PREVOI 2024

THI THỬ HỌC SINH GIỎI QUỐC GIA

TRUNG HỌC PHỔ THÔNG

NĂM HỌC 2024-2025

Môn: **TIN HỌC**

Thời gian: **180** phút (*không kể thời gian giao đề*)

Ngày thi: 18/11/2024

Đề thi gồm 06 trang, 03 bài

TỔNG QUAN ĐỀ THI

	Tên bài	File chương trình	File dữ liệu vào	File kết quả	Giới hạn thời gian	Giới hạn bộ nhớ	Điểm
Bài 1	Làm văn	WORD.*	WORD.INP	WORD.OUT	1 giây	1024 MB	7
Bài 2	Đấu bài	CARD.*	CARD.INP	CARD.OUT	1 giây	1024 MB	7
Bài 3	Đèn đỏ	LIGHT.*	LIGHT.INP	LIGHT.OUT	3 giây	1024 MB	6

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

***Hãy lập trình giải các bài toán sau*:**

Bài 1: Làm văn (7 điểm)

Bạn là một học sinh. Vì vậy, chắc hẳn bạn còn nhớ rất nhiều thơ văn. Khi làm thơ, mỗi thể loại thơ lại có một số quy tắc riêng biệt về việc sử dụng dấu thanh trong các câu liên tiếp, gọi là luật bằng trắc. Vậy có tồn tại luật bằng trắc khi làm văn hay không? Câu trả lời là có. Ta hãy thử viết lại hai câu đầu tiên của đề bài này như sau:

“Bạn là một học sinh. Vì vậy, chắc hẳn thơ văn bạn còn nhớ rất nhiều.”

Rõ ràng cách viết lại này nghe xuôi tai hơn cách viết trước. Lý do là bởi cách viết thứ nhất đã không tuân theo luật bằng trắc: kết thúc câu trước là thanh ngang (“sinh”) thì kết thúc câu sau không nên là thanh ngang (“văn”), mà nên là thanh huyền (“nhiều”).

Nhà văn Nguyên Như muốn nhờ bạn dựa vào luật bằng trắc để xây dựng một mô hình máy viết văn tự động. Ông có danh sách của N từ tiếng Việt. Một số cặp từ được phép đứng ở cạnh nhau trong cùng câu văn, còn các cặp từ khác thì không. Ngoài ra, bạn chỉ được phép bắt đầu câu văn bằng một số từ nhất định. Kết thúc câu luôn luôn là dấu chấm. Mỗi câu văn phải chứa ít nhất một từ. Một bài văn được hợp thành bởi một số câu văn có thứ tự. Bài văn không có câu văn nào cũng được coi là một bài văn.

N từ đã cho được chia thành hai nhóm: **từ xấu** và **từ đẹp**. Nếu từ cuối cùng của một câu là từ đẹp, thì câu đó được gọi là **câu hoàn hảo**. Còn nếu từ cuối cùng của câu là từ xấu, thì câu đó được gọi là **câu không hoàn hảo**. Nếu bài văn chứa câu không hoàn hảo, thì nó *chỉ được phép là câu cuối cùng trong bài văn đó*.

Vì một câu văn không nên quá dài cũng không nên quá ngắn, nhà văn Nguyên Như đã xác định trước một số nguyên *K*, từ đó ta có công thức tính **điểm số của câu văn có w từ** (w > 0, *dấu chấm không được tính là một từ*) bằng:

K + (K – 1) + (K – 2) + ... + (K – w + 1)

Bạn cũng được cho trước quan hệ hợp/không hợp giữa 6 dấu thanh (ngang, huyền, sắc, hỏi, ngã, nặng) dưới dạng một ma trận nhị phân 6 × 6. Mỗi quan hệ sẽ cho biết có nên đặt câu có kết thúc bằng dấu thanh này ở ngay sau câu có kết thúc bằng dấu thanh kia hay không. *Hai câu văn cạnh nhau kết thúc bằng hai dấu thanh hợp nhau* gọi là một **cặp câu hợp nhau**. Chỉ có một ngoại lệ: *một câu văn không hoàn hảo sẽ không được xét quan hệ hợp nhau với câu văn trước đó, bất kể chúng kết thúc bằng dấu thanh gì.*

Nhà văn Nguyên Như muốn bạn tìm thuật toán để sinh ra một bài văn dài không quá *M* từ có **độ mượt mà** là lớn nhất. Độ mượt mà của một bài văn được tính như sau:

Độ mượt mà = (tổng điểm của các câu văn) × (số cặp câu hợp nhau trong bài văn)

Yêu cầu: Cho các số nguyên *N, K, M*, đồng thời cho quan hệ giữa các từ và quan hệ giữa các dấu thanh, hãy tính độ mượt mà của bài văn có độ mượt mà lớn nhất dài không quá *M* từ.

