

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

- **Problem**
- Submissions
- Discussions

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

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

•

11. 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.

•

12. 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.

•

13. 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)

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

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
- 2. N có số lượng chữ số chẵn > số lượng chữ số lẻ
- 3. N có chữ số đầu bằng chữ số cuối
- 4. Tổng chữ số của N có tận cùng là 8
- 5. Tổng chữ số của N là số nguyên tố

-
- 6. Các chữ số đứng cạnh nhau của n chênh lệch nhau đúng 1 đơn vị (số có 1 chữ số thỏa mãn)
-
- 7. N có chữ số đầu tiên lớn hơn tất cả các chữ số còn lại của nó, (số có 1 chữ số thỏa mãn)
-
- 8. N có tổng chữ số là một số trong dãy fibonacci
-
- 9. N có tổng chữ số là một số thuận nghịch
-
- 10. 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
```

0
0

[Hàm, Lý Thuyết Số]. Bài 1. Kiểm Tra Số Nguyên Tố

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

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 3. Kiểm Tra Số Nguyên Tố Với Nhiều Test

- [Problem](#)

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

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 4. Số Nguyên Tố Có Các Chữ Số Nguyên Tố

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

Viết chương trình đếm xem trong đoạn giữa 2 số a và b có bao nhiêu số là số nguyên tố và tất cả các chữ số của nó cũng là số nguyên tố. Gợi ý : viết hàm 2 là hàm số nguyên tố và hàm kiểm tra tất cả các chữ số của nó là số nguyên tố sau đó duyệt từ a đến b và kết hợp 2 hàm này để kiểm tra.

Input Format

Gồm 2 số nguyên dương a và b.

Constraints

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

Output Format

Ghi ra số lượng số thỏa mãn trên một dòng.

Sample Input 0

1234 5678

Sample Output 0

26

[Hàm, Lý Thuyết Số]. Bài 6. Số thuần nguyên tố

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

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 8. T-prime

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

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 9. T-prime2

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

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 10. Đếm ước của một số nguyên dương

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

Cho phân tích thừa số nguyên tố của một số nguyên dương N , hãy đếm số lượng ước số của số nguyên dương đó. Ví dụ $N = 60 = 2^2 * 3^1 * 5^1$.

Input Format

Dòng đầu tiên là T : số lượng thừa số nguyên tố khác nhau của N T dòng tiếp theo, mỗi dòng là 2 số nguyên p và e lần lượt là thừa số nguyên tố và số mũ tương ứng

Constraints

$1 \leq T \leq 100$; $2 \leq p \leq 100000$; $1 \leq e \leq 100000$;

Output Format

In ra số lượng ước số của N , vì kết quả quá lớn, hãy lấy dư với số 1000000007 ($1e9 + 7$)

Sample Input 0

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

Sample Output 0

```
12
```

[Hàm, Lý Thuyết Số]. Bài 20. Số Chính Phương

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

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

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 21. Số Chính Phương trong đoạn

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

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 22. Số Chính Phương 3

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

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 10000000000

Sample Output 0

31622

[Hàm, Lý Thuyết Số]. Bài 23. Tổng ước số

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

Tính tổng ước của 1 số nguyên dương N.

Input Format

1 số nguyên dương N

Constraints

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

Output Format

Tổng ước số của N

Sample Input 0

100

Sample Output 0

217

Sample Input 1

28

Sample Output 1

56

[Hàm, Lý Thuyết Số]. Bài 25. Số có số lượng ước là số lẻ

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

Kiểm tra xem một số có số lượng ước số của nó là số lẻ

Input Format

Số nguyên dương N

Constraints

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

Output Format

In ra YES nếu N là số có số ước là số lẻ, ngược lại in NO.

Sample Input 0

```
100
```

Sample Output 0

```
YES
```

Explanation 0

Số 100 có các ước 1, 2, 4, 5, 10, 20, 25, 50, 100. Vậy 100 có 9 ước là số lẻ, nên đáp án là YES.

[Hàm, Lý Thuyết Số]. Bài 28. Số Lộc Phát

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

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 29. Số thuận nghịch, lộc phát

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

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 30. Chữ số cuối cùng lớn nhất

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

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 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. Gợi 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, Lý Thuyết Số]. Bài 11. Ước số nguyên tố nhỏ nhất

- [Problem](#)

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

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 12. Phân tích thừa số nguyên tố

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

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 15. Số Sphenic

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

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 16. Số Smith

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

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 17. Ước số nguyên tố lớn nhất

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

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 18. Bình phương nguyên tố 1

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

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 50
```

