

Thi thử HSG9 2025 – Ngày thi thứ hai

Tổng quan

	Tên bài	Mã nguồn	Dữ liệu vào	Dữ liệu ra	Điểm
Câu 1	Pickleball	PCB.*	PCB.inp	PCB.out	4
Câu 2	Lại là một bài toán về trị tuyệt đối	ABS.*	ABS.inp	ABS.out	4
Câu 3	Trò chơi con mực	SQUID.*	SQUID.inp	SQUID.out	4
Câu 4	Một đời liêm khiết	LIEM.*	LIEM.inp	LIEM.out	4
Câu 5	Kẻ về nhì	SECOND.*	SECOND.inp	SECOND.out	4

Dấu * trong tên tệp ở cột Mã nguồn có thể được thay bởi phần mở rộng của các ngôn ngữ Pascal (.pas), C++ (.cpp), Python (.py).

Các tệp dữ liệu vào và dữ liệu ra thuộc cùng một thư mục với tệp mã nguồn. Tệp mã nguồn, dữ liệu vào và dữ liệu ra đều là các tệp văn bản.

Thí sinh có thời gian làm bài là **150** phút. Thí sinh được phép nộp bài không giới hạn số lần; tuy nhiên bài nộp của thí sinh chỉ được chấm với trường hợp ví dụ trong thời gian thi. Các bài nộp sẽ được chấm lại sau khi kỳ thi kết thúc.

Các giới hạn về thời gian và bộ nhớ của mỗi bài sẽ được đề cập riêng ở từng bài.

Đề thi gồm **07** trang, **05** câu.

Câu 1. Pickleball (4 điểm)

Mã nguồn	Dữ liệu vào	Dữ liệu ra	Thời gian	Bộ nhớ
PCB.*	PCB.inp	PCB.out	1 giây	1024 MB

“Anh hẹn em Pickleball ... Ta vờn nhau Pickleball ...”

Pickleball là một bài hát được thể hiện bởi Đỗ Phú Quý và ra mắt vào tháng 11 năm 2024. Bài hát khi được ra mắt đã cực kỳ viral trên mạng, không phải vì nó hay mà là vì sự hữu dụng của nó trong rất nhiều trường hợp. Ví dụ như bạn có thể dùng tiếng nhạc để làm đồng hồ báo thức vì nó sẽ giúp bạn dậy sớm hơn nó và tắt nó trước khi nó bật lên hay là các cơ quan tổ chức không cần phải tra tấn bằng những hình thức tàn bạo như trước đó nữa mà giờ chỉ cần bật bài này cũng đủ để ép họ phải thực hiện theo. Đặc biệt hơn, sức khỏe của nhiều bệnh nhân đã cải thiện đáng kể mặc dù trước đó họ đã sắp hết cứu sau khi bài nhạc này được bật lên, thậm chí cả những người liệt giường cũng phải bật dậy để tắt bài hát này...

Ngoài ra, Pickleball còn là tên của một môn thể thao. Hôm nay là một ngày bình thường nên Alice quyết định rủ Bob đi đánh pickleball chung. Sân chơi hiện tại có thể mô phỏng bởi trục Ox . Alice và Bob sẽ chơi với nhau t lần và tại lần thứ i , Alice đang ở vị trí l_i

trong khi Bob thì lại ở vị trí r_i . Họ sẽ thực hiện đánh bóng k_i lần, bắt đầu từ Alice, sau đó là đến Bob và cứ liên tục như vậy. Cách đánh bóng của họ như sau:

- Với lượt thứ j ($1 \leq j \leq k_i$), nếu Alice đánh thì Bob sẽ phải lùi về bên phải sao cho chênh lệch vị trí của họ phải chia hết cho j . Để nhận thấy trong trường hợp Alice đánh thì j luôn là số lẻ.
- Với lượt thứ j ($1 \leq j \leq k_i$), nếu Bob đánh thì Alice sẽ phải lùi về bên trái sao cho chênh lệch vị trí của họ phải chia hết cho j . Để nhận thấy trong trường hợp Bob đánh thì j luôn là số chẵn.
- Chênh lệch vị trí của hai người được tính bằng giá trị của $r_i - l_i$.