Dữ liệu: Vào từ file **WORD.INP** gồm :

- Dòng đầu chứa ba số nguyên dương *N, K, M* (*1 ≤ N, K, M* ≤ 1000).
- Dòng thứ hai gồm *N* số nguyên trong khoảng từ 1 đến 6, biểu thị dấu thanh của các từ.
- 6 dòng tiếp theo, mỗi dòng gồm 6 số nguyên nhận một trong hai giá trị 0 hoặc 1. Nếu số thứ *j* trên dòng thứ *i* trong 6 dòng này bằng 0 thì câu trước kết thúc bằng dấu *i* sẽ không hợp với câu sau kết thúc bằng dấu *j*. Ngược lại, nếu số thứ *j* trên dòng thứ *i* trong 6 dòng này bằng 1 thì câu trước kết thúc bằng dấu *i* sẽ hợp với câu sau kết thúc bằng dấu *j*.
- *N + 1* dòng tiếp theo, mỗi dòng gồm *N + 1* số nguyên biểu thị một ma trận nhị phân kích thước *(N + 1) × (N + 1)*. Trong ma trận này:
  - Nếu số thứ *j (1 ≤ j ≤ N)* trên dòng thứ *i (1 ≤ i ≤ N)* bằng 1 thì từ thứ *j* có thể được viết ngay sau từ thứ *i* trong cùng câu văn. Ngược lại, nếu số thứ *j* trên dòng thứ *i* bằng 0 thì sẽ không được phép viết từ thứ *j* ngay sau từ thứ *i* trong cùng câu văn.
  - Nếu số thứ *N + 1* trên dòng thứ *i (1 ≤ i ≤ N)* bằng 1 thì từ thứ *i* là từ đẹp, còn ngược lại thì từ thứ *i* là từ xấu.
  - Nếu số thứ *i (1 ≤ i ≤ N)* trên dòng thứ *N + 1* bằng 1 thì bạn được phép bắt đầu câu văn bằng từ thứ *i*, còn nếu bằng 0 thì không được phép.
  - Số thứ *N + 1* trên dòng thứ *N + 1* luôn luôn bằng 0.

Kết quả: Ghi ra file văn bản **WORD.OUT** một số nguyên duy nhất là độ mượt mà của bài văn có độ mượt mà lớn nhất dài không quá *M* từ.

Ví dụ:

WORD.INP	WORD.OUT
7 4 6 2 4 1 5 6 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0	20

Giải thích ví dụ:

Có một số bài văn có độ mượt mà tối ưu bằng 20, chẳng hạn như:

- "2 3 4. 2 5. 2.", độ mượt mà bằng ((4 + 3 + 2) + (4 + 3) + 4) × 1 = 20.

- "1 3 4. 2 5. 7.", độ mượt mà bằng ((4 + 3 + 2) + (4 + 3) + 4) × 1 = 20. Tuy rằng hai câu văn cuối cùng hợp nhau về dấu thanh kết thúc, nhưng vì câu "7." là câu không hoàn hảo nên không được tính là hợp với câu "2 5.".

Ràng buộc:

- **Subtask 1:** 25% số test tương ứng với 25% số điểm có $1 \le N, M \le 7$.
- **Subtask 2:** 25% số test khác tương ứng với 25% số điểm có $1 \le N, M \le 20$.
- **Subtask 3:** 25% số test khác tương ứng với 25% số điểm có $1 \le N, M \le 100$.
- **Subtask 4:** 25% số test còn lại ứng với 25% số điểm không có ràng buộc nào thêm.

Bài 2: Đấu bài (7 điểm)

Dạo gần đây Kiên say mê chơi một trò chơi đấu bài trên máy tính. Một hôm, Kiên dẫn bạn cùng lớp là Khánh về thăm nhà sau buổi học. Khánh được Kiên giới thiệu về trò đấu bài kia. Luật chơi rất đơn giản. Có $2 \times N$ quân bài, được xếp thành hai hàng, mỗi hàng N quân. Các quân ở hàng dưới thuộc về người chơi, còn các quân bài ở hàng trên là của máy. Trên mỗi quân bài có ghi một số nguyên là sức mạnh của quân bài đó. Người chơi và máy cũng được cho một số nguyên không âm M. Đầu tiên, máy sẽ **chọn ngẫu nhiên** hai số nguyên X_c và K_c thỏa mãn $0 \le X_c \le [M/N]$ và $1 \le K_c \le N$. Máy giữ bí mật hai số nguyên này, người chơi không hề biết. Sau đó, đến lượt người chơi chọn hai số nguyên X_p và K_p thỏa mãn $0 \le X_p \le [M/N]$ và $1 \le K_p \le N$. Khi cả máy và người chơi đã chọn xong hai con số của mình, các thao tác sau đây sẽ được thực hiện:

- Sức mạnh của các quân bài ở hàng trên được cộng một lượng bằng X_c, riêng quân bài ở vị trí K_c ở hàng trên còn được cộng thêm một lượng bằng $(M – N \times X_c)$ nữa.