[Hàm, Lý Thuyết Số]. Bài 19. Bình phương số nguyên tố 2

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

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 5. Goldbach conjecture

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

Theo Goldbach conjecture, một số nguyên dương chẵn ≥ 4 đều có thể biểu diễn dưới dạng tổng của 2 số nguyên tố. Cho số nguyên dương chẵn $N \geq 4$. Hãy liệt kê các cặp số nguyên tố p, q có tổng đúng bằng N. Ví dụ $N = 6$ ta có 1 cặp số nguyên tố là $3 + 3 = 6$.

Input Format

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.

Constraints

$1 \leq T \leq 1000$; $4 \leq N \leq 1000000$

Output Format

Đưa ra kết quả mỗi test theo từng dòng, mỗi dòng là một cặp số thỏa mãn. Chú ý mỗi cặp chỉ liệt kê một lần, không xét đến thứ tự.

Sample Input 0

```
1
10
```

Sample Output 0

```
3 7
5 5
```

[Hàm, Lý Thuyết Số]. Bài 7. Cặp số nguyên tố cùng nhau

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

Hãy liệt kê các cặp số nguyên tố cùng nhau và có giá trị khác nhau trong đoạn $[a,b]$ theo thứ tự từ nhỏ đến lớn.

Input Format

Chỉ có một dòng ghi hai số a, b

Constraints

$2 \leq a \leq b \leq 1000$

Output Format

Các cặp số i,j thỏa mãn được viết lần lượt trên từng dòng theo định dạng (i,j), theo thứ tự từ điển.

Sample Input 0

5 46

Sample Output 0

(5,6)
(5,7)
(5,8)
(5,9)
(5,11)
(5,12)
(5,13)
(5,14)
(5,16)
(5,17)
(5,18)
(5,19)
(5,21)
(5,22)
(5,23)
(5,24)
(5,26)
(5,27)
(5,28)
(5,29)
(5,31)
(5,32)
(5,33)
(5,34)
(5,36)
(5,37)
(5,38)
(5,39)
(5,41)
(5,42)
(5,43)
(5,44)
(5,46)
(6,7)
(6,11)
(6,13)
(6,17)
(6,19)
(6,23)
(6,25)
(6,29)
(6,31)
(6,35)
(6,37)
(6,41)
(6,43)
(7,8)
(7,9)

(7,10)
(7,11)
(7,12)
(7,13)
(7,15)
(7,16)
(7,17)
(7,18)
(7,19)
(7,20)
(7,22)
(7,23)
(7,24)
(7,25)
(7,26)
(7,27)
(7,29)
(7,30)
(7,31)
(7,32)
(7,33)
(7,34)
(7,36)
(7,37)
(7,38)
(7,39)
(7,40)
(7,41)
(7,43)
(7,44)
(7,45)
(7,46)
(8,9)
(8,11)
(8,13)
(8,15)
(8,17)
(8,19)
(8,21)
(8,23)
(8,25)
(8,27)
(8,29)
(8,31)
(8,33)
(8,35)
(8,37)
(8,39)
(8,41)
(8,43)
(8,45)
(9,10)
(9,11)

(9,13)
(9,14)
(9,16)
(9,17)
(9,19)
(9,20)
(9,22)
(9,23)
(9,25)
(9,26)
(9,28)
(9,29)
(9,31)
(9,32)
(9,34)
(9,35)
(9,37)
(9,38)
(9,40)
(9,41)
(9,43)
(9,44)
(9,46)
(10,11)
(10,13)
(10,17)
(10,19)
(10,21)
(10,23)
(10,27)
(10,29)
(10,31)
(10,33)
(10,37)
(10,39)
(10,41)
(10,43)
(11,12)
(11,13)
(11,14)
(11,15)
(11,16)
(11,17)
(11,18)
(11,19)
(11,20)
(11,21)
(11,23)
(11,24)
(11,25)
(11,26)
(11,27)
(11,28)