Đây là một cách đánh rất thú vị vì nó luôn đảm bảo rằng họ không bao giờ “vượt mức Pickleball”. Bây giờ cả hai người thắc mắc rằng trong t lượt đánh của họ thì vị trí đứng của họ sau khi mỗi lượt đánh kết thúc là ở đâu. Bạn có thể giải quyết thắc mắc này giúp họ được không, biết đâu bạn sẽ được rủ chơi pickleball cùng đấy.

Dữ liệu vào: Nhập từ file `PCB.inp`:

- Dòng đầu tiên gồm một số nguyên dương t là số lượt đánh ($1 \leq t \leq 69$).
- t dòng tiếp theo, dòng thứ $i + 1$ chứa thông tin của lượt thứ i gồm ba số nguyên l_i, r_i, k_i ($|l_i|, |r_i| \leq 10^9, 1 \leq k_i \leq 10^5, l_i < r_i$).

Dữ liệu ra: Ghi ra file `PCB.out`:

- Với mỗi lượt đánh, in ra hai số nguyên trên một dòng là tọa độ của Alice và Bob sau khi kết thúc lượt đánh.

Ví dụ:

PCB.inp	PCB.out	Giải thích
5	6 9	Không có giải thích gì thêm.
6 9 1	5 9	
6 9 2	5 11	
6 9 3	3 11	
6 9 4	3 13	
6 9 5		

Câu 2. Lại là một bài toán về trị tuyệt đối (4 điểm)

Mã nguồn	Dữ liệu vào	Dữ liệu ra	Thời gian	Bộ nhớ
ABS.*	ABS.inp	ABS.out	1 giây	1024 MB

Chắc hẳn các bạn đã quá quen thuộc với bài toán sau:

“Cho dãy số a gồm n phần tử. Tìm giá trị x nguyên dương sao cho $|a_1 - x| + |a_2 - x| + \dots + |a_n - x|$ là nhỏ nhất có thể.”

Tuy nhiên hôm nay, chúng ta sẽ không giải quyết bài toán trên mà là một bài toán khác có nội dung cũng khá tương tự. Cụ thể hơn, nó có nội dung như sau:

Cho dãy số a gồm n phần tử. Tìm giá trị x nguyên dương sao cho giá trị $S = \max(|a_1 - x|, |a_2 - x|, \dots, |a_n - x|) - \min(|a_1 - x|, |a_2 - x|, \dots, |a_n - x|)$ là nhỏ nhất có thể.

Đề bài chỉ đơn giản như vậy thôi, chúc bạn may mắn nhé!

Dữ liệu vào: Nhập từ file `ABS.inp`:

- Dòng đầu tiên gồm một số nguyên dương n ($1 \leq n \leq 2 \times 10^5$).
- Dòng thứ hai gồm n số nguyên dương a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Dữ liệu ra: Ghi ra file `ABS.out`:

- Một dòng duy nhất là giá trị nhỏ nhất của S .

Ràng buộc bổ sung:

- 50% số điểm có $n \leq 5000, a_i \leq 5000$.
- 50% số điểm còn lại không có giới hạn gì thêm.

Ví dụ:

ABS.inp	ABS.out	Giải thích
5 1 2 4 8 9	3	Phương án tối ưu là chọn $x = 5$ hoặc $x = 6$. Trường hợp chọn $x = 5$ thì giá trị nhận được là $4 - 1 = 3$, trường hợp chọn $x = 6$ thì giá trị nhận được là $5 - 2 = 3$.

Câu 3. Trò chơi con mực (4 điểm)

Mã nguồn	Dữ liệu vào	Dữ liệu ra	Thời gian	Bộ nhớ
SQUID.*	SQUID.inp	SQUID.out	1 giây	1024 MB

Chắc hẳn các bạn đã nghe đến *Squid Game* – một bộ phim rất nổi tiếng được lấy cảm hứng từ các trò chơi dân gian ở Hàn Quốc. Hôm nay chúng ta sẽ nói về biến thể của một trong những màn chơi của *Squid Game*...

