

Bumêran

Bumêran là công cụ đi săn bằng đá hoặc gỗ của thổ dân Úc. Nếu ném một cách thích hợp, bumêran sẽ bay tới đích rồi lượn về tay người ném. Để đáp ứng nhu cầu thể thao rèn luyện kỹ năng sử dụng bumêran, một công ty quyết định chế tạo bumêran bằng nhựa. Mỗi khi được ném với một lực nào đó về phía trước, bumêran sẽ bay theo một quỹ đạo nhất định và không nhất thiết quay trở về vị trí ban đầu.

Theo thiết kế, mỗi bumêran được gắn với một chuỗi S gồm không quá 250 ký tự lấy từ tập $\{ 'F', 'R' \}$. Quỹ đạo bay của mỗi cách ném bumêran dọc theo đường thẳng đều có thể mô tả được bởi chuỗi ký tự nhận được bằng cách xoá bớt một số ký tự của chuỗi S gắn với nó, trong đó mỗi ký tự cho biết bumêran bay qua một khoảng độ dài 1 về hướng nào:

- F – bumêran bay về phía trước,
- R – bumêran bay ngược trở lại.

Một cách ném bumêran dọc theo đường thẳng được gọi là an toàn nếu thoả mãn 2 điều kiện:

- Điều kiện Bumêran: vị trí bắt đầu và kết thúc của quỹ đạo bay là trùng nhau.
- Điều kiện an toàn: Nếu vị trí ban đầu xuất hiện trên quỹ đạo bay và quá trình bay còn chưa kết thúc thì hướng bay tiếp theo phải là 'F'.

Hai cách ném gọi là khác nhau nếu hai chuỗi mô tả quỹ đạo bay của chúng là khác nhau. Với mỗi bumêran, có thể có nhiều cách ném an toàn khác nhau.

Ví dụ, với bumêran có chuỗi $S = \text{'FFFRRRRRF'}$, ta có 3 cách ném an toàn: 'FR', 'FFRR' và 'FFFRRR'.

Lưu ý là trong quỹ đạo bay của bumêran vị trí ban đầu có thể xuất hiện nhiều lần.

Yêu cầu: Cho chuỗi S , hãy xác định số cách ném an toàn khác nhau có thể thực hiện.

Input

- Chứa chuỗi S .

Output

- Một số nguyên là số lượng cách ném an toàn có thể thực hiện.

Dữ liệu vào	Kết quả ra
FFFRRRRRF	3

Trò chơi đuổi bắt (cgame.*)

Một nhóm bạn chơi trò chơi đuổi bắt trong một mê cung gồm n phòng. Có $n - 1$ hành lang có độ dài bằng nhau nối giữa các cặp phòng và đảm bảo từ một phòng có thể tới được một phòng bất kì, hay nói theo ngôn ngữ tin học thì hệ thống các phòng và hành lang là một cây. Có một lối vào mê cung từ phòng số 1 (nút gốc) và mỗi phòng tương ứng là nút lá thì có lối ra. Khi bắt đầu trò chơi, một người chạy vào mê cung từ phòng số 1 gọi là *người chạy* và sẽ chạy xuôi xuống nút lá để chạy thoát ra ngoài, những người tham gia chơi còn lại sẽ ở những phòng là nút lá và mục tiêu của họ là bắt được người chạy (những người này gọi là *người đuổi*) bằng cách chạy ngược về nút gốc. Tốc độ chạy của mọi người là như nhau, người chạy sẽ bị bắt nếu ở một thời điểm nào đó, người đuổi và người chạy gặp nhau ở một phòng hoặc hành lang. Để trò chơi thú vị, nhóm bạn muốn tính xem cần ít nhất bao nhiêu người tham gia chơi với vai trò người đuổi để luôn có thể bắt được người chạy.

Yêu cầu: Tính số người đuổi ít nhất để luôn có thể bắt được người chạy.

Input

- Dòng đầu chứa số nguyên dương n ($n \leq 10^6$);
- Tiếp theo là $n - 1$ dòng, mỗi dòng chứa hai số u, v mô tả một hành lang nối giữa hai phòng u, v .

Output

- Gồm một số nguyên dương là đáp án của bài toán.

Input	Output
4 1 2 1 3 3 4	2

Phân số

Cho phân số $\frac{P}{Q}$, tiến hành biểu diễn phân số trong hệ cơ số thập phân, sau khi loại bỏ phần nguyên và dấu chấm thập phân (dấu ngăn cách giữa phần nguyên và phần thực) ta nhận được một xâu số S có độ dài vô hạn. Đánh số các kí tự của xâu S bắt đầu từ 1, để khảo sát phân số, với một xâu mẫu X người ta muốn tìm vị trí xuất hiện thứ k của X trong S .

