



OLYMPIC TIN HỌC SINH VIÊN LẦN THỨ XXIII, 2014

Khối thi: Chuyên tin

Thời gian làm bài: 180 phút

Ngày thi: 29-10-2014

Nơi thi: **ĐẠI HỌC CÔNG NGHIỆP TP HCM**

TỔNG QUAN ĐỀ THI

Tên bài	Tên file chương trình	Tên file dữ liệu	Tên file kết quả	Hạn chế thời gian cho mỗi test
NĂM NHUẬN	LEAPYEAR.???	LEAPYEAR.INP	LEAPYEAR.OUT	0.2 giây
GIAO ĐẦU HỮU NGHỊ	FAIRPLAY.???	FAIRPLAY.INP	FAIRPLAY.OUT	0.5 giây
DÂY TỈ LỆ	RATEEQUA.???	RATEEQUA.INP	RATEEQUA.OUT	1.0 giây
HÀNG ĐỢI XOR	XORQUEUE.???	XORQUEUE.INP	XORQUEUE.OUT	2 giây

Chú ý:

- Dấu ??? được thay thế bởi đuôi ngầm định của ngôn ngữ được sử dụng để cài đặt chương trình.
- Thí sinh phải nộp cả file mã nguồn của chương trình và file chương trình thực hiện (chương trình đã được biên dịch ra file .exe).

Hãy lập trình giải các bài sau đây:

Bài 1. NĂM NHUẬN (10 điểm)

Mỗi loại lịch có ngày và tháng nhuận khác nhau. Với dương lịch, chu kỳ Trái đất quay quanh Mặt trời là $365 + \frac{1}{4}$ ngày. Nhưng theo quy ước thì mỗi năm chỉ có 365 ngày, nên năm dương lịch sẽ chênh với thời gian thực là $\frac{1}{4}$ ngày. Điều này cũng có nghĩa sau 4 năm thì dương lịch sẽ dư một ngày và sẽ có một năm nhuận một ngày. Năm nhuận này theo quy ước rơi vào tháng hai (tức là tháng có 29 ngày). Trong khi đó, một năm âm lịch có 354 ngày, và nếu so sánh với dương lịch (365 ngày) thì âm lịch ngắn hơn 11 ngày. Như vậy cứ ba năm, âm lịch lại ngắn hơn dương lịch 33 ngày, tức là ba năm âm lịch sẽ nhuận một tháng chứ không nhuận một ngày như dương lịch. Muốn tính năm âm lịch nào đó có tháng nhuận hay không chỉ cần làm phép toán đơn giản là, lấy năm dương lịch chia cho 19 nếu chia hết hoặc có các số dư 3, 6, 9, 11, 14, 17 thì chắc chắn năm đó là năm nhuận.

Cho a và b là 2 năm dương lịch, $a \leq b$, hãy cho biết trong các năm từ a đến b (kể cả a và b) có bao nhiêu năm nhuận theo âm lịch.

Dữ liệu: Vào từ file văn bản LEAPYEAR.INP gồm một dòng chứa 2 số nguyên a và b ($0 < a \leq b \leq 10^6$).

Kết quả: Đưa ra file văn bản LEAPYEAR.OUT trên một dòng 1 số nguyên – số năm nhuận theo âm lịch.

Ví dụ:

LEAPYEAR.INP
2000 2014

LEAPYEAR.OUT
6

Bài 2. GIAO ĐẦU HỮU NGHỊ (25 điểm)

Để tạo không khí vui vẻ náo nhiệt, trong buổi giao lưu giữa sinh viên các trường tham dự OLP – ACM, trường đăng cai OLP năm tới đề xuất tổ chức một cuộc thi đấu game online tay đôi giữa sinh viên trường mình với sinh viên trường sở tại. Mỗi trường cử ra một đội n người, tạo thành n cặp đấu, sinh viên cùng trường không đấu với nhau. Trò chơi được chọn là một trò chơi rất phổ biến, được các bạn trẻ yêu thích, ai cũng biết và đã từng chơi nhiều trước đó. Mọi người đều biết chỉ số năng lực của mình trong trò chơi này và biết rằng nếu đấu tay đôi, ai có năng lực cao hơn sẽ thắng. Trong các trận đấu tay đôi, người thắng sẽ được 1 điểm, người thua – 0 điểm. Thời gian chơi được quy định đủ để phân biệt thắng thua. Các trận hòa sẽ kéo dài vô hạn và sẽ bị hủy kết quả khi hết thời gian. Với tinh thần *fair play* các bạn trường đề xuất ngồi vào vị trí thi đấu, truy nhập vào hệ thống và gửi về máy chủ chỉ số năng lực của mình. Trường đoàn của trường sở tại có 0.5 giây để xử lý thông tin, phân công ai đấu với ai để tổng số điểm thu được là lớn nhất.

Hãy xác định, với cách bố trí tối ưu các cặp đấu, đội của trường sở tại sẽ có bao nhiêu điểm.

Dữ liệu: Vào từ file văn bản FAIRPLAY.INP:

- ✚ Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^5$),
- ✚ Dòng thứ 2 chứa n số nguyên a_1, a_2, \dots, a_n , trong đó a_i – chỉ số năng lực của người thứ i thuộc đội của trường đề xuất, $1 \leq a_i \leq 10^9$, $i = 1 \div n$,
- ✚ Dòng thứ 3 chứa n số nguyên b_1, b_2, \dots, b_n , trong đó b_i – chỉ số năng lực của người thứ i thuộc đội của trường sở tại, $1 \leq b_i \leq 10^9$, $i = 1 \div n$.

Kết quả: Đưa ra file văn bản FAIRPLAY.OUT một số nguyên – số điểm đội trường sở tại có thể đạt được với cách bố trí cặp chơi tối ưu.

Ví dụ:

FAIRPLAY.INP
5
10 15 30 20 25
28 24 20 16 14

FAIRPLAY.OUT
4

Bài 3. DÃY TỈ LỆ (30 điểm)

Xét dãy số Fibonacci $\{F_n\}$ theo định nghĩa:
$$\begin{cases} F_1 = F_2 = 1 \\ F_n = F_{n-1} + F_{n-2} \text{ với } n > 2 \end{cases}$$

Dãy số nguyên $\{a_n\} = \{a_1, a_2, \dots, a_n\}$ được gọi là dãy tỉ lệ của $\{F_n\}$ nếu ta có: $\frac{a_1}{F_1} =$

$$\frac{a_2}{F_2} = \dots = \frac{a_n}{F_n}$$

Cho a_1 và n , hãy tính tổng $S = (a_1 + a_2 + \dots + a_n)$ và đưa ra số dư của S chia cho (10^9+7) .

Dữ liệu: vào từ file văn bản RATEEQUA.INP gồm 1 dòng ghi 2 số nguyên a_1 và n ($0 < a_1, n \leq 10^{15}$).

Kết quả: ghi ra file văn bản RATEEQUA.OUT một số nguyên – số dư tìm được.

Ví dụ:

RATEEQUA.INP
3 5

RATEEQUA.OUT
36

Bài 4. HÀNG ĐỢI XOR (35 điểm)

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	

----- Hết -----