

1. Trò chơi với dãy số (NKSGAME.*) – VOI08

Hai bạn học sinh trong lúc nhàn rỗi nghĩ ra trò chơi sau đây. Mỗi bạn chọn trước một dãy số gồm n số nguyên. Giả sử dãy số mà bạn thứ nhất chọn là: b_1, b_2, \dots, b_n còn dãy số mà bạn thứ hai chọn là c_1, c_2, \dots, c_n .

Mỗi lượt chơi mỗi bạn đưa ra một số hạng trong dãy số của mình. Nếu bạn thứ nhất đưa ra số hạng b_i ($1 \leq i \leq n$), còn bạn thứ hai đưa ra số hạng c_j ($1 \leq j \leq n$) thì giá của lượt chơi đó sẽ là $|b_i + c_j|$.

Ví dụ: Giả sử dãy số bạn thứ nhất chọn là 1, -2; còn dãy số mà bạn thứ hai chọn là 2, 3. Khi đó các khả năng có thể của một lượt chơi là (1, 2), (1, 3), (-2, 2), (-2, 3). Như vậy, giá nhỏ nhất của một lượt chơi trong số các lượt chơi có thể là 0 tương ứng với giá của lượt chơi (-2, 2).

Yêu cầu: Hãy xác định giá nhỏ nhất của một lượt chơi trong số các lượt chơi có thể.

Dữ liệu

- Dòng đầu tiên chứa số nguyên dương n ($n \leq 10^5$)
- Dòng thứ hai chứa dãy số nguyên b_1, b_2, \dots, b_n ($|b_i| \leq 10^9, i = 1, 2, \dots, n$)
- Dòng thứ hai chứa dãy số nguyên c_1, c_2, \dots, c_n ($|c_j| \leq 10^9, i=1, 2, \dots, n$)

Hai số liên tiếp trên một dòng được ghi cách nhau bởi dấu cách.

Kết quả: Ghi ra giá nhỏ nhất tìm được.

Ràng buộc: 60% số tests ứng với 60% số điểm của bài có $1 \leq n \leq 1000$.

Ví dụ:

NKSGAME.INP	NKSGAME.OUT
2	0
1 -2	
2 3	

2. Lát nền - Tên file chương trình 'PAVING.*'

Một công ty xây dựng muốn lát nền cho công trình xây dựng của mình bằng các tấm ván gỗ. Có n loại ván có độ rộng bằng nhau và độ dài chuẩn khác nhau tương ứng lần lượt là $a_1, a_2, a_3, \dots, a_n$.

Công trình này bao gồm n hàng ván được ghép lại với nhau, hàng thứ i người ta đặt tấm ván có độ dài chuẩn a_i làm mốc (đặt sát mép tường). Trong việc thi công hàng ván thứ i người ta chỉ ghép các tấm ván có độ dài a_i theo phương của độ dài, không được

phép ghép tấm ván có độ dài khác vào hàng. Vậy, để n hàng ván có độ dài bằng nhau và đảm bảo cách ghép như trên thì người ta cần phải có bao nhiêu tấm ván của từng loại.

Yêu cầu: Hãy cho biết cần bao nhiêu tấm ván chuẩn của từng loại để n hàng ván của công trình có độ dài bằng nhau.

Dữ liệu vào từ file ‘PAVING.INP’ gồm:

- Dòng thứ nhất là số nguyên dương n ($n \leq 35$).
- Dòng thứ hai chứa n số nguyên dương a_i , mỗi số cách nhau một khoảng trắng ($a_i \leq 100$).

Kết quả ghi vào file ‘PAVING.OUT’ số lượng tấm ván của từng loại ván chuẩn tương ứng của từng hàng (mỗi số trên 1 dòng).

Ví dụ:

PAVING.INP
3
3 4 2

PAVING.OUT
4
3
6

Theo ví dụ trên, có 3 độ dài chuẩn của các loại ván tương ứng lần lượt là 3 4 2 thì: Hàng thứ 1 cần 4 tấm ván có độ dài 3, hàng thứ 2 cần 3 tấm ván có độ dài 6, hàng thứ 3 cần 6 tấm ván có độ dài 2.

