

**ÔN THI
HSG****LUYỆN TẬP****ĐỀ THI THỬ HSG SỐ 02 (BẢNG A - THCS)****Môn: Tin học***Tổng hợp đề ôn thi học sinh giỏi chất lượng***TỔNG QUAN ĐỀ THI**

STT	Tên bài	Tên tệp chương trình	Tên tệp dữ liệu vào	Tên tệp kết quả ra	Điểm
1	Trâu già cỏ non	SBB.*	SBB.INP	SBB.OUT	5,0
2	Giải và trọng	UZAW.*	UZAW.INP	UZAW.OUT	5,0
3	Giao thức	ITC.*	ITC.INP	ITC.OUT	4,0
4	Triển lãm kép	LEGO.*	LEGO.INP	LEGO.OUT	3,0
5	Đón khách	PUP.*	PUP.INP	PUP.OUT	3,0

Chú ý: Dấu * được thay thế bởi CPP, PY của ngôn ngữ lập trình được sử dụng tương ứng là C/C++ hoặc Python.

Câu I. Trâu già cỏ non (5,0 điểm)

Ở làng Vũ Đại có một chú trâu già. Nghe đồn trên đỉnh ngọn đồi sương mù có một bãi "cỏ non" cực kỳ tươi tốt, tên là Hải Vân. Quãng đường từ chân đồi bỏ lên bãi cỏ non dài đúng V mét.

Ban ngày, trâu già dồn hết sức bình sinh nên leo lên được A mét. Nhưng về đêm, sương xuống làm đường trơn trượt, cộng thêm sức yếu nên trâu ngủ quên và bị tuột dốc xuống B mét. Quá trình cứ lặp lại như vậy mỗi ngày. Biết rằng nếu trong ngày trâu chạm được tới đỉnh đồi (hoặc vượt qua đỉnh) thì nó sẽ lập tức ăn được cỏ non và không bị tuột xuống nữa.

Yêu cầu: Hãy tính xem sau bao nhiêu ngày thì trâu già vươn tới được bãi cỏ non?

Dữ liệu vào từ file văn bản SBB.INP

- Gồm một dòng duy nhất chứa ba số nguyên A, B, V cách nhau bởi dấu cách ($1 \leq B < A \leq 10^{18}, 1 \leq V \leq 10^{18}$).

Kết quả ghi ra file văn bản SBB.OUT:

- In ra một số nguyên duy nhất là số ngày trâu già cần để ăn được cỏ non.

Ví dụ:

SBB.INP	SBB.OUT
5 1 10	3

Câu II. Giải và trọng (5,0 điểm)

Trong máy tính, để tiết kiệm bộ nhớ, người ta thường tìm cách nén dữ liệu. Trong việc nén văn bản, ta sử dụng một phương pháp đếm số lần xuất hiện liên tiếp của các ký tự. Quy ước nén được định nghĩa chi tiết như sau:

- Nhóm ký tự lặp lại:** Nếu một ký tự xuất hiện liên tiếp lớn hơn 1 lần, ta viết số lượng xuất hiện ngay trước ký tự đó. Số lượng này có thể là một số có nhiều chữ số (ví dụ: **15a** biểu diễn cho 15 ký tự a liên tiếp).
- Ký tự đơn lẻ:** Nếu một ký tự chỉ xuất hiện đúng 1 lần riêng lẻ, ta chỉ viết ký tự đó mà không cần ghi số đếm 1 ở trước (ví dụ: viết **b** thay vì **1b**).

Ví dụ: Với xâu ký tự: "**aaaabbb**" sẽ được nén lại thành xâu "**4a3b**". Với xâu ký tự "**aaab**" sẽ được nén lại thành "**3ab**".

Cho một xâu S gồm các ký tự thuộc tập '**a**' ... '**z**'. Gọi T là xâu nén của xâu S theo phương pháp được mô tả như trên. Xâu T gồm N ký tự thuộc tập các ký tự '**a**' ... '**z**', '**0**' ... '**9**'.

