



Passwords

Link submit: <http://codeforces.com/problemset/problem/721/B>

Solution:

C++	https://ideone.com/6kGmKu
Java	https://ideone.com/r7q6l7
Python	https://ideone.com/AUkzRx

Tóm tắt đề:

Vanya cố gắng vào trang Codehorses. Anh ấy dùng n mật khẩu riêng biệt cho tất cả trang web, nhưng không nhớ đã dùng cái nào trong lúc đăng ký Codehorses.

Vanya nhập mật khẩu theo thứ tự độ dài không giảm, và nhập mật khẩu có cùng độ dài theo thứ tự tùy ý. Khi Vanya nhập đúng mật khẩu, anh ấy được phép vào trang web. Vanya sẽ không nhập bất kỳ mật khẩu nào hai lần.

Việc nhập mật khẩu bất kỳ sẽ tốn 1 giây. Nếu Vanya nhập sai mật khẩu k lần, thì 5 giây sau anh ta mới có thể thử lần tiếp theo. Tại mỗi thời điểm Vanya có thể nhập mật khẩu, anh ta sẽ thực hiện ngay.

Xác định Vanya cần bao nhiêu giây để vào Codehorses trong trường hợp tốt nhất (tốn ít giây nhất) và trường hợp xấu nhất (tốn nhiều giây nhất).

Input:

Dòng đầu tiên chứa hai số nguyên n và k ($1 \leq n, k \leq 100$) – số lượng mật khẩu của Vanya và số lần thử thất bại, bị chặn truy cập trang trong 5 giây sau khi thử k lần thất bại.

n dòng tiếp theo chứa các mật khẩu, trên mỗi dòng là chuỗi riêng biệt khác rỗng gồm chữ cái và chữ số latin. Mỗi mật khẩu có chiều dài không quá 100.

Dòng cuối chứa mật khẩu Codehorses của Vanya. Đảm bảo mật khẩu Codehorses của Vanya là 1 trong n mật khẩu của anh ta.

Output:

In hai số nguyên – thời gian (giây), Vanya cần để vào được Codehorses trong trường hợp tốt nhất và trường hợp xấu nhất.

Ví dụ:

5 2 cba abc bb1 abC ABC abc	1 15
4 100 11 22 1 2 22	3 4

Giải thích ví dụ:

Ví dụ 1: Tất cả mật khẩu có chiều dài giống nhau, Vanya có thể nhập mật khẩu đúng vào lần thử đầu tiên cũng như lần cuối. Nếu nhập nó vào lần thử đầu, anh ta tốn 1 giây. Vì vậy trong trường hợp tốt nhất, đáp án là 1. Mặt khác, nhập nó vào lần thử cuối, nhập 4 mật khẩu khác trước đó. Anh ta tốn 2 giây cho 2 mật khẩu đầu, sau đó đợi 5 giây khi thử sai 2 lần. Sau đó, tốn thêm 2 giây để nhập 2 mật khẩu sai, chờ 5 giây lần nữa, cuối cùng, nhập mật khẩu đúng tốn thêm 1 giây. Tóm lại trường hợp xấu nhất anh ta tốn 15 giây.

Ví dụ 2: Không có cách nào nhập mật khẩu để việc truy cập vào trang bị chặn. Mật khẩu yêu cầu có độ dài 2, Vanya nhập tất cả mật khẩu chiều dài 1, tốn hết 2 giây. Sau đó, trong trường hợp tốt nhất, anh ta nhập mật khẩu đúng ngay thì đáp án cho trường hợp này là 3, nhưng trong trường hợp xấu nhất, anh ta nhập mật khẩu có độ dài 2 sai và sau đó đúng, thì tốn 4 giây.

Hướng dẫn giải:

Theo cách nhập mật khẩu của Vanya, các mật khẩu có chiều dài nhỏ nhất sẽ được nhập trước, rồi đến các mật khẩu có chiều dài nhỏ thứ hai, cứ tiếp tục như vậy. Đến khi chiều dài bằng chiều dài mật khẩu đúng, thì trường hợp tốt sẽ nhập đúng ngay lần đầu tiên, trường hợp xấu sẽ nhập đúng sau khi đã nhập tất cả mật khẩu cùng chiều dài.

Vậy với n mật khẩu, ta sẽ đi đếm, ứng với mỗi chiều dài thì có bao nhiêu chuỗi cùng chiều dài đó, lưu lại thông tin này.

Sau đó, để tìm thời gian cho trường hợp tốt nhất, đi tính tổng số mật khẩu có chiều dài nhỏ hơn chiều dài mật khẩu đúng, đây cũng là thời gian dành để nhập các mật khẩu. Sau k lần nhập sai thì bị chặn 5 giây, nên ta lấy tổng vừa tìm được chia cho k ra được số lần bị chặn, rồi nhân cho 5, ta sẽ biết được tổng số thời gian bị chặn. Ta lấy tổng số mật khẩu tìm được trước đó cộng

với tổng thời gian bị chặn sẽ biết được tổng thời gian nhập các mật khẩu sai có chiều dài nhỏ hơn chiều dài mật khẩu đúng.

Thực hiện tương tự để tìm thời gian cho trường hợp xấu nhất, nhưng chỗ tính tổng số mật khẩu, ta cộng thêm số mật khẩu cùng chiều dài với mật khẩu đúng, nhớ trừ đi 1 (trừ mật khẩu đúng).

Lấy kết quả ở cả hai trường hợp ta đã tính được cộng thêm 1 (thời gian nhập mật khẩu đúng) là ra đáp án cuối cùng.

Độ phức tạp: $O(n + \text{length}(s))$ với n là số lượng chuỗi nhập và $\text{length}(s)$ là độ dài chuỗi s là chuỗi mật khẩu đúng.

Big-O Coding