Cặp xâu ký tự

Giới hạn thời gian: 1.0s **Giới hạn bộ nhớ:** 256M

Bài 1. Cặp xâu ký tự Tên file: string.cpp

Bài toán đặt ra là cho trước các cặp xâu ký tự, đều chỉ bao gồm các chữ cái viết thường, hay xác định xem mỗi cặp xâu có được tạo ra từ cùng một bộ ký tự giống nhau hay không.

Chú ý: các ký tự lặp cũng phải được xem xét. Ví dụ: “abc”và “aabbbcccc” không được xem là tạo nên bởi cùng một bộ ký tự vì xâu thứ 2 các chữ cái được lặp lại nhiều lần hơn.

INPUT

- Gồm nhiều bộ test, mỗi bộ test gồm hai dòng ghi hai xâu cần kiểm tra, chỉ bao gồm các ký tự viết thường và không quá 1000 ký tự. Input kết thúc khi gặp hai dòng ghi chữ END.

OUTPUT

- Với mỗi bộ test, ghi ra màn hình thứ tự bộ test và kết quả bài toán (**same** hoặc **different**).

Ví dụ:

INPUT	OUTPUT
testing intestg abc aabbbcccc abcabcbcc aabbbcccc abc xyz END END	same different same different

Xếp domino

Giới hạn thời gian: 1.0s **Giới hạn bộ nhớ:** 256M

Bài 2. Xếp domino **Tên file: domino.cpp**

Alice có một lưới với 2 hàng và n cột. Cô ấy che hoàn toàn lưới bằng cách sử dụng n quân domino có kích thước 1 × 2 - Alice có thể đặt chúng theo chiều dọc hoặc chiều ngang, và mỗi ô phải được bao phủ bởi duy nhất một quân domino.

Bây giờ, cô ấy quyết định hiển thị một hàng của lưới cho Bob. Giúp Bob và tìm ra hàng còn lại của lưới trông như thế nào!

Input: domino.inp

- Dòng đầu tiên chứa một số nguyên t ($1 \le t \le 5000$) - số lượng test. Tiếp theo là mô tả các test:
  - Dòng đầu tiên của mỗi test chứa một số nguyên n ($1 \le n \le 100$) - chiều rộng của lưới.
  - Dòng thứ hai của mỗi trường hợp thử nghiệm chứa một chuỗi s bao gồm n ký tự, mỗi ký tự là L, R, U hoặc D, đại diện cho nửa trái, phải, trên hoặc dưới của quân cờ domino. Chuỗi này đại diện cho một trong các hàng của lưới.

Output: domino.out

- Đối với mỗi test, xuất một chuỗi là hàng còn lại của lưới. Nếu có nhiều câu trả lời, hãy in câu trả lời bất kỳ.

Ví dụ:

domino.inp	domino.out
4 1 U 2 LR 5 LRDLR 6 UUUUUU	D LR LRULR DDDDDD
domino.inp	domino.out
4 (đọc bộ test) 1(kích thước của bảng) U ô phía dưới của bảng 2 LR 5 LRDLR 6 UUUUUU	D LR LRULR DDDDDD

Dãy ngoặc

Giới hạn thời gian: 1.0s **Giới hạn bộ nhớ:** 256M

Bài 3. Dãy ngoặc Tên file: NGOAC.CPP

Người ta định nghĩa một dãy ngoặc đúng theo đệ quy như sau:

- Xâu A là xâu rỗng là một dãy ngoặc đúng

- Nếu xâu A là dãy ngoặc đúng thì (A) cũng là dãy ngoặc đúng

- Nếu xâu A và xâu B là dãy ngoặc đúng thì AB cũng là dãy ngoặc đúng.

Còn những xâu chứa những ký tự khác "(" và ")" được gọi là xâu không hợp lệ.

Ví dụ:

S = "(A()B)" là dãy "KHONG HOP LE" vì chứa chữ cái A và B

S = "()()(())" là dãy ngoặc "DUNG"

S = "()())" là dãy ngoặc "KHONG DUNG"

Cho xâu S. Hãy kiểm tra xâu S là dãy ngoặc "DUNG", "KHONG DUNG" hay là xâu "KHONG HOP LE"

INPUT: NGOAC.INP:

- Một xâu S chứa tối đa 10^6 phần tử

OUTPUT: NGOAC.OUT:

- Thông báo "KHONG HOP LE" nếu xâu không hợp lệ
- Thông báo "DUNG" nếu xâu đúng
- Thông báo "KHONG DUNG" nếu xâu không đúng

Ví dụ:

INPUT	**OUTPUT**
(A()B)	KHONG HOP LE
()()(())	DUNG
((())	KHONG DUNG

Giới hạn thời gian: 1.0s **Giới hạn bộ nhớ:** 256M

Bài 4. Xâu đối xứng tên file: MAXPALIN.CPP

Xâu đối xứng là xâu có độ dài ít nhất bằng 1, và không thay đổi khi đọc từ trái sang phải hoặc từ phải sang trái.

