

# Bài tập Xử lý chuỗi – Tuần 1

## Bài 1 – Chấm thi

Trong các cuộc thi lập trình một vấn đề quan trọng mà các thí sinh chưa có kinh nghiệm thường không lưu ý là kết quả của họ phải chính xác hoàn toàn với kết quả của giám khảo. Thường các thí sinh thêm các khoảng trắng vào kết quả của mình vì họ thấy là điều này làm cho kết quả được thể hiện đẹp mắt hơn. Yêu cầu trong thực tế là kết quả phải khớp hoàn toàn từng kí tự một với kết quả của giám khảo. Nếu kết quả có thêm khoảng trắng thì khi chấm sẽ nhận được đánh giá 'Loi Dinh dang'. Nếu kết quả vẫn không khớp sau khi bỏ qua khoảng trắng, đánh giá 'Tra loi Sai' sẽ được đưa ra. Đánh giá 'Dung' sẽ được đưa ra khi kết quả khớp hoàn toàn. Trong bài tập này, bạn sẽ phải quyết định đánh giá cho những lần chạy dựa vào kết quả của lý sinh và giám khảo.

**Dữ liệu:** Vào từ file CHAMTHI.INP. Dòng đầu tiên chứa một số nguyên dương  $t < 20$  trong đó  $t$  cho biết số lượng bộ dữ liệu. Mỗi bộ chứa hai dòng. Dòng đầu tiên là kết quả của thí sinh và dòng thứ hai là kết quả của giám khảo. Mỗi dòng chứa từ một đến 20 kí tự. Kết quả của thí sinh bao gồm chữ cái và khoảng trắng. Kết quả của giám khảo chỉ chứa khoảng trắng.

**Kết quả:** Với mỗi bộ dữ liệu, xuất ra một dòng kết quả vào file CHAMTHI.OUT. Mỗi dòng sẽ chứa số thứ tự bộ dữ liệu và kết luận đánh giá. Xem kết quả ví dụ để biết định dạng chính xác.

**Ví dụ:**

CHAMTHI.INP	CHAMTHI.OUT
3 dung dung PhanBietHoaThuong phanbiethoathuong khong khoang trang khongkhoangtrang	Dung Tra loi Sai Loi Dinh dang

## Bài 2 – Tìm mật khẩu

Thông thường khi mã hoá một thông điệp người nhận sẽ biết trước mật khẩu để giải mã tuy nhiên cách này có vấn đề vì mật khẩu sẽ bị cố định và có thể bị đoán ra. Một cách sáng tạo để gửi mật khẩu ẩn trong thông điệp mã hoá. Điểm thú vị là người nhận thông điệp chỉ cần phải biết độ dài của mật khẩu và tìm mật khẩu trong thông điệp nhận được.

Một mật khẩu độ dài  $N$  có thể được tìm thấy bằng cách tìm đoạn chuỗi con  $N$  kí tự xuất hiện nhiều lần nhất. Sau khi tìm được mật khẩu, tất cả các chuỗi con giống với mật khẩu sẽ được xoá khỏi đoạn văn bản và văn bản giờ đây có thể được giải mã.

Công việc của bạn là viết một chương trình nhận vào kích thước của mật khẩu và thông điệp mã hoá, hãy xác định mật khẩu theo cách như trên. Xét ví dụ kích thước mật khẩu là ba ( $N=3$ ) và thông điệp văn bản chỉ là **'baababacb'**. Mật khẩu sẽ là **aba** bởi vì đây là chuỗi con có 3 kí tự xuất hiện nhiều lần nhất trong toàn bộ văn bản (nó xuất hiện hai lần) trong khi sáu chuỗi con khác chỉ xuất hiện một lần (**baa** ; **aab** ; **bab** ; **bac** ; **acb**).

**Dữ liệu:** File dữ liệu MATKHAU.INP bao gồm nhiều bộ dữ liệu, mỗi bộ bao gồm một dòng chứa kích thước mật khẩu,  $0 < N \leq 10$ , và sau đó là văn bản biểu diễn thông điệp mã hoá. Văn bản chỉ chứa các kí tự viết thường.