- Sức mạnh của các quân bài ở hàng dưới được cộng một lượng bằng X_p, riêng quân bài ở vị trí K_p ở hàng dưới còn được cộng thêm một lượng bằng $(M – N \times X_p)$ nữa.

Sau các thao tác trên, ta sẽ so sánh các lá bài ở cùng vị trí trên hai hàng. Cụ thể, gọi *C[i]* là sức mạnh của lá bài ở vị trí thứ *i* ở hàng trên $(1 \le i \le N)$ và *P[i]* là sức mạnh của lá bài ở vị trí thứ *i* ở hàng dưới $(1 \le i \le N)$. Ở mỗi vị trí thứ *i* trong số *N* vị trí, nếu *C[i] > P[i]* thì máy được cộng 1 điểm, còn nếu *C[i] < P[i]* thì người chơi được cộng 1 điểm. Người chơi sẽ thắng cuộc nếu có số điểm lớn hơn số điểm của máy.

Với mỗi phương án chọn (X_p, K_p), ta gọi **khả năng chiến thắng** của nó là số cặp (X_c, K_c) mà máy chọn dẫn đến chiến thắng dành cho người chơi.

Phổ biến luật chơi cho Khánh xong, Kiên cho Khánh chơi thử một ván. Vừa nhìn thấy $2 \times N$ lá bài và số *M*, Kiên đã thở dài: “Bài xấu thế này làm sao mà thắng được”.

Nhưng Khánh không nghĩ thế. Khánh bảo với Kiên: “Thắng được tốt!”.

Kiên không tin, nên bắt Khánh phải chứng minh những gì mình vừa nói là sự thật.

Yêu cầu: Hãy giúp Khánh chứng minh luận điểm của mình bằng cách tìm cặp số nguyên (X_p, K_p) mang lại khả năng chiến thắng cao nhất.

Dữ liệu: Vào từ file **CARD.INP** gồm :

- Dòng đầu chứa hai số nguyên dương *N*, *M* ($1 \le N, M \le 10^5$)
- Dòng thứ hai chứa *N* số nguyên có giá trị tuyệt đối không vượt quá 10^9, là sức mạnh của các quân bài ở hàng trên mà máy sở hữu.
- Dòng thứ ba chứa *N* số nguyên có giá trị tuyệt đối không vượt quá 10^9, là sức mạnh của các quân bài ở hàng dưới mà người chơi sở hữu.

Kết quả: Ghi ra file văn bản **CARD.OUT** hai số nguyên X_p và K_p mang lại khả năng chiến thắng cao nhất, cùng với khả năng chiến thắng của phương án chọn (X_p, K_p), tất cả ba số trên một dòng và cách nhau bởi dấu cách. Nếu có nhiều đáp án cùng mang lại khả năng chiến thắng cao nhất, hãy in ra đáp án có X_p nhỏ nhất. Nếu vẫn có nhiều đáp án cùng giá trị X_p này mà mang lại khả năng chiến thắng thắng cao nhất, hãy in ra đáp án có K_p nhỏ nhất.

Ví dụ:

CARD.INP	CARD.OUT
3 11 1 -2 3 0 -3 5	3 1 8

Giải thích ví dụ:

Khả năng chiến thắng của các lựa chọn X_p và K_p như sau:

X_p	K_p	Sức mạnh các lá bài của người chơi sau khi cộng	Khả năng chiến thắng
0	1	11 -3 5	3
0	2	0 8 5	3
0	3	0 -3 16	0
1	1	9 -2 6	4
1	2	1 6 6	4
1	3	1 -2 14	0
2	1	7 -1 7	7
2	2	2 4 7	7
2	3	2 -1 12	3
3	*1*	*5 0 8*	*8*
3	2	3 2 8	8
3	3	3 0 10	6

Khả năng chiến thắng của phương án (X_p = 3, K_p = 1) là 8 vì có 8 trường hợp tương ứng của (X_c, K_c) dẫn đến chiến thắng cho người chơi, đó là (X_c, K_c) ∈ {(0, 1), (0, 2), (0, 3), (1, 1), (1, 2), (1, 3), (2, 2), (3, 2)}.