(11,29)
(11,30)
(11,31)
(11,32)
(11,34)
(11,35)
(11,36)
(11,37)
(11,38)
(11,39)
(11,40)
(11,41)
(11,42)
(11,43)
(11,45)
(11,46)
(12,13)
(12,17)
(12,19)
(12,23)
(12,25)
(12,29)
(12,31)
(12,35)
(12,37)
(12,41)
(12,43)
(13,14)
(13,15)
(13,16)
(13,17)
(13,18)
(13,19)
(13,20)
(13,21)
(13,22)
(13,23)
(13,24)
(13,25)
(13,27)
(13,28)
(13,29)
(13,30)
(13,31)
(13,32)
(13,33)
(13,34)
(13,35)
(13,36)
(13,37)
(13,38)
(13,40)
(13,41)

(13,42)
(13,43)
(13,44)
(13,45)
(13,46)
(14,15)
(14,17)
(14,19)
(14,23)
(14,25)
(14,27)
(14,29)
(14,31)
(14,33)
(14,37)
(14,39)
(14,41)
(14,43)
(14,45)
(15,16)
(15,17)
(15,19)
(15,22)
(15,23)
(15,26)
(15,28)
(15,29)
(15,31)
(15,32)
(15,34)
(15,37)
(15,38)
(15,41)
(15,43)
(15,44)
(15,46)
(16,17)
(16,19)
(16,21)
(16,23)
(16,25)
(16,27)
(16,29)
(16,31)
(16,33)
(16,35)
(16,37)
(16,39)
(16,41)
(16,43)
(16,45)
(17,18)
(17,19)

(17,20)
(17,21)
(17,22)
(17,23)
(17,24)
(17,25)
(17,26)
(17,27)
(17,28)
(17,29)
(17,30)
(17,31)
(17,32)
(17,33)
(17,35)
(17,36)
(17,37)
(17,38)
(17,39)
(17,40)
(17,41)
(17,42)
(17,43)
(17,44)
(17,45)
(17,46)
(18,19)
(18,23)
(18,25)
(18,29)
(18,31)
(18,35)
(18,37)
(18,41)
(18,43)
(19,20)
(19,21)
(19,22)
(19,23)
(19,24)
(19,25)
(19,26)
(19,27)
(19,28)
(19,29)
(19,30)
(19,31)
(19,32)
(19,33)
(19,34)
(19,35)
(19,36)
(19,37)

(19,39)
(19,40)
(19,41)
(19,42)
(19,43)
(19,44)
(19,45)
(19,46)
(20,21)
(20,23)
(20,27)
(20,29)
(20,31)
(20,33)
(20,37)
(20,39)
(20,41)
(20,43)
(21,22)
(21,23)
(21,25)
(21,26)
(21,29)
(21,31)
(21,32)
(21,34)
(21,37)
(21,38)
(21,40)
(21,41)
(21,43)
(21,44)
(21,46)
(22,23)
(22,25)
(22,27)
(22,29)
(22,31)
(22,35)
(22,37)
(22,39)
(22,41)
(22,43)
(22,45)
(23,24)
(23,25)
(23,26)
(23,27)
(23,28)
(23,29)
(23,30)
(23,31)
(23,32)

(23,33)
(23,34)
(23,35)
(23,36)
(23,37)
(23,38)
(23,39)
(23,40)
(23,41)
(23,42)
(23,43)
(23,44)
(23,45)
(24,25)
(24,29)
(24,31)
(24,35)
(24,37)
(24,41)
(24,43)
(25,26)
(25,27)
(25,28)
(25,29)
(25,31)
(25,32)
(25,33)
(25,34)
(25,36)
(25,37)
(25,38)
(25,39)
(25,41)
(25,42)
(25,43)
(25,44)
(25,46)
(26,27)
(26,29)
(26,31)
(26,33)
(26,35)
(26,37)
(26,41)
(26,43)
(26,45)
(27,28)
(27,29)
(27,31)
(27,32)
(27,34)
(27,35)
(27,37)