Cho một xâu ký tự S (chỉ gồm các chữ cái In thường) có độ dài tối đa là 5000 ký tự.

Hãy tìm một xâu con liên tiếp đối xứng dài nhất của S.

Input: maxpalin.inp

- Dòng đầu tiên chứa số nguyên n ($1 \le n \le 5000$) – độ dài của xâu S
- Dòng thứ hai chứa xâu S – chỉ gồm các chữ cái in thường

Output: maxpalin.out

- Đưa ra độ dài của xâu con liên tiếp đối xứng dài nhất

Ví dụ:

maxpalin.inp	maxpalin.out
32 fhj**abgthettiyryittehtgba**yutuyioe	21

Di chuyển ngoặc

Giới hạn thời gian: 1.0s **Giới hạn bộ nhớ:** 256M

Bài 5. Di chuyển ngoặc Tên file: MBRACKETS.CPP

Bạn cho trước một dãy ngoặc độ dài n, trong đó n là chẵn gồm n/2 ký tự mở ngoặc và n/2 ký tự đóng ngoặc.

Trong một lần di chuyển, bạn có thể chọn chính xác một dấu ngoặc và di chuyển nó đến đầu chuỗi hoặc đến cuối chuỗi (tức là bạn chọn một số vị trí i, loại bỏ ký tự thứ i của s và chèn nó vào trước hoặc sau khi tất cả ký tự còn lại của s).

Nhiệm vụ của bạn là tìm số lần di chuyển tối thiểu cần thiết để có được chuỗi dấu ngoặc đúng từ s. Có thể chứng minh rằng câu trả lời luôn tồn tại dưới các ràng buộc đã cho.

Nhắc lại chuỗi dấu ngoặc đúng là:

- "()" là chuỗi dấu ngoặc đúng;
- nếu s là chuỗi ngoặc đúng thì "(" + s + ")" là chuỗi ngoặc đúng;
- nếu s và t là chuỗi ngoặc đúng thì s + t là chuỗi ngoặc đúng.

Ví dụ: "() ()", "(()) ()", "(())" và "()" là các chuỗi dấu ngoặc đúng, nhưng ") (", "() (" và ")))" không.

INPUT: MBRACKETS.INP

- Dòng đầu tiên chứa số nguyên dương t ($1 \le t \le 1000$) – số lượng test thử:
  - Dòng đầu tiên của test thử chứa số n (chẵn và $2 \le n \le 1000$) – độ dài dãy ngoặc
  - Dòng tiếp theo là dãy ngoặc, trong đó n/2 mở ngoặc và n/2 đóng ngoặc

OUTPUT: MBRACKETS.OUT

- Đối với mỗi trường hợp kiểm tra, in câu trả lời - số lần di chuyển tối thiểu cần thiết để có được chuỗi dấu ngoặc đúng từ s. Có thể chứng minh rằng câu trả lời luôn tồn tại dưới các ràng buộc đã cho.

Ví dụ:

MBRACKETS.INP	MBRACKETS.OUT
4 **2**)(4 ()() 8 ())()()(10)))((((())	1 0 1 3

Giới hạn thời gian: 1.0s **Giới hạn bộ nhớ:** 256M

Bài 6. Bội chung nhỏ nhất của xâu Tên file: STRLCM.CPP

Hãy định nghĩa một phép nhân giữa một chuỗi ***a*** và một số nguyên dương ***x***: ***a*** * ***x*** là một chuỗi kết quả của việc ghép xâu ***a*** ***x*** lần.

Ví dụ: "abc" * 2 = "abcabc", "a" * 5 = "aaaaa".

Chuỗi ***a*** chia hết cho chuỗi ***b*** nếu tồn tại số nguyên ***x*** sao cho ***b*** * ***x*** = ***a***. Ví dụ: "abababab" chia hết cho "ab", nhưng không chia hết cho "ababab" hoặc "aa".

LCM của hai chuỗi ***s*** và ***t*** (được định nghĩa là LCM (***s***, ***t***)) là chuỗi không rỗng ngắn nhất chia hết cho cả ***s*** và ***t***.

Bạn được cung cấp hai chuỗi ***s*** và ***t***. Tìm LCM (***s***, ***t***) hoặc thông báo là không tìm được. Có thể chỉ ra rằng nếu tồn tại LCM (s, t) thì nó là duy nhất.

