HỘI THI TIN HỌC TRẺ
TỈNH BÀ RỊA-VŨNG TÀU
Lần thứ XXVII - Năm 2023
__oOo__
ĐỀ CHÍNH THỨC

ĐỀ THI BẢNG C - THPT
Thời gian làm bài 150 phút
Ngày thi: 28/05/2023
(Đề thi gồm 03 trang)

TỔNG QUAN VỀ ĐỀ THI

Tên bài	Điểm thẳng hàng	Trả lời câu hỏi	Mua vé
Tên file làm bài	PSTRLINE.*	QANSWERS.*	SBTICKET.*
Tên file dữ liệu	PSTRLINE.INP	QANSWERS.INP	SBTICKET.INP
Tên file kết quả	PSTRLINE.OUT	QANSWERS.OUT	SBTICKET.OUT
Tổng điểm	40	30	30

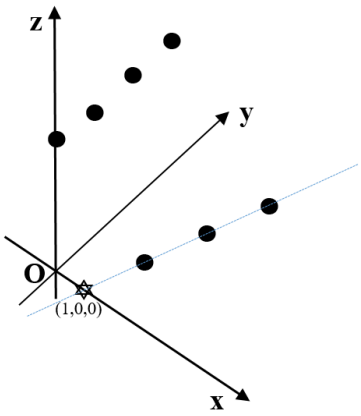
*(Dấu * được thay thế bởi PAS, PY, C hoặc CPP của ngôn ngữ lập trình Pascal, Python, C hoặc C++)*

Hướng dẫn làm bài:

- *Tên file làm bài, file dữ liệu và kết quả phải được đặt đúng theo yêu cầu của đề bài.*
- *Thí sinh tạo thư mục trên đĩa cứng có tên "**C**SBD - Ho ten" (ví dụ: "C123 - Nguyen Van Anh"), tất cả các **file làm bài** đều lưu vào thư mục này. Trong đó, chữ **C** là tên bảng dự thi, **123** là số báo danh của bạn. Chú ý: họ tên đặt không dấu.*
- *Không được ghi SBD, Họ tên hay các dấu hiệu đặc biệt gì khác vào file làm bài.*

Bài 1: Điểm thẳng hàng — Tên chương trình: PSTRLINE.*

Trên hệ trục tọa độ Descartes vuông góc Oxyz, trong đó có mặt phẳng Oxy (bao gồm các tọa độ có z = 0) theo phương ngang, trục Oz hướng lên trên. Cho ***n*** điểm có toạ độ nguyên được đánh số thứ tự từ 1 tới ***n***, điểm thứ ***i*** có tọa độ (x_i, y_i, z_i) trong đó $z_i > 0$ (có thể có nhiều điểm ở cùng một tọa độ). Từ một vị trí nào đó trên mặt phẳng Oxy, người ta muốn vẽ một đường thẳng đi qua một số điểm đã cho trong ***n*** điểm trên.



Yêu cầu: Hãy cho biết có ***nhiều nhất bao nhiêu điểm*** mà đường thẳng đó có thể vẽ đi qua được.

Dữ liệu: Vào từ file **PSTRLINE.INP**

- Dòng đầu chứa số nguyên dương ***n*** ($n \le 2000$);
- Trong ***n*** dòng tiếp theo, dòng thứ ***i*** là ba số nguyên x_i, y_i, z_i ($|x_i|, |y_i| \le 2000$; $0 < z_i \le 2000$).

  (Hai số kế nhau trong file dữ liệu cách nhau ít nhất một khoảng trắng)

Kết quả: Ghi vào file **PSTRLINE.OUT** một số nguyên duy nhất là số lượng nhiều nhất các điểm mà một đường thẳng có thể vẽ đi qua được.

Ví dụ:

PSTRLINE.INP	PSTRLINE.OUT
7 2 1 1 3 2 2 4 3 3 0 0 4 0 1 4 0 2 4 0 3 4	3

Giải thích: *Trên mặt phẳng Oxy, ở vị trí có toạ độ (1, 0, 0) ta có thể vẽ được một đường thẳng qua nhiều nhất là **3** điểm đã cho là (2, 1, 1); (3, 2, 2); (4, 3, 3).*

Bài 2: Trả lời câu hỏi **Tên chương trình: QANSWERS.***

An là một học sinh rất yêu thích môn Tin học, đặc biệt là lập trình. Để giúp con chuẩn bị cho kỳ thi Tin học trẻ tỉnh Bà Rịa-Vũng Tàu, Bố của bạn An đã đưa ra cho An bài toán để luyện tập kỹ năng lập trình như sau:

Cho dãy số nguyên dương gồm n phần tử $a_1, a_2, ..., a_n$.

