

SỐ LỚN NHẤT (MAXNUM.*)

Mỗi số nguyên bất kỳ có thể biểu diễn một cách duy nhất dưới dạng cơ số -2.

Dãy bit $b_k b_{k-1} \dots b_1 b_0$ ở dạng cơ số -2 biểu diễn số $\sum_{i=0}^k b_i (-2)^i$.

Với dãy n bit $an, an-1, a1, a0$ (có thể có các bit 0 ở đầu) ta có số nguyên x . Từ đây có thể dễ dàng tìm ra dãy bit $bn, bn-1, \dots, b1, b0$ (có thể có các bit 0 ở đầu) xác định số nguyên y , sao cho $x+y$ trong dạng biểu diễn cơ số -2 có nhiều bit 1 nhất. Ví dụ, với dãy bit ban đầu là 10101, các số $y = 10110$, $y = 10101$ và $y = 11010$ đều cho tổng $x+y$ có nhiều bit 1 nhất trong dạng biểu diễn ở cơ số -2 :

$$10101 + 10110 = 1111011$$

$$10101 + 10101 = 1111110$$

$$10101 + 11010 = 1101111$$

Trong số các y cho tổng có nhiều bit 1 nhất thì số $y = 10101 = 21_{10}$ là lớn nhất.

Yêu cầu: Cho số nguyên x dưới dạng xâu bit (có thể chứa các bit 0 ở đầu) độ dài không quá 50. Hãy tìm số nguyên y lớn nhất thỏa mãn các điều kiện:

- Trong dạng biểu diễn ở cơ số -2 có độ dài không quá độ dài xâu đã cho,
- Tổng $x+y$ có nhiều bit 1 nhất,
- Là số có giá trị lớn nhất thỏa mãn 2 điều kiện trên.

Đưa ra số tìm được dưới dạng cơ số -2 với số bit bằng số bit của xâu ban đầu (trong trường hợp cần thiết có thể bổ sung thêm các bit 0 không có nghĩa).

Dữ liệu: Vào từ file văn bản MAXNUM.INP gồm nhiều tests, mỗi test cho trên một dòng.

Kết quả: Đưa ra file văn bản MAXNUM.OUT, kết quả mỗi test đưa ra trên một dòng dưới dạng xâu bit có độ dài bằng xâu bit đã cho.

Ví dụ:

MAXNUM.INP
1
10101
0001
001011110001110000101

MAXNUM.OUT
1
10101
1110
101010001110001111010