Trên bàn chơi hiện tại đang có n cầu hình, tương ứng với một trong bốn loại cầu hình là hình tròn, hình tam giác, hình ngôi sao và hình cái dù. Nhiệm vụ của bạn là chọn ra một số cầu hình trong n cầu hình trên bàn sao cho trong 4 cầu hình liên tiếp được chọn không được tồn tại đủ cả 4 cầu hình khác nhau. Hay nói cách khác, bạn cần:

- Chọn một tập con gồm k cầu hình ở các vị trí i_1, i_2, \dots, i_k ($1 \leq i_1 < i_2 < \dots < i_k \leq n$) sao cho với mọi giá trị j thỏa mãn $1 \leq j \leq k - 3$ thì các cầu hình ở các

vị trí $i_j, i_{j+1}, i_{j+2}, i_{j+3}$ không được đôi một phân biệt. Lưu ý là nếu $k \leq 3$ thì tập được chọn luôn thỏa mãn điều kiện và không được chọn tập rỗng.

Vì yêu cầu trên có vẻ quá dễ nên bạn cần phải đếm số cách chọn thỏa mãn điều kiện trên. Kết quả có thể rất lớn nên hãy xuất kết quả sau khi chia lấy dư cho 998244353.

Dữ liệu vào: Nhập từ file `SQUID.inp`:

- Dòng đầu tiên gồm một số nguyên dương n là số cấu hình trên bàn ($1 \leq n \leq 2 \times 10^5$).
- Dòng thứ hai gồm n số nguyên dương a_i tương ứng với các cấu hình trên bàn, a_i có giá trị từ 1 đến 4, trong đó các giá trị $a_i = 1, 2, 3, 4$ tương ứng lần lượt với hình tròn, hình tam giác, hình ngôi sao và hình cái dù.

Dữ liệu ra: Ghi ra file `SQUID.out`:

- Một dòng duy nhất là số cách chọn cấu hình thỏa mãn. Vì kết quả có thể rất lớn nên hãy xuất kết quả sau khi chia lấy dư cho 998244353.

Ràng buộc bổ sung:

- 30% số điểm có $n \leq 20$.
- 30% số điểm khác có $a_i \leq 3$.
- 40% số điểm còn lại không có giới hạn gì thêm.

Ví dụ:

SQUID.inp	SQUID.out	Giải thích
4 1 2 3 4	14	Có 15 cách chọn tập con, trong đó chỉ có cách chọn toàn bộ tập cấu hình là cách không thỏa mãn.
5 4 2 1 2 3	29	Có 25 cách chọn từ 1 đến 3 cấu hình, có 3 cách để chọn 4 cấu hình và 1 cách chọn 5 cấu hình. Vậy đáp án là $25 + 3 + 1 = 29$.

Câu 4. Một đời liêm khiết (4 điểm)

Mã nguồn	Dữ liệu vào	Dữ liệu ra	Thời gian	Bộ nhớ
LIEM.*	LIEM.inp	LIEM.out	1 giây	1024 MB

Dựa trên một câu chuyện không có thật...

Năm 2093, SQRTTOJ đã trở thành một OJ rất phổ biến với các bạn trẻ theo đuổi con đường CP (hay là Chillpetitive Programming). Với số lượng bài toán lên đến 6969 bài và 420 contest, thuật toán đa dạng đã ghi điểm rất lớn đến người dùng. Giờ có thể coi như là nhà nhà ai có con học CP cũng dùng SQRTTOJ và nó đã trở thành OJ quốc dân.