Ràng buộc:

- **Subtask 1:** 25% số test tương ứng với 25% số điểm có $N, M \leq 300$.
- **Subtask 2:** 25% số test khác tương ứng với 25% số điểm có $M \leq 2000$.
- **Subtask 3:** 25% số test khác tương ứng với 25% số điểm có $N \leq 30$.
- **Subtask 4:** 25% số test còn lại ứng với 25% số điểm không có ràng buộc nào thêm.

Bài 3: Đèn đỏ (6 điểm)

Người dân đang cảm thấy không hài lòng về hệ thống đèn tín hiệu giao thông ở thành phố Omega. Họ than phiền rằng thời gian chờ đèn đỏ khi đi lại giữa các địa điểm trong thành phố là quá dài. Trước khi quyết định thay đổi hệ thống đèn giao thông, thị trưởng thành phố là ngài PVH muốn kiểm chứng lại phản ánh này của người dân.

Thành phố Omega có N địa điểm, được nối với nhau bởi N - 1 con đường hai chiều. Mỗi con đường nối giữa hai địa điểm khác nhau. Hệ thống các con đường đảm bảo với mọi cặp địa điểm (u, v) thỏa mãn $1 \leq u, v \leq N$ và $u \neq v$, luôn có một và chỉ một đường đi từ địa điểm u đến địa điểm v thông qua một hoặc nhiều con đường. Trên mỗi con đường có một cột đèn giao thông. Chu kỳ đèn của mọi cột đèn là K giây. Cột đèn giao thông trên con đường thứ i trong số N-1 con đường sẽ bật đèn xanh trong g_i giây, bắt đầu từ thời điểm 0, sau đó chuyển sang bật đèn đỏ trong K - g_i giây, rồi lại chuyển sang bật đèn xanh trong g_i giây, sau đó chuyển sang bật đèn đỏ trong K - g_i giây,... cứ thế tiếp tục quá trình này mãi mãi. Khi đèn đang đỏ trên con đường nối hai địa điểm u và v, các phương tiện không được phép xuất phát từ u để sang v hoặc xuất phát từ v để sang u, mà phải chờ cho đến khi đèn xanh bật lên mới được di chuyển (các phương tiện đang ở giữa con đường nối u và v thì vẫn được tiếp tục di chuyển). Còn khi đèn đang xanh thì các phương tiện có thể thoải mái di chuyển trên con đường nối u và v. Con đường thứ i trong

số N-1 con đường có độ dài c_i, là số giây cần để một phương tiện đi hết con đường đó (ta giả sử mọi phương tiện ở thành phố Omega di chuyển với cùng tốc độ).

Yêu cầu: Cho biết hệ thống đường xá và đèn giao thông ở thành phố Omega, đồng thời cho Q truy vấn, truy vấn thứ i gồm hai địa điểm x_i và y_i. Với mỗi truy vấn, hãy giúp ngài PVH tính số giây cần để đi từ x_i đến y_i, giả sử rằng người lái xe bắt đầu hành trình ở thời điểm 0.

Dữ liệu: Vào từ file **LIGHT.INP** :

- Dòng đầu chứa ba số nguyên N, Q, K ($1 \le N, Q \le 50000$, $2 \le K \le 6$).
- Dòng thứ i trong số N-1 dòng tiếp theo gồm ba số nguyên p_i, g_i, c_i ($1 \le p_i \le i$, $1 \le g_i \le K$, $1 \le c_i \le 10000$), biểu thị có một con đường nối giữa địa điểm p_i và địa điểm $(i + 1)$, với số giây đèn xanh trong chu kỳ là g_i và độ dài con đường là c_i.
- Q dòng tiếp theo, mỗi dòng gồm hai số nguyên x_i và y_i ($1 \le x_i, y_i \le N$).

Kết quả: Ghi ra file **LIGHT.OUT**, với mỗi truy vấn in ra một số nguyên trên một dòng là số giây cần để đi từ x_i đến y_i, giả sử rằng người lái xe bắt đầu hành trình ở thời điểm 0.

Ví dụ:

LIGHT.INP	LIGHT.OUT
5 4 5 1 2 2 2 3 5 2 1 9 3 4 1 1 2 5 1 4 3 3 4	2 8 15 14

Ràng buộc:

- **Subtask 1:** 25% số test khác tương ứng với 25% số điểm có $N \le 1000$.
- **Subtask 2:** 25% số test tương ứng với 25% số điểm thỏa mãn con đường thứ i nối giữa hai địa điểm i và $i+1$ ($1 \le i < N$).
- **Subtask 3:** 25% số test tương ứng với 25% số điểm thỏa mãn mọi con đường đều có $g_i = 1$ ($1 \le i < N$).
- **Subtask 4:** 25% số test còn lại ứng với 25% số điểm không có ràng buộc nào thêm.

---------- HẾT ----------

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