){  
        ++dem;  
        n /= 10;  
    }  
    return dem;  
}  
  
int ham13(int n){  
    int sum = 0;  
    int d = count(n);  
    while(n){  
        sum += pow(n % 10, d);  
        n /= 10;  
    }  
    return sum;  
}  
  
int main(){  
    int n; cin >> n;  
    cout << ham1(n) << endl;  
    cout << ham2(n) << endl;  
    cout << ham3(n) << endl;  
    cout << ham4(n) << endl;  
    cout << ham5(n) << endl;  
    cout << ham6(n) << endl;  
    cout << ham7(n) << endl;  
    cout << ham8(n) << endl;  
    cout << ham9(n) << endl;  
    cout << ham10(n) << endl;  
    cout << ham11(n) << endl;  
    cout << ham12(n) << endl;  
    cout << ham13(n) << endl;
```

```
}
```

### Constraints

$2 \leq n \leq 1000$ ;

### Output Format

In ra 13 dòng tương ứng với các yêu cầu ở trên.

### Sample Input 0

```
36
```

### Sample Output 0

```
0
9
6
3
63
2
3
1
0
726
0
1
45
```

### [Hàm, Lý Thuyết Số]. Luyện tập viết hàm 2 (Quan trọng)

Cho số nguyên không âm N, bạn hãy viết 10 hàm tương ứng để in ra kết quả, in ra 1 hoặc 0 cho mỗi yêu cầu (Mã nguồn tham khảo

Java : <https://ideone.com/t8Gcmd>)

- 

1. Số lượng chữ số chẵn của n là một số lẻ

- 

1. N có số lượng chữ số chẵn > số lượng chữ số lẻ

-

- 1. N có chữ số đầu bằng chữ số cuối
- 
- 1. Tổng chữ số của N có tận cùng là 8
- 
- 1. Tổng chữ số của N là số nguyên tố
- 
- 1. Các chữ số đứng cạnh nhau của n chênh lệch nhau đúng 1 đơn vị (số có 1 chữ số thỏa mãn)
- 
- 1. N có chữ số đầu tiên lớn hơn tất cả các chữ số còn lại của nó, (số có 1 chữ số thỏa mãn)
- 
- 1. N có tổng chữ số là một số trong dãy fibonacci
- 
- 1. N có tổng chữ số là một số thuận nghịch
- 
- 1. N chỉ bao gồm các chữ số 0, 6 hoặc 8

### **Input Format**

- Dòng 1 là T : số bộ test
- T dòng tiếp theo mỗi dòng là số tự nhiên N

### **Constraints**

- $1 \leq T \leq 1000$
- $0 \leq N \leq 10^{18}$

### **Output Format**

- Đối với mỗi test in ra 10 dòng tương ứng với kết quả của 10 yêu cầu trên, mỗi test cách nhau 1 dòng trống



**Sample Input 0**

1  
87654418

**Sample Output 0**

1  
1  
1  
0  
1  
0  
0  
0  
0  
0

**[Hàm, Lý Thuyết Số]. Bài 3. Kiểm Tra Số Nguyên Tố**

Kiểm tra một số nguyên không âm N có phải là số nguyên tố hay không?

**Input Format**

Dòng duy nhất chứa số nguyên dương N

**Constraints**

$0 \leq N \leq 10^9$

**Output Format**

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

**Sample Input 0**

999999999

**Sample Output 0**

NO

**Sample Input 1**

17

**Sample Output 1**

YES

### [Hàm, Lý Thuyết Số]. Bài 4. Số Chính Phương

Kiểm tra một số nguyên có phải là số chính phương hay không? Định nghĩa số chính

phương: [https://vi.wikipedia.org/wiki/S%E1%BB%91\\_ch%C3%ADnh\\_ph%C6%B0%C6%A1ng](https://vi.wikipedia.org/wiki/S%E1%BB%91_ch%C3%ADnh_ph%C6%B0%C6%A1ng)

#### Input Format

Một số nguyên dương N

#### Constraints

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

#### Output Format

In ra YES nếu N là số chính phương, ngược lại in NO

#### Sample Input 0