**Kết quả:** Xuất ra file MATKHAU.OUT chuỗi mật khẩu tìm được ứng với từng bộ dữ liệu trên, mỗi mật khẩu một dòng.

Ví dụ:

MATKHAU.INP	MATKHAU.OUT
3 baababacb	aba

## Bài 3 – Ô chữ

Một ô chữ được tạo ra bằng cách in một cặp từ, một ngang và một dọc, sao cho chúng có 1 kí tự chung. Một *ô chữ chính* có kí tự chung nằm ở vị trí gần nhất với đầu của từ ngang và kí tự này cũng nằm càng gần đầu của từ dọc càng tốt. Do đó DALAT và BINHTHUAN sẽ cắt ở kí tự 'A' đầu tiên của mỗi từ, còn BINHTHUAN và DALAT sẽ cắt ở kí tự 'T'. *Ô chữ kép* sử dụng hai cặp từ được bố trí sao cho hai từ ngang nằm trên cùng một dòng và mỗi cặp hình thành một ô chữ chính.

Hãy viết một chương trình đọc một tập bốn từ và trình bày chúng dưới dạng ô chữ kép.

**Dữ liệu:** Vào từ file OCHU.INP. Dữ liệu bao gồm một số dòng, mỗi dòng chứa bốn từ (hai cặp). Một từ gồm 1 đến 10 kí tự viết hoa và cách nhau bởi ít nhất một khoảng trắng. File sẽ kết thúc bởi một dòng chứa dấu '#'.  
**Kết quả:** Kết quả xuất ra file OCHU.OUT bao gồm một dãy các ô chữ kép như định nghĩa ở trên. Các từ ngang cách nhau đúng 3 khoảng trắng. Nếu không thể tạo ra hai ô chữ, xuất ra thông báo 'Khong tao duoc o chu'. Xuất 1 dòng trắng giữa các tập kết quả.

**Ví dụ:**

OCHU.INP	OCHU.OUT
SONDOONG QUANGBINH SAPA LAOCAI	Q
VIENTHONG LAPTOP PHANMEM VIRUS	U
#	A                      L
	SONDOONG      SAPA
	G                      O
	B                      C
	I                      A
	N                      I
	H
	Khong tao duoc o chu

## Bài 4 – Hàng đợi XOR

Cho hàng đợi **qa** quản lý các số nguyên, ban đầu chỉ chứa một số nguyên  $a_0 = 0$ . Người ta lần lượt bổ sung vào hàng đợi các số nguyên  $a_1, a_2, \dots, a_n$ . Mỗi số nguyên  $a_i$  khi vào hàng đợi **qa** sẽ làm xuất hiện một số nguyên dẫn xuất  $b_i$  vào hàng đợi **qb** theo công thức  $b_0 = a_0, b_i = a_0 \text{ xor } a_1 \text{ xor } \dots \text{ xor } a_i, i = 0 \div n$  (xor là phép ^ trong C/C++). Trong quá trình khai thác dữ liệu người ta có nhu cầu thực hiện các phép xử lý:

**POP** – xóa phần tử đứng đầu hàng đợi **qa**, điều này kéo theo phần tử tương ứng trong **qb** cũng sẽ bị xóa khỏi **qb** đồng thời làm thay đổi giá trị các phần tử trong hàng đợi **qb** vì trong công thức tính toán của  $b_i$  sẽ không có sự tham gia của phần tử vừa bị xóa khỏi **qa**,

**PUSH x** – bổ sung vào cuối hàng đợi **qa** phần tử **x**, một phần tử mới sẽ xuất hiện trong **qb** theo công thức tính đã nêu,

**COUNT u v** – đếm số phần tử **y** trong **qb** thỏa mãn điều kiện  $u \leq y \leq v$ . Hãy lập trình đưa ra kết quả với mỗi phép **COUNT**.

**Dữ liệu:** Vào từ file văn bản XORQUEUE.INP: Dòng đầu tiên chứa số nguyên  $n$  ( $1 \leq n \leq 2 \times 10^5$ ), Dòng thứ 2 chứa  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i \leq 10^{15}, i = 1 \div n$ ), Dòng thứ 3 chứa số nguyên  $m$  ( $1 \leq m \leq 10^5$ ), Mỗi dòng trong  $m$  dòng tiếp theo chứa thông tin theo quy cách đã nêu về một phép xử lý cần thực hiện. Các số nguyên trong phép xử lý (nếu có) đều nằm trong đoạn  $[0, 10^{15}]$ .

**Kết quả:** Đưa ra file văn bản XORQUEUE.OUT kết quả tìm được đối với mỗi phép **COUNT**, mỗi kết quả đưa ra trên một dòng dưới dạng số nguyên.

**Ví dụ:**

XORQUEUE.INP	XORQUEUE.OUT
5	4
2 14 9 12 6	5
6	6
POP	5
COUNT 1 14	
PUSH 1	
COUNT 1 14	
COUNT 0 16	
COUNT 3 15	