3. ÉCH NHẢY 1 – AFROG.*

Có N bậc đá, được đánh số từ 1, 2, ... N . Mỗi bậc đá i ($1 \leq i \leq N$) có độ cao h_i

Một con ếch đang ở bậc đá thứ nhất. Nó có thể nhảy lên bậc đá tiếp theo bằng 2 cách:

- Nếu nó ở hòn đá thứ i thì nó có thể nhảy đến hòn đá thứ tự $i + 1$, hoặc hòn đá thứ $i + 2$
- Chi phí cho 1 lần nhảy là $|h_j - h_i|$ với h_j là hòn đá đích mà ếch nhảy tới.

Yêu cầu: Tìm chi phí thấp nhất để giúp ếch nhảy đến hòn đá thứ N .

Ràng buộc:

- Các giá trị là số nguyên.
- $2 \leq N \leq 10^5$
- $1 \leq h_i \leq 10^4$

Dữ liệu: đọc vào từ file AFrog.inp gồm:

- N – số hòn đá

- Dòng tiếp theo là chiều cao của N hòn đá.

Kết quả: ghi ra file AFrog.out

- Tổng chi phí phát sinh tối thiểu.

Ví dụ:

AFrog.inp	AFrog.out	
4 10 30 40 20	30	$1 \rightarrow 2 \rightarrow 4$, tổng chi phí phát sinh sẽ là $ 10-30 + 30-20 =30$.
6 30 10 60 10 60 50	40	$1 \rightarrow 3 \rightarrow 5 \rightarrow 6$, tổng chi phí phát sinh sẽ là $ 30-60 + 60-60 + 60-50 =40$.

4. ÉCH NHẢY B – BFROG.*

Có N bậc đá, được đánh số từ 1, 2, ...N. Mỗi bậc đá $i (1 \leq i \leq N)$ có độ cao h_i

Một con ếch đang ở bậc đá thứ nhất. Nó có thể nhảy lên bậc đá tiếp theo các cách:

- Nếu nó ở hòn đá thứ i thì nó có thể nhảy đến hòn đá thứ tự $i + 1, i + 2, \dots, i + k$,
- Chi phí cho 1 lần nhảy là $|h_j - h_i|$ với h_j là hòn đá đích mà ếch nhảy tới.

Yêu cầu: Tìm chi phí thấp nhất để giúp ếch nhảy đến hòn đá thứ N.

Ràng buộc:

- Các giá trị là số nguyên.
- $2 \leq N \leq 10^5$
- $1 \leq K \leq 100$
- $1 \leq h_i \leq 10^4$

Dữ liệu: đọc vào từ file BFrog.inp gồm:

- N – số hòn đá, K – số bậc đá tối đa con ếch có thể nhảy qua
- Dòng tiếp theo là chiều cao của N hòn đá.

Kết quả: ghi ra file BFrog.out

- Tổng chi phí phát sinh tối thiểu.

Ví dụ:

BFrog.inp	BFrog.out	
5 3 10 30 40 50 20	30	$1 \rightarrow 2 \rightarrow 5$, tổng chi phí phát sinh sẽ là $ 10-30 + 30-20 =30$.
10 4 40 10 20 70 80 10 20 70 80 60	40	$1 \rightarrow 4 \rightarrow 8 \rightarrow 10$, tổng chi phí phát sinh sẽ là $ 40-70 + 70-70 + 70-60 =40$.

5. NGHỈ HÈ – CSUMMER.*

Kỳ nghỉ hè của Nam bắt đầu vào ngày mai, và anh ấy đã quyết định lên kế hoạch cho kỳ nghỉ ngay bây giờ.

Nam được nghỉ N ngày. Với mỗi ngày nghỉ thứ i ($1 \leq i \leq N$), Nam sẽ chọn một trong các hoạt động sau và thực hiện để được tích điểm khen thưởng:

- A: Bơi ở biển. Thu được a_i điểm.
- B: Bắt sâu cho vườn hoa của mẹ. Thu được b_i điểm.
- C: Làm bài tập ở nhà. Thu được c_i điểm.

Vốn dễ chán nên Nam không thể thực hiện các hoạt động giống nhau trong hai ngày liên tiếp.