```
169
```

#### Sample Output 0

```
YES
```

### [Hàm, Lý Thuyết Số]. Bài 5. Số Chính Phương trong đoạn

In ra các số chính phương trong đoạn từ a tới b. Bài này bạn nào code bằng java thì có thể bỏ qua vì test lớn quá Java không chạy xong trong 8s.

#### Input Format

2 số nguyên dương a, b

#### Constraints

$$1 \leq a \leq b \leq 10^{12}$$

#### Output Format

In ra các số chính phương trong đoạn giữa 2 số a, b trên một dòng. Các số cách nhau một khoảng trắng.

#### Sample Input 0

```
10 20
```

### Sample Output 0

16

### [Hàm, Lý Thuyết Số]. Bài 6. 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 Format

Một dòng hai số nguyên dương tương ứng, cách nhau một khoảng trống.

#### Constraints

Các số đều không vượt quá 9 chữ số.

#### Output Format

Viết ra số lượng các số thuần nguyên tố tương ứng

### Sample Input 0

2345 6789

### Sample Output 0

15

### [Hàm, Lý Thuyết Số]. Bài 7. Số Chính Phương 3

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

#### Input Format

2 số nguyên dương a, b

#### Constraints

$1 \leq a \leq b \leq 10^{18}$

#### Output Format

Số lượng số chính phương trong đoạn  $[a, b]$

**Sample Input 0**

```
1 1000000000
```

**Sample Output 0**

```
31622
```

**[Hàm, Lý Thuyết Số]. Bài 8. Số Lộc Phát**

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

**Input Format**

Số nguyên  $n$

**Constraints**

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

**Output Format**

In ra 1 nếu  $n$  là số lộc phát, ngược lại in 0

**Sample Input 0**

```
60806
```

**Sample Output 0**

```
1
```

**[Hàm, Lý Thuyết Số]. Bài 9. Ước số nguyên tố nhỏ nhất**

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 Format**

Một số N được ghi trên một dòng.

### Constraints

$$1 \leq N \leq 100000$$

### Output Format

Đưa ra kết quả theo từng dòng

### Sample Input 0

```
6
```

### Sample Output 0

```
1
2
3
2
5
2
```

## [Hàm, Lý Thuyết Số]. Bài 10. Phân tích thừa số nguyên tố

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

### Input Format

Số nguyên dương N

### Constraints

$$2 \leq N \leq 10^{16}$$

### Output Format

Phân tích thừa số nguyên tố của N, xem ví dụ để rõ hơn format.

### Sample Input 0

```
60
```

### Sample Output 0

```
2^2 * 3^1 * 5^1
```

## [Hàm, Lý Thuyết Số]. Bài 11. Số Sphenic

Số nguyên dương  $N$  được gọi là số Sphenic nếu  $N$  được phân tích duy nhất dưới dạng tích của ba thừa số nguyên tố 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 Format

Một số nguyên dương  $N$

### Constraints

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

### Output Format

Đưa ra 1 hoặc 0 tương ứng với  $N$  là số Sphenic hoặc không.

### Sample Input 0

```
999923001838986077
```

### Sample Output 0

```
1
```

### Sample Input 1

```
30
```

### Sample Output 1

```
1
```

## [Hàm, Lý Thuyết Số]. Bài 12. 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 thừa số nguyên tố trong phân tích của  $N$ . Ví dụ  $N = 666$  có các thừa số nguyên tố là 2, 3, 3, 37 có tổng các chữ số là 18.

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

```

using ll = long long;

int tong(int n){
    int sum = 0;
    while(n){
        sum += n % 10;
        n /= 10;
    }
    return sum;
}

bool smith(int n){
    int tong1 = tong(n); // tong chu so cua n
    int tong2 = 0;
    int tmp = n;
    for(int i = 2; i <= sqrt(n); i++){
        if(n % i == 0){
            while(n % i == 0){
                tong2 += tong(i);
                n /= i;
            }
        }
    }
    if(tmp == n) return false; // n la snt
    if(n > 1)
        tong2 += tong(n);
    return tong1 == tong2;
}

int main(){
    ll n; cin >> n;
    if(smith(n)) cout << "YES\n";
    else cout << "NO\n";
}

```

### Input Format

Số nguyên dương N

**Constraints**

$1 \leq N \leq 10^8$ .

**Output Format**

In ra YES nếu N là số Smith, ngược lại in ra NO.

**Sample Input 0**

22

**Sample Output 0**

YES

**[Hàm, Lý Thuyết Số]. Bài 13. Ước số nguyên tố lớn nhất**

Tìm ước số nguyên tố lớn nhất của một số nguyên dương.

**Input Format**

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

**Constraints**

$1 \leq T \leq 500$ ;  $2 \leq N \leq 10000000$

**Output Format**

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

**Sample Input 0**

2  
10  
17

**Sample Output 0**

5  
17

**[Hàm, Lý Thuyết Số]. Bài 14. Bình phương nguyên tố 1**



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 Format

2 số nguyên dương  $a, b$

### Constraints

$$1 \leq a \leq b \leq 10^6$$

### Output Format

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

### Sample Input 0

