

Bài 1: PHÂN TÍCH SỐ

Cho số tự nhiên $n \leq 100$. Hãy cho biết có bao nhiêu các phân tích số n thành tổng của dãy các số nguyên dương, các cách phân tích là hoán vị của nhau chỉ tính là một cách.

Ví dụ, với $n = 5$, ta có 7 cách phân tích:

- $5 = 1 + 1 + 1 + 1 + 1$
 - $5 = 1 + 1 + 1 + 2$
 - $5 = 1 + 1 + 1 + 3$
 - $5 = 1 + 2 + 2$
 - $5 = 1 + 4$
 - $5 = 2 + 3$
 - $5 = 5$
- Dữ liệu vào: Chứa trong tệp “PHANTICH.INP”.
 - Gồm 1 dòng duy nhất là số n .
 - Dữ liệu ra: Chứa trong tệp “PHANTICH.OUT”.
 - Gồm 1 dòng duy nhất là số các phân tích số n thành tổng của dãy các số nguyên dương. Ví dụ:

PHANTICH.INP	PHANTICH.OUT
5	7

Bài 2: HÌNH VUÔNG LỚN NHẤT

Cho một bảng $m \times n$ ($0 < m, n \leq 1000$) các ô nhỏ chứa các số 0 và 1. Hãy tìm hình vuông chứa toàn số 1 và có diện tích lớn nhất.

- Dữ liệu vào: chứa trong tệp “HINHUVUONG.INP”.
 - Dòng đầu tiên chứa 2 số m và n .
 - m dòng tiếp theo chứa các số 0 và 1 là biểu diễn của bảng.
- Dữ liệu ra: chứa trong tệp “HINHUVUONG.OUT”.
 - Một dòng chứa diện tích lớn nhất của hình vuông tìm được.

Ví dụ:

HINHUVUONG.INP	HINHUVUONG.OUT
4 5 0 1 1 0 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1	9

Bài 3: Dãy con đơn điệu 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

Bài 4: TỔ HỢP

Biết công thức khai triển nhị thức Newton: $(a + b)^n = \sum_{k=0}^n C_n^k a^k b^{n-k}$. Cho biết n và k , hãy tính C_n^k .

- Dữ liệu vào: Chứa trong tệp “TOHOP.INP”.
 - Gồm 1 dòng chứa 2 số nguyên n và k ($0 < k \leq n \leq 1000$).
- Dữ liệu ra: Chứa trong tệp “TOHOP.OUT”.
 - Gồm 1 dòng duy nhất chứa kết quả C_n^k .

Ví dụ:

TOHOP.INP	TOHOP.OUT
4 2	6

BÀI 5: HANSTAND.*

Có N người xếp thành một hàng từ trái qua phải.

Bạn được cung cấp một chuỗi S chiều dài n bao gồm 2 số 0 và 1, và một số nguyên dương K tương ứng với N người xếp hàng. Người thứ i tính từ bên trái đang đứng trên đôi chân nếu ký tự i tương ứng trong S là 0, và đứng trên tay (trông cây chuối) nếu ký tự đó là 1.

Bạn được đưa ra chỉ dẫn để thực hiện K lần như sau (có thể bằng không):

- + Chọn số nguyên l và r sao cho $1 \leq l \leq r \leq n$ và lật theo thứ tự từ người thứ $l, (l + 1), \dots, r$, và người thứ r . Nghĩa là, với mỗi $i = l, l + 1, \dots, r$, người thứ i từ bên trái bây giờ đứng trên tay nếu anh ấy / cô ấy đang đứng bằng chân, và đứng trên đôi chân nếu anh ấy/cô ấy đang đứng bằng tay.

Tìm số người **liên tiếp** đứng trên tay nhiều nhất có thể có sau K lần.

Giới hạn:

- N là một số nguyên thỏa mãn $1 \leq n \leq 10^5$.
- K là một số nguyên thỏa mãn $1 \leq K \leq 10^5$.
- Chiều dài của chuỗi S là n .
- Mỗi ký tự của chuỗi S là 0 hoặc 1.

Dữ liệu vào: từ file Hanstand.inp gồm

- Dòng đầu tiên chứa 2 số nguyên n và K
- Dòng thứ 2 là chuỗi S tương ứng

Kết quả: Ghi ra file Hanstand.out: In số người đứng trên tay liên tiếp tối đa có thể có sau nhiều nhất K lần.

Ví dụ:

Hanstand.inp	Hanstand.out
5 1 00010	4
14 2 11101010110011	8
1 1 1	1

Giải thích: Chúng ta có thể có bốn người liên tiếp đứng trên tay, đó là kết quả tối đa, bằng cách đưa ra cách lật như sau:

- Với $l = 1$, $r = 3$, lật người thứ nhất, thứ hai và thứ ba từ bên trái.

BÀI 6: MẢNG K – ARRAYK.*

Bạn được cung cấp một chuỗi các số nguyên dương có độ dài n , $A = a_1, a_2, \dots, a_N$ và một số nguyên K .

Yêu cầu: đếm xem bao nhiêu dãy con liên kề của A thỏa mãn điều kiện: Tổng các phần tử trong dãy con liên kề ít nhất là K .

Lưu ý: Coi hai dãy con liên kề là khác nhau nếu chúng xuất phát từ các vị trí khác nhau trong A , ngay cả khi chúng giống nhau về nội dung.

Ràng buộc:

- $1 \leq A_i \leq 10^5$
- $1 \leq n \leq 10^5$
- $1 \leq K \leq 10^{10}$

Dữ liệu: Đọc vào từ file ArrayK.inp gồm:

- Dòng đầu tiên chứa 2 số nguyên n và k

- Dòng tiếp theo chứa n số nguyên của mảng a

Kết quả: Ghi ra file ArrayK.out: số lượng các dãy con liên tiếp của A thỏa mãn điều kiện.

Ví dụ:

ArrayK.inp	ArrayK.out
4 10 6 1 2 7	2
3 5 3 3 3	3

Hai dãy con liên tiếp sau đây thỏa mãn điều kiện:

- $A[1..4] = a_1, a_2, a_3, a_4$ với tổng số 16
- $A[2..4] = a_2, a_3, a_4$, với tổng số 10

BÀI 7: ĐỌC SÁCH – READB.*

Giới hạn thời gian: 2 giây / Giới hạn bộ nhớ: 1024 MB

Ghi bàn :3000điểm

Báo cáo vấn đề

Chúng ta có hai cái bàn: A và B. Bàn A có một ngăn xếp thẳng đứng chứa n sách trên đó và Bàn B tương tự có M sách trên đó.

Giả sử có A_i phút để đọc cuốn sách thứ i từ trên cùng trên Bàn A ($1 \leq i \leq N$), và B_i phút để đọc cuốn sách thứ i từ trên cùng trên Bàn B ($1 \leq i \leq M$).

Hãy xem xét hành động sau:

- Chọn một bàn có sách còn lại, đọc cuốn sách trên cùng trên bàn đó và xóa nó khỏi bàn.

Yêu cầu: Chúng ta có thể đọc tối đa bao nhiêu cuốn sách bằng cách lặp lại hành động này để chúng ta có thể đọc nhiều nhất trong tổng K số phút? Chúng ta bỏ qua thời gian để làm bất cứ điều gì khác ngoài việc đọc.

Hạn chế:

$1 \leq N, M \leq 200000$

- $1 \leq K \leq 10^9$
- $1 \leq A_i, B_i \leq 10^9$
- Tất cả các giá trị trong đầu vào là số nguyên.

Đầu vào: Từ file ReadB.inp

- Dòng đầu tiên chứa 2 số n và m tương ứng với số sách trên bàn A và B
- Dòng thứ hai: là thời gian đọc các quyển sách tương ứng trên bàn A.
- Dòng thứ ba: là thời gian đọc các quyển sách tương ứng trên bàn B.

Kết quả: Ghi ra file ReadB.out: In một số nguyên biểu thị số sách tối đa có thể đọc được.

Ví dụ:

ReadB.inp	ReadB.out
3 4 240 60 90 120 80 150 80 150	3

Giải thích: Trong trường hợp này, chúng tôi phải 60, 90, 120 phút để đọc các cuốn sách thứ 1, 2, 3 ở trên cùng trên Bàn A, và 80, 150, 80, 150 phút để đọc các cuốn sách thứ 1, 2, 3, 4 từ trên cùng trên Bàn B, tương ứng.

Chúng ta có thể đọc ba cuốn sách trong 230 phút, như được hiển thị bên dưới và đây là số sách tối đa chúng tôi có thể đọc trong 240 phút.

- Đọc cuốn sách hay nhất trên Bàn A trong 60 phút, và xóa cuốn sách đó khỏi bàn làm việc.
- Đọc cuốn sách hay nhất trên Bàn B trong 80 phút, và xóa cuốn sách đó khỏi bàn làm việc.
- Đọc cuốn sách hay nhất trên Bàn A trong 90 phút, và xóa cuốn sách đó khỏi bàn làm việc.

Bài 8: 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$.

BÀI 9: ÉCH NHẢY B – BFROG.*

Có N bậc đá, được đánh số từ $1, 2, \dots, N$. 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$.

BÀI 10: 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 ai điểm.
- B: Bắt sâu cho vườn hoa của mẹ. Thu được bi điểm.
- C: Làm bài tập ở nhà. Thu được ci 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.

- $1 \leq N \leq 105$
- $1 \leq a_i, b_i, c_i \leq 104$

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 2 3 4 7 5 1	46	