Các admin SQRTTOJ hiện tại đang có rất nhiều kế hoạch trong năm (dạy học, ra các kỳ thi pre-HSG và pre-TS10, SQRTTOJ CUP lần thứ 69, ...) nên họ đã chọn ra một đội ngũ ra đề nhằm mục đích làm ra những contest chất lượng cho các kế hoạch tương lai của họ. Nhưng các admin không biết đội ngũ họ chọn đều là những người không theo chủ nghĩa “một đời liêm khiết” và không làm theo 5 điều anh Liêm dạy. Cách ra đề của các thành viên này không giống cách ra đề bình thường. Theo quy trình bình thường thì để làm ra 1 bài toán sẽ là tìm cảm hứng và ý tưởng cho bài, viết bài, suy nghĩ và code sau đó debug chính bài của mình, sau đó là sinh test và đăng bài. Nhưng thay vì họ tìm cảm hứng và ý tưởng cho bài thì họ lại tìm và lấy bài trên các OJ khác, thay vì viết bài thì họ xào nấu lại nội dung đề hoặc dịch lại đề nếu như đó là bài nước ngoài, thay vì ngồi code và debug đến chết thì họ có thể bỏ qua hoặc chăm hơn là đọc thuật chuẩn và code lại, trường hợp lười quá thay vì sinh test thì “xin” luôn bộ test từ bài gốc. Cách này tuy nhanh vì có thể lấy được những bài chất lượng nhưng còn về chất lượng đề và test có ổn không thì còn phải xem xét lại. Suy cho cùng thì họ là đại diện cho những người LIEM nhưng mà là LIEM H... (mình xin bỏ qua phần này), cơ bản thì những người này từ nhỏ vì không được ăn uống đầy đủ hoặc vì lý do nào đó nên họ rất thiếu chất xám.

Thời gian sắp tới, các admin SQRTTOJ sẽ trở lại với series các kỳ thi pre-HSG nên đội ngũ ra đề cần chuẩn bị rất nhiều contest. Trong tầng hầm của họ đã có n trang OJ có thể chọn bài, với trang OJ thứ i có s_i bài có sẵn. Hiện tại họ có hai cách để tạo ra bộ đề mới:

- Với mọi i từ 1 đến n , chọn a_i bài từ trang OJ thứ i .
- Với mọi i từ 1 đến n , chọn b_i bài từ trang OJ thứ i .

Lưu ý rằng khi chọn bài từ một trang OJ nào đó, họ sẽ không bao giờ chọn lại bài đã lấy trước đó vì sẽ bị admin nghi ngờ. Ngoài ra, đội ngũ ra đề còn “rất uy tín trong việc chọn bài” cụ thể là không cân bằng được đề từ bài mình đã chọn nên có thể mặc định rằng: Khi chọn k bài từ OJ thứ i thì sau đó chỉ còn $s_i - k$ bài có thể chọn trong OJ đó. Nhiệm vụ của họ hiện tại là cần tối đa hóa số contest có thể tạo được, nhưng vì họ chỉ đủ trình để bú bài chứ không đủ trình CP nên phải nhờ các bạn giải quyết vấn đề hóc búa này vậy.

Dữ liệu vào: Nhập từ file LIEM.inp:

- Dòng đầu tiên gồm một số nguyên dương t là số trường hợp thử nghiệm ($1 \leq t \leq 10$). Với mỗi trường hợp thử nghiệm:
 - Dòng đầu tiên gồm một số nguyên dương n là số trang OJ có thể chọn bài ($1 \leq n \leq 5 \times 10^4$).
 - Dòng thứ hai gồm n số nguyên không âm s_1, s_2, \dots, s_n tương ứng với số bài có thể chọn với mỗi trang OJ ($0 \leq s_i \leq 10^9$).
 - Dòng thứ ba gồm n số nguyên không âm a_1, a_2, \dots, a_n tương ứng với số bài họ sẽ chọn với mỗi trang OJ trong phương án thứ nhất ($0 \leq a_i \leq 10^9, a_1 + a_2 + \dots + a_n > 0$).

- Dòng thứ tư gồm n số nguyên không âm b_1, b_2, \dots, b_n tương ứng với số bài họ sẽ chọn với mỗi trang OJ trong phương án thứ hai ($0 \leq b_i \leq 10^9, b_1 + b_2 + \dots + b_n > 0$).

Dữ liệu ra: Ghi ra file LIEM.out:

- In ra t dòng, mỗi dòng gồm duy nhất một số nguyên không âm tương ứng với kết quả của trường hợp thử nghiệm tương ứng.

Ràng buộc bổ sung:

- 25% số điểm có $n \leq 69; a_i, b_i, s_i \leq 420$.
- 25% số điểm khác có $n \leq 420; a_i, b_i, s_i \leq 2025$.
- 25% số điểm khác có $n \leq 2025; a_i, b_i, s_i \leq 6942$.
- 25% số điểm còn lại không có giới hạn gì thêm.

