Fibary

Computer Science Society
Programming Contest
Winter 2009

We are familiar with radix number representations where digits read from right-to-left (least significant to most significant) signify increasing powers of some radix (i.e. base) number, e.g. $10^0, 10^1, 10^2, 10^3, \ldots$ in decimal representation or $2^0, 2^1, 2^2, 2^3, \ldots$ in binary representation. For example, the binary number `101001` represents

$$1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 41.$$

In this problem, we consider a number representation where digits read from right-to-left (least significant to most significant) signify increasing Fibonacci numbers $1, 2, 3, 5, 8, 13, 21, \ldots$ we'll call them *fibary* numbers. Each digit of a fibary number is either `0` or `1`. For example, the fibary number `101001` represents

$$1 \cdot 13 + 0 \cdot 8 + 1 \cdot 5 + 0 \cdot 3 + 0 \cdot 2 + 1 \cdot 1 = 19.$$

While each number has exactly one radix representation without leading zeroes, it can have more than one fibary representation without leading zeroes. For example, the fibary number `11111` also represents

$$1 \cdot 8 + 1 \cdot 5 + 1 \cdot 3 + 1 \cdot 2 + 1 \cdot 1 = 19.$$

However, each number has exactly one fibary representation without leading zeroes or successive ones—its canonical fibary representation. Of all the fibary representations of a number, the canonical one is the largest when viewed as a binary number.

Input Format

Each line contains the decimal representation of a nonnegative number less than 2^{31}.

Output Format

For each decimal number input, output a line containing its canonical fibary representation.

Input Sample	*Output Sample*
`0`	`0`
`19`	`101001`
`10`	`10010`
`100`	`1000010100`
`1000000000`	`1010000100100001010101000001000101000101001`
`123456`	`1000000001001001000000000`
`654321`	`10010001000001000000000000001`
`2009`	`1001000010000001`
`317810`	`1010101010101010101010101010`