Yêu cầu: Cho T câu hỏi, với mỗi câu hỏi cung cấp cặp 2 số x_i, y_i ($1 \le x_i < y_i \le n$) và yêu cầu cho biết độ chênh lệch lớn nhất của 2 số bất kì có trong các phần tử liên tiếp nhau của dãy số từ phần tử ở vị trí x_i đến phần tử ở vị trí y_i. Hãy cùng bạn An giải bài toán này!

Dữ liệu: Vào từ file **QANSWERS.INP:**

- Dòng thứ nhất chứa số nguyên dương n ($1 < n \le 10^5$);
- Dòng thứ 2 chứa n số nguyên dương $a_1, a_2, ..., a_n$ ($a_i \le 10^9$; với $i = 1, 2, ..., n$);
- Dòng thứ 3 chứa giá trị T;
- Dòng thứ i trong T dòng tiếp theo chứa 2 số x_i, y_i ($i = 1, 2, ..., T$).

  (Hai số kế nhau trong file dữ liệu cách nhau ít nhất một khoảng trắng)

Kết quả: Ghi vào file **QANSWERS.OUT** gồm T dòng, dòng thứ i chứa một số nguyên duy nhất là đáp án của câu hỏi thứ i.

Ví dụ:

QANSWERS.INP	QANSWERS.OUT
7 1 3 5 6 3 8 7 3 1 4 2 5 5 7	5 3 5

Giải thích:

- *Đoạn từ a_1 đến a_4 có độ chênh lệch lớn nhất là $a_4 - a_1 = 6 - 1 = 5$;*
- *Đoạn từ a_2 đến a_5 có độ chênh lệch lớn nhất là $a_4 - a_2 = 6 - 3 = 3$;*
- *Đoạn từ a_5 đến a_7 có độ chênh lệch lớn nhất là $a_6 - a_5 = 8 - 3 = 5$;*

Ràng buộc dữ liệu:

- *20% tests ứng với 20% số điểm ứng với T = 1;*
- *80% tests còn lại ứng với 80% số điểm (không ràng buộc gì thêm).*

https://hpcode.edu.vn

Bài 3: Mua vé **Tên chương trình: SBTICKET.***

Có n cổ động viên (*CĐV*) nhiệt thành đang xếp hàng để mua vé vào xem trận bóng đá và được đánh số thứ tự từ 1 đến n tính từ đầu hàng đến cuối hàng, cổ động viên thứ i (i = 1, 2, ..., n) có chiều cao h_i. Việc chờ đợi để được mua vé là rất mất thời gian, cho nên mỗi cổ động viên j đứng trong hàng sẽ nhờ những cổ động viên i đứng trước mà cổ động viên j nhìn thấy cổ động viên i.

Hai cổ động viên i, j ($1 \le i \le j \le n$) được gọi là "***nhìn thấy nhau***" nếu xảy ra một trong hai trường hợp sau:

- $j - i = 1$ (hai cổ động viên đứng kế nhau);
- $j - i > 1$ và chiều cao của tất cả các cổ động viên đứng giữa i và j không cao hơn chiều cao chiều cao bất kì hai động viên i, j ($max(h_k) \le min(h_i,h_j)$; $i + 1 \le k \le j - 1$).

Yêu cầu: Với mỗi cổ động viên thứ i (i = 1, 2, ..., n) cho biết số D_i là số lượng cổ động viên đứng trước mà cổ động viên i có thể ***nhìn thấy***.

Dữ liệu: Vào từ file **SBTICKET.INP**

- Dòng thứ nhất chứa số nguyên dương n ($1 < n \le 10^5$);
- Dòng thứ 2 chứa n số nguyên dương $h_1, h_2, ..., h_n$ ($h_i \le 10^9$; i = 1, 2, ..., n).

  (Hai số kế nhau trong file dữ liệu cách nhau ít nhất một kí tự trắng)

Kết quả: Ghi vào file **SBTICKET.OUT** các giá trị $D_1, D_2, ..., D_n$ (hai số kế nhau cách nhau một kí tự trắng).

Ví dụ:

SBTICKET.INP	SBTICKET.OUT
7 2 4 6 1 2 3 6	0 1 1 1 2 2 2

Giải thích:

- *CĐV 1: Không có CĐV đứng trước;*
- *CĐV 2: nhìn thấy CĐV 1;*
- *CĐV 3: nhìn thấy CĐV 2;*
- *CĐV 4: nhìn thấy CĐV 3;*
- *CĐV 5: nhìn thấy CĐV 3, 4;*
- *CĐV 6: nhìn thấy CĐV 3, 5;*
- *CĐV 7: nhìn thấy CĐV 3, 6.*

Ràng buộc dữ liệu:

- *30% tests ứng với 30% số điểm ứng với n ≤ 100;*
- *70% tests còn lại ứng với 70% số điểm (không ràng buộc gì thêm)*

========HẾT========

https://hpcode.edu.vn