Mỗi ký tự trong xâu được quy định một mức trọng số tương ứng với vị trí của nó trong bảng chữ cái tiếng Anh. Cụ thể: trọng số của '**a**' là 1, '**b**' là 2, '**c**' là 3,..., '**z**' là 26.

Yêu cầu: Hãy giải nén xâu T để được xâu gốc S , đồng thời tính tổng trọng số của tất cả các ký tự có trong xâu S sau khi đã giải nén.

Dữ liệu vào từ file văn bản UZAW.INP:

- Một xâu T

Kết quả ghi ra file văn bản UZAW.OUT:

- Dòng đầu tiên in ra xâu ký tự S sau khi giải nén.
- Dòng thứ hai in ra một số nguyên là tổng trọng số của xâu S .
- Dữ liệu đảm bảo số lượng ký tự sau khi giải nén không quá 10^6 .

Ví dụ:

UZAW.INP	UZAW.OUT
4a3b	aaaabbbb 10

Câu III. Giao thức (4,0 điểm)

Vinh là một Hacker đang xâm nhập vào hệ thống lõi của tập đoàn công nghệ MegaCorp. Hệ thống này truyền tải dữ liệu thông qua một luồng gồm N gói tin liên tiếp. Mỗi gói tin được bảo vệ bởi một "chữ ký số" là một chuỗi ký tự (**String**).

Để bẻ khóa và trích xuất dữ liệu, Vinh phát hiện ra một lỗ hổng trong giao thức đồng bộ của chúng: Hai gói tin có thể triệt tiêu lớp bảo mật của nhau nếu chữ ký số của gói này chính là chuỗi đảo ngược của gói kia (ví dụ: gói **root** có thể ghép với gói **toor**, gói **sys** ghép với chính nó là **sys**).

Thuật toán đánh chặn của bạn hoạt động như sau:

Hệ thống của bạn quét luồng dữ liệu từ gói đầu tiên đến gói thứ N . Khi quét đến gói tin tại mốc thời gian i , phần mềm sẽ dò tìm ngược về quá khứ để tìm một gói tin ở mốc j ($j < i$) thỏa mãn:

1. Chữ ký của gói j là chuỗi đảo ngược của gói i .
2. Gói j chưa bị bẻ khóa (chưa từng ghép cặp với gói nào khác).
3. Gói j là gói gần i nhất có thể.

Nếu tìm thấy, phần mềm sẽ ép hai gói này ghép cặp với nhau. Lượng dữ liệu (tính bằng Megabyte) bạn trích xuất được từ mỗi lần bẻ khóa thành công bằng đúng độ trễ thời gian giữa chúng: $(i - j)$ MB. Kể từ đó, cả hai gói tin này hoàn toàn biến mất khỏi luồng. Nếu không tìm thấy, gói tin thứ i sẽ tạm thời bị lưu vào hàng đợi để chờ các gói phía sau.

Yêu cầu: Tính tổng lượng dữ liệu bạn có thể trích xuất được sau khi quét hết toàn bộ luồng N gói tin.

Dữ liệu vào từ file văn bản ITC.INP

- Dòng đầu tiên chứa số nguyên dương N ($1 \leq N \leq 10^5$) là số lượng gói tin.
- Dòng thứ hai chứa N chuỗi chữ ký số, mỗi chuỗi cách nhau một khoảng trắng.
- Độ dài mỗi chuỗi không vượt quá 10 ký tự, chỉ gồm các chữ cái in thường.

Kết quả ghi ra file văn bản ITC.OUT:

- In ra một số nguyên là tổng lượng dữ liệu trích xuất được.

Ràng buộc:

- 50% số test tương ứng với 50% số điểm có $n \leq 5000$.
- 50% số test tương ứng với 50% số điểm còn lại không có giới hạn gì thêm.

