

Tháp Hà Nội

Có 3 cọc A, B, C và có N chiếc đĩa đánh số từ 1 đến N có kích thước tương ứng là 1, 2, ..., N. Trạng thái ban đầu cả N chiếc đĩa đều ở cọc A, và đĩa to luôn ở dưới đĩa nhỏ. Xét phương pháp chuyển đĩa sau để chuyển đĩa từ cọc A sang cọc C sao cho *các đĩa to luôn luôn ở dưới các đĩa nhỏ* :

gọi thủ tục Chuyen(N, A, B, C).

Trong đó :

```
void Chuyen(n , c1, c2, c3){ /*
    if (n == 1) {
        chuyển đĩa nằm trên cùng c1 sang c3;
    };
    else {
        Chuyen(n-1, c1, c3, c2);
        Chuyen( 1, c1, c2, c3);
        Chuyen(n-1, c2, c1, c3);
    }
}
```

có nghĩa là ta chuyển n đĩa trên cùng của c1 sang c3 thông qua c2.

Số bước chuyển đĩa là $2^N - 1$ (rất lớn); do đó sẽ có $2^N - 1$ lần gọi { * }. Vậy ta có 2 bài toán sau để giải quyết:

1. Cho trước 1 số P, $P < 2^N$. Hỏi sau lần gọi thứ P thì trạng thái của N đĩa như thế nào?
2. Cho trước một trạng thái của N đĩa, bạn hãy xét xem trạng thái đó có xuất hiện trong quá trình chuyển đĩa từ A sang C theo phương pháp trên hay không? Nếu có xuất hiện thì đó là sau lần gọi { * } thứ bao nhiêu? (Giả sử là số Q)

Dữ liệu vào file HANOI.INP :

- + Dòng 1: ghi số N ($N < 101$).
- + Dòng 2: ghi số P (nên nhớ P có thể bằng $2^{100} - 1$).
- + Dòng 3: ghi 1 xâu gồm N ký tự chỉ gồm A,B,C là trạng thái của các đĩa (tất nhiên các đĩa to luôn luôn ở dưới các đĩa nhỏ).

Dữ liệu ra file HANOI.OUT :

- + Dòng 1: ghi 1 xâu gồm N ký tự là trạng thái đĩa sau lần gọi thứ P.
- + Dòng 2: là số Q. (Q = -1 nếu trạng thái đó không xuất hiện).

hanoi.inp	hanoi.out
3	CBA
2	7
CCC	