(27,38)
(27,40)
(27,41)
(27,43)
(27,44)
(27,46)
(28,29)
(28,31)
(28,33)
(28,37)
(28,39)
(28,41)
(28,43)
(28,45)
(29,30)
(29,31)
(29,32)
(29,33)
(29,34)
(29,35)
(29,36)
(29,37)
(29,38)
(29,39)
(29,40)
(29,41)
(29,42)
(29,43)
(29,44)
(29,45)
(29,46)
(30,31)
(30,37)
(30,41)
(30,43)
(31,32)
(31,33)
(31,34)
(31,35)
(31,36)
(31,37)
(31,38)
(31,39)
(31,40)
(31,41)
(31,42)
(31,43)
(31,44)
(31,45)
(31,46)
(32,33)
(32,35)
(32,37)

(32,39)
(32,41)
(32,43)
(32,45)
(33,34)
(33,35)
(33,37)
(33,38)
(33,40)
(33,41)
(33,43)
(33,46)
(34,35)
(34,37)
(34,39)
(34,41)
(34,43)
(34,45)
(35,36)
(35,37)
(35,38)
(35,39)
(35,41)
(35,43)
(35,44)
(35,46)
(36,37)
(36,41)
(36,43)
(37,38)
(37,39)
(37,40)
(37,41)
(37,42)
(37,43)
(37,44)
(37,45)
(37,46)
(38,39)
(38,41)
(38,43)
(38,45)
(39,40)
(39,41)
(39,43)
(39,44)
(39,46)
(40,41)
(40,43)
(41,42)
(41,43)
(41,44)
(41,45)

(41,46)
(42,43)
(43,44)
(43,45)
(43,46)
(44,45)
(45,46)

[Hàm, Lý Thuyết Số]. Bài 32. Thừa số nguyên tố thứ K

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

Đưa ra số nguyên tố thứ k trong phân tích thừa số nguyên tố của một số nguyên dương n. Ví dụ n=28, k=3 ta có kết quả là 7 vì $28=2 \times 2 \times 7$.

Input Format

2 số n,k

Constraints

$1 \leq n, k \leq 10^9$

Output Format

In ra thừa số nguyên tố thứ k của n, nếu n không có thừa số nguyên tố thứ k thì in ra -1.

Sample Input 0

28 3

Sample Output 0

7

Sample Input 1

8 5

Sample Output 1

-1

[Hàm, Lý Thuyết Số]. Bài 33. Liệt kê chữ số nguyên tố

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

Liệt kê số lần xuất hiện của chữ số nguyên tố của 1 số theo thứ tự từ nhỏ đến lớn

Input Format

Số nguyên dương n

Constraints

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

Output Format

Liệt kê các chữ số nguyên tố của n cùng số lần xuất hiện, theo thứ tự từ bé đến lớn. Sau đó cách ra 1 dòng và liệt kê các chữ số nguyên tố của n cùng số lần xuất hiện, nhưng theo thứ tự xuất hiện trong n . Xem thêm test case để rõ hơn.

Sample Input 0

```
2273
```

Sample Output 0

```
2 2
3 1
7 1

2 2
7 1
3 1
```

[Hàm, Lý Thuyết Số]. Bài 35. Ước chung lớn nhất, bội chung nhỏ nhất

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

- [Discussions](#)

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 41. Số gần thuận nghịch

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

Một số được coi là đẹp nếu chữ số đầu gấp đôi chữ số cuối hoặc ngược lại; đồng thời các chữ số từ vị trí thứ 2 đến gần cuối thỏa mãn là một số thuận nghịch. Ví dụ: các số 36788766; 12345654322 là các số đẹp. Viết chương trình kiểm tra số đẹp theo tiêu chí trên.

Input Format

Số nguyên dương n

Constraints

$99 \leq n \leq 10^{18}$

Output Format

Ghi ra YES tương ứng với số đẹp, NO trong trường hợp ngược lại

Sample Input 0

```
122222
```

Sample Output 0

```
YES
```

[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]. 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

[Hàm, Lý Thuyết Số]. Bài 63. Số Fibonacci 1

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

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 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 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