Yêu cầu: Tìm tổng điểm khen thưởng tối đa mà Nam có thể đạt được.

Ràng buộc

- Tất cả các giá trị trong đầu vào là số nguyên.
- $1 \leq N \leq 10^5$
- $1 \leq a_i, b_i, c_i \leq 10^4$

Dữ liệu: Đọc vào từ file CSUMMER.INP gồm:

Dòng đầu tiên chứa N – số ngày nghỉ hè

N dòng tiếp theo chứa các giá trị điểm a_i, b_i, c_i tương ứng với ngày thứ i

Kết quả: Ghi ra file CSUMMER.OUT: Tổng điểm hạnh phúc tối đa mà Nam có thể đạt được.

Ví dụ:

CSUMMER.INP	CSUMMER.OUT	
3 10 40 70 20 50 80 30 60 90	210	Nếu Nam thực hiện các hoạt động theo thứ tự C, B, C, anh ta sẽ đạt được $70+50+90=210$ điểm.
7 6 7 8 8 8 3 2 5 2 7 8 6 4 6 8	46	

2 3 4		
7 5 1		

6. DÂY CON ĐƠN ĐIỀU TĂNG DÀI NHẤT.

Cho một dãy số nguyên gồm N phần tử $A[1], A[2], \dots, A[N]$. Biết rằng dãy con tăng đơn điệu là 1 dãy $A[i_1], \dots, A[i_k]$ thỏa mãn $i_1 < i_2 < \dots < i_k$ và $A[i_1] < A[i_2] < \dots < A[i_k]$. Hãy cho biết dãy con tăng đơn điệu dài nhất của dãy này có bao nhiêu phần tử.

- Dữ liệu vào: Chứa trong tệp “DAYCON.INP”.
 - Dòng 1 gồm 1 số nguyên là số N ($1 \leq N \leq 1000$).
 - Dòng thứ 2 ghi N số nguyên $A[1], A[2], \dots, A[N]$ ($1 \leq A[i] \leq 10000$).
- Dữ liệu ra: Chứa trong tệp “DAYCON.OUT”.
 - Dòng 1 là độ dài của dãy con tăng dài nhất.
 - Dòng 2 là các phần tử của dãy con tăng dài nhất, mỗi phần tử cách nhau 1 khoảng trắng. Ví dụ:

DAYCON.INP	DAYCON.OUT
6	4
1 2 5 4 6 2	1 2 4 6

7. LIS.*

Cho dãy A_1, A_2, \dots, A_n . Hãy tìm một dãy con tăng ngặt dài nhất của dãy (dãy con có thể không liên tiếp).

Dữ liệu: Vào từ file LIS.INP gồm:

- Dòng đầu tiên chứa số nguyên N ($1 \leq N \leq 10^5$)
- Dòng thứ hai chứa N số nguyên a_i .

Kết quả: Ghi ra file LIS.OUT: một số nguyên là kết quả bài toán.

Ví dụ:

LIS.INP	LIS.OUT
5	3
1 3 2 2 5	

8. Mua vé xem phim – TICKET.*

Tự hào vì đã được các bạn học sinh giỏi tin học chứng minh mình làm đúng bài toán của Mai và cũng nhân dịp ngày 20/10 - ngày Phụ nữ Việt Nam sắp đến, Dũng quyết định rủ Mai đi xem phim. Tuy nhiên Mai là cô gái khó tính, cô chỉ muốn ngồi xem ở hàng ghế M - chữ cái đầu tiên của mình. Hơn nữa, Mai lại muốn rủ các bạn gái của mình cùng đi

xem - mà số bạn gái của cô nhiều vô kể vì cô vốn là một bí thư Đoàn nổi tiếng trong trường. Cô muốn tất cả mọi người phải ngồi xem ở một dãy ghế liên tục ở hàng M và muốn Dũng mời nhiều người đi xem phim nhất có thể.

