

TỔNG QUAN ĐỀ THI

Bài	Tên bài	Tệp chương trình	Tệp dữ liệu	Tệp kết quả	Bộ nhớ	Thời gian / test	Điểm
1	Kẹo ngọt	candy.*	candy.inp	candy.out	1024 MB	1 giây	6
2	Hộp kim	alloy.*	alloy.inp	alloy.out	1024 MB	1 giây	6
3	Đếm xâu	cstring.*	cstring.inp	cstring.out	1024 MB	1 giây	5
4	Chi phí	cost.*	cost.inp	cost.out	1024 MB	1 giây	3

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

Hãy lập trình giải các bài toán sau:

Bài 1. Kẹo ngọt (6 điểm)

Nhà máy kẹo "Ngọt ngào" sản xuất thanh kẹo có độ dày gồm các lớp kẹo cam và dâu xen kẽ. Mỗi chu kỳ sản xuất bao gồm a lớp kẹo cam liên tiếp, sau đó là b lớp kẹo dâu liên tiếp. Chu kỳ này được lặp đi lặp lại cho đến khi thanh kẹo có đúng N lớp.

Lưu ý rằng a lớp kẹo cam và b lớp kẹo dâu không nhất thiết phải được lặp lại nguyên vẹn ở cuối. Điều này có nghĩa là nếu sau các chu kỳ lặp đầy đủ, số lớp kẹo đã gần đủ N thì chỉ một phần của chu kỳ có thể được thêm vào để hoàn thành thanh kẹo với chính xác N lớp.

Với các số nguyên dương N, a, b cho trước, các công nhân nhà máy cần biết chính xác số lớp kẹo cam và số lớp kẹo dâu có trong thanh kẹo.

Yêu cầu: Hãy tính số lớp kẹo cam và số lớp kẹo dâu có trong thanh kẹo với đúng N lớp.

Dữ liệu: Vào từ file văn bản **candy.inp**: Gồm một dòng duy nhất chứa 3 số nguyên dương N, a, b ($N \leq 10^{18}; a + b \leq 10^{18}$).

Kết quả: Ghi ra file văn bản **candy.out**: Gồm hai số nguyên theo thứ tự là số lớp kẹo cam, số lớp kẹo dâu có trong thanh kẹo.

Các số được viết trên một dòng của file dữ liệu vào/ra được ghi cách nhau bởi dấu cách.

Ví dụ:

candy.inp	candy.out	Giải thích
8 3 4	4 4	Thanh kẹo được tạo ra theo chu kỳ gồm 3 lớp kẹo cam và 4 lớp kẹo dâu. Với $N = 8$, đầu tiên ta tạo ra một chu kỳ đầy đủ gồm 3 lớp cam và 4 lớp dâu. Sau đó, ta còn 1 lớp nữa để đạt đúng $N = 8$, nên ta thêm 1 lớp cam. Do đó, thanh kẹo có 4 lớp cam và 4 lớp dâu.

Ràng buộc:

- Có 60% số test tương ứng 60% số điểm của bài có $1 \leq N \leq 10^7, a + b \leq 10^9$;
- 40% số test còn lại tương ứng với 40% số điểm của bài không có ràng buộc gì thêm..

Bài 2. Hợp kim (6 điểm)

Eldoria là một vương quốc cổ kính, được bao phủ bởi những cánh rừng rậm rạp, những dòng sông trong vắt và những ngọn núi hùng vĩ. Nơi đây, phép thuật hòa quyện với tự nhiên, tạo nên một thế giới kỳ ảo và đầy màu sắc. Người dân Eldoria sống trong hòa bình và hạnh phúc, họ biết ơn những điều kỳ diệu mà đất mẹ ban tặng.

Tuy nhiên, một thế lực hắc ám từ đáy vực sâu đã trỗi dậy. Đó là Morgoth, một chúa quỷ cổ đại, hắn ta cùng với lũ quái vật đáng sợ đã xâm chiếm những vùng đất lân cận và giờ đây, hắn ta nhắm đến Eldoria. Morgoth muốn biến Eldoria thành một vương quốc của bóng tối, nơi mà nỗi sợ hãi và đau khổ ngự trị.

Trước mối đe dọa khủng khiếp này, người dân Eldoria vô cùng lo lắng. Họ đã tìm đến Gorim, một chàng thợ rèn người lùn nổi tiếng với tài năng chế tạo những vũ khí phép thuật. Với đôi bàn tay khéo léo và một trái tim nhiệt huyết, Gorim đã nhận lời giúp đỡ người dân Eldoria. Anh ta biết rằng để đánh bại Morgoth và lũ quái vật của hắn, cần phải có những vũ khí đặc biệt, được rèn từ hai loại nguyên liệu quý hiếm là: “*tinh thể lửa*”, được sinh ra từ lòng đất, tượng trưng cho sự dũng cảm và sức mạnh; “*bột gió*”, được tạo thành từ những cơn lốc xoáy, lại mang đến sự nhanh nhẹn và linh hoạt.

Gorim đã khám phá ra rằng việc kết hợp “*tinh thể lửa*” và “*bột gió*” theo những tỷ lệ khác nhau sẽ tạo ra những vũ khí phép thuật vô cùng mạnh mẽ. Để tìm ra những tỷ lệ này thì ứng với mỗi đoạn số $[a; b]$ được cho trước Gorim phải thực hiện hai phép toán:

1. *Tìm số lượng số nguyên tố*: Đầu tiên, Gorim cần tìm số lượng số nguyên tố nằm trong đoạn số $[a; b]$. Số lượng này sẽ là số lượng “*tinh thể lửa*” cần phải có.
2. *Tính trung bình cộng các số nguyên tố*: Tiếp theo, Gorim cần tính trung bình cộng của tất cả các số nguyên tố trong đoạn $[a; b]$. Giá trị trung bình cộng này chính là số lượng cần có của nguyên liệu “*bột gió*”.