Ví dụ:

ITC.INP	ITC.OUT	Giải thích
4 net root toor ten	4	- Gói 1 (net) và 2 (root): Đưa vào hàng đợi. - Gói 3 (toor): Đảo ngược của nó là root. Tìm thấy gói 2 đang chờ. Ghép cặp (3, 2). Dữ liệu = $3 - 2 = 1$ MB. - Gói 4 (ten): Đảo ngược của nó là net. Có thể ghép với gói 1. Ghép (4, 1). Dữ liệu = $4 - 1 = 3$ MB. Tổng trích xuất: $1 + 3 = 4$ MB.

Câu IV. Triển lãm kép (3,0 điểm)

Trường mầm non SuperKids quyết định mở rộng quy mô bằng cách chia khu vực trưng bày lego thành hai phòng riêng biệt dành cho hai nhóm tuổi khác nhau. Có tất cả n mô hình lego đánh số từ 1 tới n , mô hình thứ i có kích thước là a_i .

Để đảm bảo tính thẩm mỹ và phù hợp với từng độ tuổi, ban tổ chức đặt ra quy tắc:

- *Phòng trưng bày 1 (dành cho trẻ nhỏ)*: Hai mô hình lego bất kỳ trong phòng này phải có kích thước chênh lệch nhau không quá K_1 .
- *Phòng trưng bày 2 (dành cho trẻ lớn)*: Hai mô hình lego bất kỳ trong phòng này phải có kích thước chênh lệch nhau không quá K_2 .
- *Lưu ý*: Tùy thuộc vào phương án sắp xếp, một trong hai phòng trưng bày hoàn toàn có thể bị bỏ trống (không chứa mô hình nào).

Yêu cầu: Hãy giúp ban tổ chức chọn ra một số lượng tối đa các mô hình lego để phân vào hai phòng trưng bày sao cho thỏa mãn các ràng buộc nêu trên (một mô hình chỉ được đặt ở tối đa một phòng).

Dữ liệu vào từ file văn bản LEGO.INP

- Dòng đầu tiên chứa ba số nguyên dương n, K_1, K_2 ($n \leq 10^5$; $K_1, K_2 \leq 10^9$).
- Dòng tiếp theo chứa n số nguyên dương $a_1, a_2, \dots, a_n$ ($a_i \leq 10^9$)

Kết quả ghi ra file văn bản LEGO.OUT:

- In ra một số nguyên duy nhất là số lượng mô hình lego tối đa có thể trưng bày.

Ràng buộc:

- 30% số test tương ứng với 30% số điểm có $n \leq 100$.
- 30% số test tương ứng với 30% số điểm tiếp theo có $n \leq 2000$.
- 20% số test tương ứng với 20% số điểm tiếp theo $K_1 = K_2$
- 20% số test tương ứng với 20% số điểm còn lại không có giới hạn gì thêm.

Ví dụ:

LEGO.INP	LEGO.OUT	Giải thích
9 2 5 3 4 5 7 9 1 10 11 100	7	- Chọn các mô hình có kích thước 3, 4 và 5 để đưa vào Phòng 1. Kích thước chênh lệch lớn nhất giữa hai mô hình bất kỳ trong phòng này là $5 - 3 = 2$ (thỏa mãn điều kiện $\leq K_1 = 2$). - Chọn các mô hình có kích thước 7, 9, 10 và 11 để đưa vào Phòng 2. Kích thước chênh lệch lớn nhất giữa hai mô hình bất kỳ trong phòng này là $11 - 7 = 4$ (thỏa mãn điều kiện $\leq K_2 = 5$). Tổng số mô hình trưng bày được là $3 + 4 = 7$.