Khi đi mua vé, một số ghế ở hàng M đã có người mua vé trước đó, Dũng chỉ được chọn mua vé trong số các ghế trống còn lại. Vì không muốn mua nhiều vé do sợ không đủ tiền nên Dũng quyết định mua vé toàn bộ một đoạn dãy ghế liên tục chưa được mua vé (không còn ghế chưa được mua vé nào ở hai đầu đoạn này) có tối thiểu hai chỗ và tất nhiên số lượng ghế trong đoạn phải ít nhất. Lần này Dũng quyết định nhờ đến các bạn học sinh giỏi tin học.

Yêu cầu: Biết hàng M có n ghế đánh số lần lượt $1, 2, \dots, n$ từ trái sang phải. Tình trạng ghế trước khi Dũng mua vé được mô tả bằng dãy số nguyên a_1, a_2, \dots, a_n trong đó $a_i = 0$ nếu ghế i còn trống và $a_i = 1$ nếu ghế i đã được mua vé trước đó ($i = 1, 2, \dots, n$). Hãy tính số lượng vé tối thiểu mà Dũng phải mua vé.

Dữ liệu: Nhập từ bàn phím lần lượt các giá trị sau:

- Số nguyên dương n ($n \leq 10^5$)
- Lần lượt các số nguyên a_1, a_2, \dots, a_n ($a_i \in \{0, 1\} \forall i = 1, 2, \dots, n$)

Dữ liệu đảm bảo rằng Dũng luôn có thể mua được vé.

Kết quả: In ra màn hình một số nguyên - số lượng vé tối thiểu mà Dũng phải mua.

Ví dụ:

TICKET.INP	TICKET.OUT
10 1 0 0 0 1 0 0 0 0 1	3

Giải thích: Dũng có hai lựa chọn là hoặc mua 3 vé ở các số ghế 2, 3 và 4, hoặc mua 4 vé ở các số ghế 6, 7, 8 và 9. Tất nhiên phương án lựa chọn là phương án chỉ mua 3 vé.

9. Hàng cây - Tên file chương trình TREES.*

Thành phố Vũng Tàu nổi tiếng với cung đường ven biển tuyệt đẹp, dọc cung đường này người ta trồng các cây phi lao để chắn gió, giữ đất. Sau một thời gian, các cây phi lao đều phát triển nhanh và đẹp, tuy nhiên chiều cao của chúng không đồng đều vì vậy người ta quyết định loại bỏ bớt một số cây để cho hàng phi lao đẹp hơn.

Giả sử có N cây phi lao được đánh số thứ tự từ 1 tới N tính từ đầu tới cuối cung đường. Cây thứ i có chiều cao là a_i . Trong N cây này người ta chọn một cây

có thứ tự là K làm chuẩn, sau đó chọn các cây ở phía hai bên của cây thứ K theo nguyên tắc sau:

- Với các cây có thứ tự nhỏ hơn K , cây được chọn gần cây thứ K phải có chiều cao lớn hơn cây được chọn xa cây thứ K .

- Với các cây có thứ tự lớn hơn K , cây được chọn gần cây thứ K phải có chiều cao nhỏ hơn cây được chọn xa cây thứ K .

Yêu cầu: Em hãy viết chương trình giúp lựa chọn các cây, sao cho số cây phải loại bỏ là ít nhất (Cây thứ K luôn được chọn).

Dữ liệu vào: Từ file **TREES.INP**

- Dòng đầu tiên là hai số nguyên dương N, K cách nhau một khoảng trắng ($K < N \leq 10^4$)
- N dòng tiếp theo mỗi dòng một số nguyên a_i là chiều cao của cây thứ i ($a_i \leq 10^6$)

Dữ liệu ra: Ghi vào file **TREES.OUT** một số nguyên duy nhất là số cây cần loại bỏ.

Ví dụ:

TREES.INP	TREES.OUT
9 5	4
3	
7	
2	
4	
6	
4	
7	
5	
8	

10. Trò chơi xếp hình - Tên chương trình: GAMEP.*

An vừa được Bố mua tặng một bộ đồ chơi block ghép hình sáng tạo. Trong bộ đồ chơi có n block, các block được đánh thứ tự lần lượt từ 1 đến n , block thứ i có độ cao là a_i (với $i=1, 2, \dots, n$). An thích trò chơi xếp các block thành một hình trụ để mô phỏng trụ phát sóng của viễn thông. Để tăng tính thẩm mỹ, An chỉ thích chọn ra một loại block có cùng độ cao để xếp thành một trụ đứng nhưng không biết chọn loại block nào để xếp được một trụ cao nhất (với số lượng block ít nhất nếu có).