```
4 50
```

### Sample Output 0

```
4 8 9 12 16 18 20 24 25 27 28 32 36 40 44 45 48 49 5
```

## [Hàm, Lý Thuyết Số]. Bài 15. Bình phương số nguyên tố 2

Một số được coi là số đẹp khi nếu nó chia hết cho một số nguyên tố nào đó **thì cũng phải** 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 Format

2 số nguyên dương  $a, b$

### Constraints

$$1 \leq a \leq b \leq 10^6$$

### Output Format

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

### Sample Input 0

```
3 49
```

### Sample Output 0

4 8 9 16 25 27 32 36 49

### [Hàm, Lý Thuyết Số]. Bài 16. 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.

Định lý Euclid - Euler : ' Nếu  $p$  là số nguyên tố và  $2^p - 1$  cũng là số nguyên tố thì :  $2^{(p-1)} * (2^p - 1)$  sẽ tạo thành 1 số hoàn hảo. Ví dụ  $p = 2$ ,  $2^2 - 1 = 3$ ,  $2 * 3 = 6 \Rightarrow$  HH Ví dụ  $p = 3$ ,  $2^3 - 1 = 7$ ,  $4 * 7 = 28 \Rightarrow$  HH

#### Input Format

Số nguyên dương  $N$

#### Constraints

$1 \leq N \leq 9 * 10^{18}$

#### Output Format

In YES nếu  $N$  là số hoàn hảo, ngược lại in NO

#### Sample Input 0

28

#### Sample Output 0

YES

#### Explanation 0

28 có các ước thực sự là 1, 2, 4, 7, 14 có tổng bằng 28.

### [Hàm, Lý Thuyết Số]. Bài 17. Số thuận nghịch có 3 ước 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 Format

2 số a, b

### Constraints

$$1 \leq a \leq b \leq 10^7$$

### Output Format

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

### Sample Input 0

```
1 1000
```

### Sample Output 0

```
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
```

## [Hàm, Lý Thuyết Số]. Bài 18. Số thuận nghịch, 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 các số đẹp trong đoạn giữa 2 số nguyên cho trước, các số cách nhau một dấu cách.

### Input Format

2 số nguyên a, b

### Constraints

$$1 \leq a \leq b \leq 10^6$$

### Output Format

Liệt kê các số đẹp trong đoạn, các số viết cách nhau một khoảng trống

### Sample Input 0

```
1 400
```

### Sample Output 0

```
161
```

### [Hàm, Lý Thuyết Số]. Bài 19. Chữ số cuối cùng lớn nhất

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

#### Input Format

Số nguyên dương  $n$

#### Constraints

$$1 \leq n \leq 10^7$$

#### Output Format

Dòng đầu tiên liệt kê các số thỏa mãn, và dòng thứ 2 in ra số lượng số thỏa mãn.

#### Sample Input 0

```
200
```

#### Sample Output 0

```
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
```

### [Hàm, Lý Thuyết Số]. Bài 20. Ước chung lớn nhất, bội chung nhỏ nhất

Tìm ước chung lớn nhất và bội chung nhỏ nhất của 2 số nguyên

#### Input Format

2 số nguyên  $a, b$

#### Constraints

$$1 \leq a, b \leq 10^{12}$$

#### Output Format

Ước chung lớn nhất và bội chung nhỏ nhất, dữ liệu đảm bảo BCNN của 2 số không vượt quá số int 64bit

**Sample Input 0**

20 50

**Sample Output 0**

10 100

**[Hàm, Lý Thuyết Số]. Bài 21. Kiểm Tra Số Nguyên Tố Với Nhiều Test**

Ở bài tập này yêu cầu bạn kiểm tra số nguyên tố với nhiều trường hợp khác nhau.

**Input Format**

Dòng đầu tiên là số lượng test case T; Mỗi test case là một số nguyên n

**Constraints**

$1 \leq T \leq 1000$ ;  $0 \leq n \leq 10^6$

**Output Format**

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

**Sample Input 0**

20  
364  
12401  
4152  
4624  
12783  
1868  
14521  
24213  
4740  
19037  
6992  
9390  
8929  
27797  
18685