Input: **STRLCM.INP**

- Dòng đầu tiên chứa một số nguyên ***q*** ($1 \le q \le 2000$) – số truy vấn
- Mỗi trường hợp kiểm tra bao gồm hai dòng, chứa các chuỗi ***s*** và ***t*** ($1 \le |s|, |t| \le 200$). Mỗi ký tự trong mỗi chuỗi này là '***a***' hoặc '***b***'.

Output: STRLCM.OUT

- Đối với mỗi truy vấn, in LCM (s, t) nếu nó tồn tại; nếu không, in -1. Có thể chỉ ra rằng nếu tồn tại LCM (s, t) thì nó là duy nhất.
- Ví dụ:

STRLCM.INP	STRLCM.OUT
3 baba ba aa aaa aba ab	baba aaaaaa -1

Giới hạn thời gian: 1.0s **Giới hạn bộ nhớ:** 256M

Bài 7. Độ đo Tên file: AD.CPP

Hai xâu ký tự được gọi là đảo của nhau nếu ta có thể hoán vị các ký tự của xâu để được xâu còn lại. Ví dụ: xâu "occurs" là đảo của xâu "succor", tuy nhiên "dear" không phải là đảo của xâu "daerd" (vì chữ "d" xuất hiện 2 lần trong "deard" còn "dear" thì chỉ xuất hiện 1 lần).

Độ đo giữa 2 xâu ký tự là số ký tự ít nhất cần phải xóa (trên cả 2 xâu) để 2 xâu còn lại là đảo của nhau. Ví dụ: độ đo giữa hai xâu "sleep" và "leap" là 3, độ đo giữa hai xâu "dog" và "cat" là 6.

Yêu cầu: Hãy tìm độ đo giữa hai xâu cho trước.

Dữ liệu vào: file **AD.INP:**

- Gồm 2 dòng, mỗi dòng chứa một xâu ký tự chỉ gồm các chữ cái tiếng Anh thường, mỗi dòng chứa không quá 1 triệu ký tự.

Kết quả: file **AD.OUT:**

- Một số nguyên duy nhất là độ đo giữa hai xâu cho trước.

Ví du:

AD.INP	AD.OUT
begin end	4

Giới hạn thời gian: 1.0s **Giới hạn bộ nhớ:** 256M

Bài 8. Đếm từ Tên file: CWORD.CPP

Vì không có bố mẹ từ bé nên hằng ngày ngoài thời gian đến trường Chí phải đi chăn bò thuê kiếm tiền. Vào một ngày nọ, Chí đã gặp một dòng văn bản hấp dẫn được khắc vào một tảng đá lớn ở giữa vùng chăn thả bò yêu thích của mình. Ý nghĩa của dòng văn bản dường như là từ một ngôn ngữ cổ xưa bí ẩn liên quan đến một bảng chữ cái chỉ gồm ba ký tự C, O, và W. Mặc dù Chí không thể giải mã văn bản nhưng COW là mẫu từ yêu thích của Chí, và cậu tự hỏi có bao nhiêu lần COW xuất hiện trong dòng văn bản.

Chí không phiền lòng nếu có những kí tự khác xen kẽ trong COW, miễn rằng các kí tự xuất hiện theo thứ tự đúng là C, O, W. Chí cũng không ngại nếu các lần xuất hiện khác nhau của COW có chung một số chữ cái. Ví dụ, COW xuất hiện một lần trong CWOW, hai lần trong CCOW, và tám lần trong CCOOWW.

Em hãy vui lòng giúp Chí đếm xem có bao nhiêu lần COW xuất hiện trong dòng văn bản đã gặp.

Dữ liệu: Vào từ file văn bản **cword.inp**:

Dòng đầu tiên gồm một số nguyên duy nhất $N \leq 10^5$.

Dòng thứ hai chứa một chuỗi gồm N ký tự C, O, hay W.

Kết quả: Ghi ra file văn bản **cword.out** chỉ một số nguyên duy nhất là số lần COW xuất hiện như một dãy con (các kí tự không nhất thiết phải liên tục) trong chuỗi input.

Ví dụ:

cword.inp	cword.out
6 COOWWW	6

Giới hạn thời gian: 1.0s **Giới hạn bộ nhớ:** 256M

Bài 9. Tên file — Tên file: FileName.cpp

Khi bạn cố gắng gửi một tệp trong mạng xã hội "Codehorses", bạn đã gặp phải một sự cố không mong muốn. Nếu tên của tệp chứa ba hoặc nhiều hơn "x" (chữ cái Latinh viết thường "x") liên tiếp, hệ thống cho rằng nội dung tệp không tương ứng với chủ đề mạng xã hội. Trong trường hợp này, tệp không được gửi và thông báo lỗi được hiển thị.