Yêu cầu: Hãy giúp An chọn loại block để xếp hình trụ theo yêu cầu của em.

Dữ liệu: vào từ file **GAMEP.INP**

- Dòng đầu chứa số nguyên dương n ($n \leq 10^6$).
- Trong n dòng sau, dòng thứ i chứa số nguyên dương a_i ($a_i \leq 10^6$; $i=1, 2, \dots, n$).

Kết quả: ghi vào file **GAMEP.OUT** một dòng duy nhất ghi 2 số lần lượt là **số lượng các block** và **chiều cao của loại block được chọn** (2 số cách nhau ít nhất 1 khoảng trắng). **Nếu có nhiều loại block xếp thành hình trụ có cùng độ cao lớn nhất thì ghi ra loại có ít block nhất.**

Ví dụ:

GAMEP.INP	GAMEP.OUT
6	3 4
4	
2	
3	
4	
11	
4	

Giải thích: chọn loại block có độ cao 4, với số lượng 3, tổng chiều cao là 12

11. Chọn tiền xu 1 – Coins1.*

Ở đất nước ABC có một hệ thống tiền bao gồm n đồng xu. Mỗi đồng xu có một giá trị nguyên dương. Giả sử số lượng các đồng xu là vô hạn

Yêu cầu: Tính số cách mà bạn có thể tạo ra một khoản tiền x bằng cách sử dụng các đồng tiền có sẵn trong hệ thống tiền tệ.

Ví dụ, nếu ta có các loại tiền xu $\{2, 3, 5\}$ và số tiền mong muốn là 9, có số 8 cách chọn các đồng xu như sau:

$$2 + 2 + 5$$

$$2 + 5 + 2$$

$$5 + 2 + 2$$

$$3 + 3 + 3$$

$$2 + 2 + 2 + 3$$

$$2 + 2 + 3 + 2$$

$$2 + 3 + 2 + 2$$

$$3 + 2 + 2 + 2$$

Dữ liệu vào: từ file **Coins1.inp** gồm

- Đầu tiên có hai số nguyên n và x : số xu và số tiền mong muốn.
- Dòng thứ hai có n số nguyên riêng biệt c_1, c_2, \dots, c_n : giá trị của mỗi đồng xu.

Kết quả: ghi ra file Coins1.out

- In một số nguyên: số cách modulo $10^9 + 7$.

Ràng buộc

- $1 \leq n \leq 100$
- $1 \leq x \leq 10^6$
- $1 \leq c_i \leq 10^6$

Ví dụ:

Coins1.inp	Coins1.out
3 9	8
2 3 5	

12. Chọn tiền xu 2 – Coins2.*

Ở đất nước ABC có một hệ thống tiền bao gồm n đồng xu. Mỗi đồng xu có một giá trị nguyên dương. Giả sử số lượng các đồng xu là vô hạn.

Yêu cầu: Tính số cách mà bạn có thể tạo ra một khoản tiền x (với cách chọn khác nhau) bằng cách sử dụng các đồng tiền có sẵn.

Ví dụ, nếu ta có các loại tiền xu $\{2, 3, 5\}$ và số tiền mong muốn là 9, có số 3 cách chọn các đồng xu như sau:

$$2 + 2 + 5$$

$$3 + 3 + 3$$

$$2 + 2 + 2 + 3$$

Dữ liệu vào: từ file Coins2.inp gồm

- Đầu tiên có hai số nguyên n và x : số xu và số tiền mong muốn.
- Dòng thứ hai có n số nguyên riêng biệt c_1, c_2, \dots, c_n : giá trị của mỗi đồng xu.

Kết quả: ghi ra file Coins2.out

- In một số nguyên: số cách modulo $10^9 + 7$.

Ràng buộc

- $1 \leq n \leq 100$
- $1 \leq x \leq 10^6$
- $1 \leq c_i \leq 10^6$

Ví dụ

Coins2.inp	Coins2.out
3 9 2 3 5	3