

TÌM BIT (BDIGIT.CPP)

Từ chuỗi nhị phân S ban đầu chỉ gồm một bit 0, người ta tạo ra chuỗi nhị phân mới bằng cách ghép chuỗi S ban đầu với chính nó sau khi đã đảo tất cả các bit của S (nghĩa là đổi bit 1 thành bit 0 và bit 0 thành bit 1) và cứ lặp đi lặp lại các thao tác trên cho đến khi chuỗi S có 2^{64} bit. Ví dụ chuỗi S sau 4 phép biến đổi

```
0 (Ban đầu)
01
0110
01101001
0110100110010110
```

Các bit trong S sau đó được đánh số từ trái qua phải bắt đầu từ 0 trở đi

Yêu cầu: Cho biết số nguyên $n \in [0; 2^{64})$. Hãy tìm bit thứ n của chuỗi S

Dữ liệu:

- ☀ Dòng 1 chứa số nguyên dương $T \leq 10^5$ là số test
- ☀ T dòng tiếp theo, mỗi dòng chứa số một số nguyên n ứng với 1 test ($0 \leq n < 2^{64}$)

Kết quả: Ghi ra T dòng, mỗi dòng ghi chữ số thứ n của xâu nhị phân S ứng với một bộ dữ liệu

Ví dụ:

Sample Input	Sample Output
3	0
10	0
5	1
8	