Yêu cầu: Cho P, Q, k và xâu X , hãy xác định vị trí xuất hiện thứ k của X trong S , trong đó S là biểu diễn của phân số $\frac{P}{Q}$ trong hệ cơ số thập phân sau khi loại bỏ dấu chấm ngăn cách giữa phần nguyên và phần thực.

Input

- Dòng đầu tiên chứa ba số nguyên dương P, Q, k ($0 < P, Q, k \leq 10^6$);
- Dòng thứ hai chứa một xâu số X có độ dài không vượt quá 10^5 .

Output

- Ghi một số là vị trí xuất hiện thứ k của X trong S , nếu không tồn tại ghi số 0.

Ràng buộc:

- Có 40% số test ứng với 40% số điểm của bài thỏa mãn: độ dài xâu X bằng 1;
- 40% số test khác ứng với 40% số điểm của bài thỏa mãn: $k = 1$;
- 20% số test còn lại ứng với 20% số điểm của bài thỏa mãn: không có ràng buộc nào thêm.

Ví dụ:

Input	Output
3 7 2 2	8

Input	Output
3 5 2 00	3

Truyền tin

Có n người đánh số từ 1 đến n xếp thành một hàng và cùng nhau chơi trò chơi truyền tin. Người thứ i ($1 \leq i \leq n$) có độ trễ khi truyền tin là d_i , khi đó độ trễ người thứ i truyền tin cho người thứ j ($1 \leq i \leq j \leq n$) được tính bằng $D(i, j) = \max\{d_i, d_{i+1}, \dots, d_j\}$.

Người quản trò muốn tìm ra k ($1 \leq k \leq n$) người chơi để tổng độ trễ liên lạc là nhỏ nhất. Một cách hình thức, cần chọn ra k chỉ số $1 \leq i_1 < i_2 < \dots < i_k \leq n$ sao cho $w = \sum_{1 \leq x \leq y \leq k} D(i_x, i_y)$ là nhỏ nhất.

Yêu cầu: Tính giá trị w nhỏ nhất.

Input

- Dòng đầu chứa hai số nguyên n, k ($n \leq 10^4$);
- Dòng thứ hai chứa n số nguyên dương d_1, d_2, \dots, d_n ($d_i \leq 10^9$).

Output

- Ghi một số nguyên duy nhất là giá trị w tính được.

Ràng buộc:

- Có 20% số test ứng với 20% số điểm của bài thỏa mãn: $n \leq 20$;
- 30% số test khác ứng với 30% số điểm của bài thỏa mãn: $k = 3$ và $n \leq 10^4$;
- 30% số test khác ứng với 30% số điểm của bài thỏa mãn: $n \leq 500$;
- 20% số test còn lại ứng với 20% số điểm của bài thỏa mãn: $n \leq 10^4$.

Ví dụ:

Input	Output	Giải thích
4 3 1 2 2 1	10	Chọn ba người 1, 2, 4 có tổng độ trễ nhỏ nhất bằng: $D(1,2) + D(1,4) + D(2,4) + D(1,1) + D(2,2) + D(4,4) = 10$

Tập số

Cho số nguyên không âm n ở hệ cơ số 10, có không quá 1000 chữ số và không chứa các số 0 không có nghĩa ở đầu. Bằng cách xóa một hoặc một vài chữ số của n (nhưng không xóa hết tất cả các chữ số của n) ta nhận được những số mới. Số mới được chuẩn hóa bằng cách xóa các chữ số 0 vô nghĩa nếu có. Tập số nguyên D được xây dựng bằng cách đưa vào nó số n , các số mới khác nhau đã chuẩn hóa và khác n . Ví dụ, với $n = 101$ ta có thể nhận được các số mới như sau:

- Bằng cách xóa một chữ số ta có các số: 1 (từ 01), 11, 10;
- Bằng cách xóa hai chữ số ta có các số: 1, 1, 0;

Tập D nhận được từ n chứa các số $\{0, 1, 10, 11, 101\}$.

Yêu cầu: Cho số nguyên n và hai số nguyên không âm A, B . Hãy xác định số lượng số nằm trong $[A, B]$ có mặt trong tập D được tạo thành từ n .

Input

- Dòng 1: chứa số nguyên n , có không quá 1000 chữ số;
- Dòng 2: chứa số nguyên A , có không quá 1000 chữ số;
- Dòng 3: chứa số nguyên B , có không quá 1000 chữ số.

Output

- Một số nguyên là số lượng số tìm được.

Ví dụ:

Dữ liệu vào	Kết quả ra
101	3
1	
100	

Chú ý: 50% test $n, A, B < 10^{18}$