Ví dụ:

LIEM.inp	LIEM.out	Giải thích
2 2 6 9 0 1 1 0 5 9 6 9 6 9 1 2 1 3 1 2 1 1 2 3	15 3	<p>Với trường hợp thử nghiệm thứ nhất: Họ có thể tổ chức 9 contest từ phương án thứ nhất và 6 contest từ phương án thứ hai, như vậy tổng cộng có 15 contest có thể tổ chức, cũng là số contest tối đa.</p> <p>Với trường hợp thử nghiệm thứ hai: Gọi a là số contest có thể tổ chức từ phương án thứ nhất, b là số contest có thể tổ chức từ phương án thứ hai. Khi đó các cách thỏa mãn bao gồm các trường hợp sau:</p> <ul style="list-style-type: none"> - $a = 0, b = 0$. - $a = 0, b = 1$. - $a = 0, b = 2$. - $a = 0, b = 3$. - $a = 1, b = 0$. - $a = 1, b = 1$. - $a = 2, b = 0$. <p>Trong tất cả cách trên, phương án tối ưu là chọn $a = 0, b = 3$ có tổng $a + b = 3$ là số contest tối đa có thể tạo.</p>

Câu 5. Kể về nhì (4 điểm)

Mã nguồn	Dữ liệu vào	Dữ liệu ra	Thời gian	Bộ nhớ
SECOND.*	SECOND.inp	SECOND.out	1 giây	1024 MB

Bạn đã bao giờ trải nghiệm cảm giác về nhì chưa? Nếu rồi thì mình chắc chắn các bạn cũng đã từng tự dằn vặt bản thân rằng: “Chỉ còn một chút thôi mà...” và sau đó lại nghi vấn về trình độ bản thân. Không sao, bởi vì mình cũng từng như vậy... Nhưng dù gì thì mình biết các bạn đã rất cố gắng rồi, vì thế hãy vẫn luôn giữ ngọn lửa đam mê trong lòng các bạn nhé. Đừng dằn vặt gì mà hãy tiếp tục theo đuổi con đường mà bạn chọn, rồi thành công sẽ đến với bạn sớm thôi ...

Nói vậy là đủ rồi, hôm nay mình có một bài toán dành cho bạn như sau:

Cho hai số nguyên dương l, r . Gọi $f(x)$ là chữ số lớn thứ nhì của x . Hãy tính $f(l) + f(l+1) + \dots + f(r)$. Vì kết quả có thể rất lớn nên hãy xuất kết quả sau khi chia lấy dư cho $7 \times 10^9 + 1$.

Dữ liệu vào: Nhập từ file `SECOND.inp`:

- Dòng đầu tiên gồm một số nguyên dương t là số trường hợp thử nghiệm ($1 \leq t \leq 15 \times 10^4$).
- t dòng tiếp theo, mỗi dòng gồm hai số nguyên dương l, r ($10 \leq l \leq r < 10^{18}$).

Dữ liệu ra: Ghi ra file `SECOND.out`:

- Gồm t dòng, dòng thứ i gồm duy nhất một số là kết quả của trường hợp thử nghiệm thứ i sau khi chia lấy dư cho $7 \times 10^9 + 1$.

Ràng buộc bổ sung:

- 30% số điểm có $10 \leq l \leq r \leq 10^6$.
- 40% số điểm khác có $t \leq 69, r - l \leq 10^6$.
- 30% số điểm còn lại không có giới hạn gì thêm.

Ví dụ:

SECOND.inp	SECOND.out	Giải thích
2 12 24 1023 1030	15 15	Với trường hợp thử nghiệm thứ nhất: Các số từ 12 đến 19 có chữ số nhỏ thứ nhì là 1. Số 20 có chữ số nhỏ thứ nhì là 0. Số 21 có chữ số nhỏ thứ nhì là 1. Các số từ 22 đến 24 có chữ số nhỏ thứ nhì là 2. Vậy đáp án cho trường hợp thử nghiệm thứ nhất là $1 \times 8 + 0 + 1 + 2 \times 3 = 15$.

--- Hết ---

Thí sinh không được sử dụng tài liệu.

Giám thị coi thi không giải thích gì thêm.