Câu V. Đón khách (3,0 điểm)

Tại thành phố Hà Nội, có N bến xe được phân bố trên một tuyến đường du lịch khép kín hình vòng tròn. Theo chiều kim đồng hồ, bến xe thứ i liền kề với bến thứ $i + 1$ ($1 \leq i < N$), và đặc biệt bến thứ N sẽ liền kề với bến thứ 1. Tại bến thứ i , ban đầu có A_i hành khách đang chờ.

Hai nhà xe Hưng và Bình cùng xuất phát tại bến đầu tiên và di chuyển liên tục theo chiều kim đồng hồ. Cả hai xe có thể chạy quanh vòng tròn nhiều vòng, quá trình này chỉ dừng lại khi tất cả khách trên toàn tuyến đều đã được đón hết.

Quy tắc hoạt động của hai nhà xe như sau:

- Thời gian di chuyển giữa hai bến xe liên tiếp tiêu tốn 1 đơn vị thời gian.
- Tại một bến đang còn X người chờ, nhà xe có thể quyết định đón một lượng Y khách ($0 \leq Y \leq X$).
- Thời gian để thực hiện việc đón người là đúng Y đơn vị thời gian. Số khách này lập tức thuộc về nhà xe đó một cách vĩnh viễn, và số khách chờ tại bến giảm xuống còn $X - Y$.
- Nếu một nhà xe đón khách tại bến i (tức là chọn $Y > 0$), nhà xe đó sẽ bị cấm không được phép đón khách ở K bến tiếp theo mà họ đi qua.
- Nếu tại bất kỳ thời điểm nào cả hai nhà xe đến cùng một bến, Hưng luôn là người được quyền ưu tiên ra quyết định đón khách trước.

- Biết rằng cả Hưng và Bình đều chơi hoàn toàn tối ưu nhằm mục đích tối đa hóa tổng số lượng hành khách mà mình đón được.

Yêu cầu: Hãy tìm chênh lệch số khách giữa hai nhà xe khi quá trình kết thúc.

Dữ liệu vào từ file văn bản PUP.INP

- Dòng đầu tiên gồm số nguyên T ($1 \leq T \leq 10$) là số lượng testcase nhập vào:
- Dòng đầu tiên chứa hai số nguyên N và K ($3 \leq K \leq N \leq 10^6$).
- Dòng thứ hai chứa dãy N số nguyên $A_1, A_2, \dots, A_N$ ($1 \leq A_i \leq 10^9$) tương ứng với số hành khách ban đầu tại mỗi bến.

Kết quả ghi ra file văn bản PUP.OUT:

- Với mỗi testcase, in ra một số nguyên duy nhất là kết quả bài toán tương ứng với testcase đó.

Ràng buộc:

- Có 25% số test ứng với 25% số điểm của bài có $A_1 + A_2 + \dots + A_N \leq 20$.
- Có 25% số test ứng với 25% số điểm của bài có $N \leq 1000$.
- Có 25% số test ứng với 25% số điểm của bài có $A_1 = A_2 = A_3 = \dots = A_N$.
- Có 25% số test ứng với 25% số điểm của bài không có ràng buộc gì thêm.

Ví dụ:

PUP.INP	PUP.OUT	Giải thích
1 3 1 1 2 1	0	Hưng sẽ đón khách tại bến 1, 3 được 2 khách. Bình sẽ đón khách tại bến 2 được 2 khách. Chênh lệch lượng khách giữa hai nhà xe bằng $2 - 2 = 0$
2 1 2 5	2	Hưng sẽ đón khách tại bến 3 được 5 khách. Bình sẽ đón khách tại bến 2 được 2 khách. Và trong thời gian Hưng đang đợi khách lên xe thì Bình đi được 2 vòng để đón khách tại bến 1. Chênh lệch lượng khách giữa hai nhà xe bằng $5 - 3 = 2$

----- Hết -----

Cán bộ coi thi không giải thích gì thêm; các dữ liệu vào là đúng đắn không cần kiểm tra; làm bài với các tên tệp đúng như quy định trong đề.