```
13291
11424
10292
3534
30641
```

### Sample Output 0

```
NO
YES
NO
NO
NO
NO
NO
NO
NO
NO
YES
NO
NO
YES
NO
NO
YES
NO
NO
NO
NO
```

### [Hàm, Lý Thuyết Số]. Bài 22. T-prime

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

### Input Format

Số nguyên dương  $N$

### Constraints

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

### Output Format

Đưa ra kết quả trên một dòng

### Sample Input 0

```
100
```

### Sample Output 0

```
4 9 25 49
```

## [Hàm, Lý Thuyết Số]. Bài 23. T-prime2

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

### Input Format

Số nguyên dương  $N$

### Constraints

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

### Output Format

Đưa ra kết quả mỗi theo từng dòng.

### Sample Input 0

```
838000000000
```

### Sample Output 0

```
72397
```

## [Hàm, Lý Thuyết Số]. Bài 24. Bậc của thừa số nguyên tố trong $N!$

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 Format

Cặp số  $N, p$  được viết cách nhau một khoảng trống.

### Constraints

$$1 \leq N \leq 10^{14}; 2 \leq p \leq 5000$$

### Output Format

Đưa ra kết quả trên một dòng

### Sample Input 0

```
7 3
```

### Sample Output 0

```
2
```

## [Hàm, Lý Thuyết Số]. Bài 25. Đếm chữ số 0 liên tiếp tính từ cuối của $N!$

In ra số lượng chữ số 0 liên tiếp tính từ cuối của  $N!$  Ví dụ bạn có  $N = 10$ ,  $10! = 3628800$ . Như vậy,  $10!$  có 2 chữ số 0 liên tiếp tính từ cuối.

Gợi ý : Số 0 ở cuối của  $N!$  có được bằng cách nhân số 2 với số 5, cứ mỗi cặp (2, 5) trong  $N!$  sẽ tạo ra 1 số 0 ở cuối vì thế số lượng chữ số 0 liên tiếp tính từ cuối của  $N!$  sẽ bằng số cặp (2, 5) trong  $N!$ , mà số lượng số 2 trong  $N!$  bao giờ cũng lớn hơn số lượng số 5! trong  $N!$   $\Rightarrow$  Số cặp (2, 5) trong  $N!$  chính là số lượng số 5 trong  $N!$  hay bậc của 5 trong  $N!$

### Input Format

Số nguyên dương  $N$

### Constraints

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

### Output Format

In ra số lượng chữ số 0 liên tiếp tính từ cuối của  $N!$ . Kết quả lấy dư với 1000000007.

### Sample Input 0

```
10
```



### Sample Output 0

2

### [Hàm, Lý Thuyết Số]. Bài 26. Số Fibonacci 1

Dãy số Fibonacci được định nghĩa như sau:  $F_1 = 0$ ,  $F_2 = 1$ ;  $F_i = F_{i-1} + F_{i-2}$ . Hãy viết chương trình in ra số Fibonacci thứ  $n$ .

#### Input Format

Số nguyên dương  $n$

#### Constraints

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

#### Output Format

Số fibonacci thứ  $n$  lấy dư với 1000000007

### Sample Input 0

5

### Sample Output 0

3

#### Explanation 0

Các số fibonacci đầu tiên : 0 1 1 2 3 5 8. Vậy số fibonacci thứ 5 là 3

### [Hàm, Lý Thuyết Số]. Bài 27. Số Fibonacci 2

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].Bài 28. Fibonacci 3**

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].Bài 29. Tổng chia dư

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, Lý Thuyết Số].Bài 30. Tích chia dư

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, Lý thuyết số]. Bài 31. Chữ số cuối cùng**

Cho số  $p$  và  $x$ , nhiệm vụ của bạn là tìm chữ số cuối cùng của  $p^x$

**Input Format**

1 dòng duy nhất chứa 2 số  $p$  và  $x$

**Constraints**

$1 \leq p \leq 1000$ ;  $0 \leq x \leq 100000$ ;

**Output Format**

In ra chữ số cuối cùng của  $p^x$

**Sample Input 0**

```
8 7
```

**Sample Output 0**

```
2
```

**Sample Input 1**

```
893 99679
```

**Sample Output 1**

```
7
```