Xác định số ký tự tối thiểu để loại bỏ khỏi tên tệp để sau đó tên không chứa "xxx" như một chuỗi con. In ra 0 nếu tên tệp ban đầu không chứa chuỗi con bị cấm "xxx".

Bạn có thể xóa các ký tự ở các vị trí tùy ý (không nhất thiết phải liên tiếp). Nếu bạn xóa một ký tự, thì độ dài của chuỗi sẽ giảm đi 1. Ví dụ: nếu bạn xóa ký tự ở vị trí 2 khỏi chuỗi "exxxii", thì chuỗi kết quả là "exxii".

Input: FileName.inp:

- Dòng đầu là số n ($1 \leq n \leq 1000$) – độ dài tên file
- Dòng thứ hai chứa xâu S chỉ chứa các chữ cái LATINH in thường là tên của file cần gửi

Output: FileName.out: Số lượng ký tự cần xóa để S không còn 3 ký tự "x" liên tiếp.

Ví dụ:

FileName.inp	FileName.out
6 xxxiii	1
5 xxoxx	0
10 xxxxxxxxxx	8

Giới hạn thời gian: 1.0s **Giới hạn bộ nhớ:** 256M

Bài 10. GIẢI MÃ CHỮ VIẾT CỦA NGƯỜI MAIA Tên file: WRITING.CPP

Công việc giải mã chữ viết của người MAIA là khó khăn hơn người ta tưởng nhiều. Trải qua hơn 200 năm mà người ta vẫn hiểu rất ít về các chữ viết này. Chỉ trong 3 thập niên gần đây do công nghệ phát triển việc giải mã này mới có nhiều tiến bộ.

Chữ viết Maia dựa trên các ký hiệu nhỏ gọi là nét vẽ, mỗi nét vẽ tương ứng với một âm giọng nói. Mỗi từ trong chữ viết Maia sẽ bao gồm một tập hợp các nét vẽ như vậy kết hợp lại với nhiều kiểu dáng khác nhau. Mỗi nét vẽ có thể hiểu là một ký tự ta hiểu ngày nay.

Một trong những vấn đề lớn khi giải mã chữ Maia là thứ tự đọc các nét vẽ. Do người Maia trình bày các nét vẽ này không theo thứ tự phát âm, mà theo cách thể hiện của chúng. Do vậy nhiều khi đã biết hết các nét vẽ của một từ rồi nhưng vẫn không thể tìm ra được chính xác cách ghi và đọc của từ này.

Các nhà khảo cổ đang đi tìm kiếm một từ đặc biệt W. Họ đã biết rõ tất cả các nét vẽ của từ này nhưng vẫn chưa biết các cách viết ra của từ này. Vì họ biết có các thí sinh IOI'06 sẽ đến nên muốn sự trợ giúp của các sinh viên này. Họ sẽ đưa ra toàn bộ g nét vẽ của từ W và dãy S tất cả các nét vẽ có trong hang đá cổ. Bạn hãy giúp các nhà khảo cổ tính xem có bao nhiêu khả năng xuất hiện từ W trong hang đá.

Yêu cầu: Hãy viết chương trình, cho trước các ký tự của từ W và dãy S các nét vẽ trong hang đá, tính tổng số khả năng xuất hiện của từ W trong dãy S, nghĩa là số lần xuất hiện một hoán vị các ký tự của dãy g ký tự trong S.

Các ràng buộc

- $1 \le g \le 3.000$, số nét vẽ trong W
- $g \le |S| \le 3.000.000$, $|S|$ là số các nét vẽ của dãy S

INPUT: WRITING.INP:

- Dòng 1: chứa 2 số g và $|S|$ cách nhau bởi dấu cách.
- Dòng 2: chứa g ký tự liền nhau là các nét vẽ của từ W. Các ký tư hợp lệ là 'a'-'z' và 'A'-'Z'. Các chữ in hoa và in thường là khác nhau.
- Dòng 3: Chứa $|S|$ ký tự là dãy các nét vẽ tìm thấy trong hang. Các ký tư hợp lệ là 'a'-'z' và 'A'-'Z'. Các chữ in hoa và in thường là khác nhau.

OUTPUT: WRITING.OUT:

- Chứa đúng 1 số là khả năng xuất hiện của từ W trong dãy S.

Ví dụ:

WRITING.INP	WRITING.OUT
4 11 cAda AbrAcadAbRa	2