* [**Problem**](https://www.hackerrank.com/contests/js-club-codetour-2023-contest-02-bracket-b/challenges/change-to-binary-number)
* [**Submissions**](https://www.hackerrank.com/contests/js-club-codetour-2023-contest-02-bracket-b/challenges/change-to-binary-number/submissions)
* [**Leaderboard**](https://www.hackerrank.com/contests/js-club-codetour-2023-contest-02-bracket-b/challenges/change-to-binary-number/leaderboard)
* [**Discussions**](https://www.hackerrank.com/contests/js-club-codetour-2023-contest-02-bracket-b/challenges/change-to-binary-number/forum)

Sau khi viết code thì các dòng code sẽ được chuyển đổi về binary code để máy thực thi. Nhiệm vụ cua bạn lần này là chuyển một số decimal thành số binary(số nhị phân)

**Input Format**

Nhập một số

**Constraints**

Chỉ nhập số không nhập chữ

**Output Format**

số nhị phân

**Sample Input 0**

33

**Sample Output 0**

100001

**Sample Input 1**

A

**Sample Output 1**

Invalid

Ngày xửa ngày xưa, ở một ngôi làng nhỏ, có một lập trình viên trẻ tên là JS, người say mê thế giới phép thuật của JavaScript. Anh ta đã nghe kể về một bài thần bí có thể biến đổi một chuỗi thành một số bằng cách sử dụng huyền thoại hàm gọi là parseInt.

Một ngày nọ, một nhóm người làng đến gặp JS, yêu cầu anh ta giúp đỡ giải quyết một vấn đề. Họ đã thu thập được một chuỗi đầy các con số thần kỳ, nhưng họ cần thực hiện các phép tính số học với những con số đó. Tuy nhiên, kiến thức của họ chỉ giới hạn trong việc làm việc với chuỗi, và họ không biết cách chuyển đổi chúng thành các số thực sự.

Với lòng tự tin vào khả năng của mình, JS nhận lời thách thức. Anh bắt đầu bằng cách viết một hàm sử dụng nghệ thuật cổ xưa của JavaScript. Anh đặt tên cho nó là chuyenDoiThanhSo, và nó có một tham số duy nhất gọi là chuoiSo.

Với quyết tâm mãnh liệt, John thực hiện hàm này, sử dụng phép thuật mạnh mẽ gọi là parseInt. Anh biết rằng phép thuật này có thể trích xuất giá trị số từ một chuỗi. Tuy nhiên, anh cần phải thận trọng, vì phép thuật có thể không hoạt động đúng nếu chuỗi chứa các ký tự không phải là số.

Để đảm bảo việc chuyển đổi thành công, JS trước tiên loại bỏ bất kỳ khoảng trắng đầu chuỗi nào, sử dụng phép thuật trim. Tiếp theo, anh sử dụng phép thuật mạnh mẽ parseInt trên chuỗi đã được làm sạch, biến nó thành một giá trị số.

Nhưng JS không dừng lại ở đó. Anh biết rằng số thu được có thể quá lớn hoặc quá nhỏ cho một số phép tính cụ thể. Để tránh bất kỳ sự cố nào, anh kiểm tra xem số đó có nằm trong phạm vi an toàn của một số nguyên có dấu 32-bit không. Nếu số vượt quá giới hạn, anh một cách tinh tế giữ số lại ở giá trị tối đa hoặc tối thiểu tương ứng.

Cuối cùng, với bài phép hoàn chỉnh, JS trả lại số đã biến đổi cho những người dân trong làng, người ta ngưỡng mộ khả năng JavaScript của anh. Họ có thể thực hiện các phép tính của mình dễ dàng, nhờ khả năng thần kỳ của JS trong nghệ thuật phép thuật.

Và vì thế, danh tiếng của JS như một pháp sư JavaScript tài ba lan tỏa khắp nơi.

**Input Format**

String s

**Constraints**

Không giới hạn

**Output Format**

Tổng tất cả các số có trong chuỗi(Lưu ý số trong String có thể rất lớn)

**Sample Input 0**

JSwith1/6love

**Sample Output 0**

7