Yêu cầu: Bạn hãy lập trình giúp Gorim hoàn thành công việc tính toán phức tạp nói trên.

Đầu vào: Vào từ file văn bản **alloy.inp**:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^4$) cho biết số lượng vũ khí Gorim phải rèn;
- n dòng tiếp theo, mỗi dòng ghi hai số nguyên dương a, b ($1 \leq a < b \leq 10^6$).

Kết quả: Ghi ra file văn bản **alloy.out**: Gồm n dòng, mỗi dòng chứa hai giá trị phân biệt nhau bởi một khoảng trắng, ứng với số lượng “*tinh thể lửa*” và “*bột gió*” cần thiết cho mỗi loại vũ khí. Trong đó giá trị “*bột gió*” được làm tròn đến chữ số thập phân thứ hai.

Các số được viết trên một dòng của file dữ liệu vào/ra được ghi cách nhau bởi dấu cách.

Ví dụ:

alloy.inp	alloy.out
2	4 4.25
1 10	2 4.00
3 5	

Ràng buộc:

- Có 40% số test tương ứng với 40% số điểm của bài có $1 \leq n \leq 10$; $0 \leq a < b \leq 10^4$;
- Có 30% số test tương ứng với 30% số điểm của bài có $1 \leq n \leq 10^2$; $0 \leq a < b \leq 5 \times 10^5$;
- 30% số test còn lại tương ứng với 30% số điểm của bài không có ràng buộc gì thêm..

Bài 3. Đếm xâu (5 điểm)

Cho hai xâu kí tự S và T . Đếm số lần xuất hiện của xâu S trong xâu T . Các ký tự trong xâu S và T là các chữ cái la tinh bao gồm các chữ cái in hoa: (A, B, C, \dots, Z) , in thường (a, b, c, \dots, z) .

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

- Dòng thứ nhất chứa xâu S (độ dài xâu S tối đa là 10^6);
- Dòng thứ hai chứa xâu T (độ dài xâu T tối đa là 10^6).

Kết quả: Ghi ra file văn bản **cstring.out**: Một số nguyên là số lần xâu S xuất hiện trong xâu T .

Ví dụ:

cstring.inp	cstring.out
ab	3
abcabcab	

Ràng buộc:

- Có 10% số test tương ứng 10% số điểm có độ dài xâu S là 1;
- Có 30% số test tương ứng 30% số điểm có độ dài hai xâu S và T tối đa là 10^3 ;
- 60% số test còn lại tương ứng 60% số điểm có độ dài hai xâu S và T tối đa là 10^6 .

Bài 4. Chi phí (3 điểm)

Chuẩn bị vào năm học mới, một trường học muốn sử dụng dịch vụ đưa đón trẻ đến trường. Nhà trường vẽ sơ đồ có n điểm đón. Các điểm đón được thông với nhau bởi $n - 1$ con đường. Con đường giữa hai điểm đón x_i và y_i có độ dài w_i (mét). Mỗi điểm đón có t_i học sinh. Nhà trường dự kiến số tiền phải chi trả cho dịch vụ đưa đón trẻ là k (đồng).

Được biết chi phí để vận chuyển một học sinh đi trên đoạn đường 1 mét là 1 đồng.

Công ty cung cấp dịch vụ đưa đón trẻ muốn tính chi phí đưa đón toàn bộ học sinh, để đánh giá xem công ty có đủ khả năng ký hợp đồng với nhà trường hay không. Có 3 mức độ đánh giá:

- + “NO” – mức độ ứng với k nhỏ hơn trường hợp chi phí ít nhất.
- + “YES” – mức độ ứng với k lớn hơn hoặc bằng trường hợp chi phí ít nhất và k nhỏ hơn hoặc bằng trường hợp chi phí lớn nhất.
- + “GOOD” – mức độ ứng với k lớn hơn trường hợp chi phí lớn nhất.

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

- Dòng đầu chứa số nguyên dương n ($1 \leq n \leq 10^5$) - số điểm đón, và số nguyên dương k ($1 \leq k \leq 10^{18}$);
- $n - 1$ dòng tiếp theo, dòng thứ i chứa 3 số nguyên dương x_i, y_i, w_i xác định điểm đón thứ i ($1 \leq x_i, y_i \leq n$; $1 \leq w_i \leq 10^4$);
- Dòng cuối chứa n số nguyên t_1, t_2, \dots, t_n ($t_i \leq 10^4$) là số lượng học sinh cần đón của từng điểm theo thứ tự tăng dần từ 1 đến n .

Kết quả: Ghi ra file văn bản **cost.out**: Trên một dòng, gồm hai số nguyên là chi phí nhỏ nhất, chi phí lớn nhất và tiếp đó là mức độ đánh giá của công ty (NO, YES, GOOD).

Các số, từ được viết trên một dòng của file dữ liệu vào/ra được ghi cách nhau bởi dấu cách.

Ví dụ:

cost.inp	cost.out
3 20 1 2 5 2 3 6 3 1 2	27 39 NO
5 39 2 3 2 1 3 3 2 4 1 4 5 2 1 2 3 2 1	16 39 YES
4 200 2 1 5 3 2 10 2 4 1 1 5 2 9	33 164 GOOD

Ràng buộc:

- Có 60% số test tương ứng 60% số điểm có $n \leq 10^3$;
- 40% số test còn lại tương ứng 40% số điểm có $n \leq 10^5$.

----- HẾT -----

- Thí sinh không được sử dụng tài liệu;
- Giám thị không giải thích gì thêm.

Họ và tên thí sinh: Số báo danh: