Bedao Mini Contest 07 - RATE

Time Limit: 1.0s **Memory Limit:** 256M

Cho một xâu S, mỗi kí tự trong S là chữ cái Latin thường, và khi quy đổi những chữ cái này thành những thứ tự ASCII, và từ những thứ tự đó chuyển qua hệ nhị phân ta được một ma trận như sau:

- Số 0 là những ô không đi dược
- Số 1 là những ô đi được
- Kí tự # biểu thị cho kết thúc một hàng trong ma trận

Yêu cầu: xác định độ dài hàng, cột và tỉ lệ của các ô đi được với các ô không đi được sau khi dựng được ma trận từ xâu S

Input

- Dòng đầu tiên gồm số nguyên l là độ dài của xâu S $(1 \le l \le 10^6)$
- Dòng thứ hai gồm duy nhất một xâu S gồm các kí tự là chữ cái Latin thường hoặc là kí tự #, luôn đảm bảo rằng dữ liệu được cho sẽ không có nhiều dấu # liền kề nhau mà mỗi dấu # trong xâu S sẽ được cách nhau ít nhất là 1 kí tự và sẽ luôn không tồn tại kí tự # ở cuối của xâu S

Output

- ullet Dòng đầu tiên yêu cầu in ra số n và m lần lượt là số hàng và số cột của ma trận được tạo ra từ xâu S
- Dòng thử hai yêu cầu in ra tỉ lệ của các ô đi được với các ô không đi được, đáp án được in ra đúng 9 chữ số sau dấu phẩy.

Sample Input

4 ab#a

Sample Output

2 7

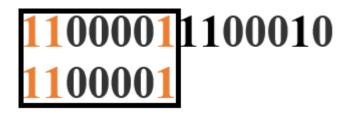
0.750000000

Note

- ullet Kí tự đầu tiên là a có mã ASCII là 97, đổi sang hệ nhị phân ta được 1100001
- Kí tự thứ hai là b có mã ASCII là 98, đổi sang hệ nhị phân ta được 1100010

- Kí tự thứ ba là # biểu thị cho kết thúc một hàng
- ullet Kí tự thứ tư là a có mã ASCII là 97, đổi sang hệ nhị phân ta được 1100001

Vậy tổng quát, ma trận sau khi đổi từ xâu S sang hệ nhị phân của ta là:



Số hàng của ta là n=2, số cột của ta là m=7 do ta sẽ lấy hàng có số ít phần tử liên tiếp nhất đầu tiên

Xét ma trận 2×7 được khoanh trong hình, ta có:

Số ô đi được là 6 và số ô không đi được là 8, vậy tỉ lệ của số ô đi được và số ô không đi được là: $\frac{6}{8}=0.750000000$