
Problem NDCCARD

Input file: NDCCARD.INP
Output file: NDCCARD.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

Người tham gia trò chơi được cung cấp một tập gồm N lá bài và một số nguyên dương M . Nhiệm vụ của người chơi là phải chọn ra 3 lá bài từ tập lá bài đã cho sao cho tổng các số trên 3 lá bài đã chọn là lớn nhất và không vượt quá M .

Yêu cầu: Bạn hãy tìm kết quả tốt nhất có thể có của trò chơi trên.

Input

- Dòng đầu ghi số nguyên dương N, M ($N \leq 10000, M \leq 500000$).
- N dòng sau ghi N số nguyên dương đối với nhau là các số được ghi trên N lá bài ($1 \leq a[i] \leq 10000$).

Output

- Ghi trên một dòng duy nhất là kết quả bài toán. Test luôn đảm bảo có kết quả.

Example

NDCCARD.INP	NDCCARD.OUT
6 20	20
7 9 6 2 1 5	

Subtask

- Subtask 1 (30%): $N \leq 100$.
- Subtask 2 (40%): $N \leq 500$.
- Subtask 3 (30%): Không có giới hạn gì thêm.

Problem NKSGAME

Input file: NKSGAME.INP
Output file: NKSGAME.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

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.

Input

- 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 ba chứa dãy số nguyên c_1, c_2, \dots, c_n ($|c_i| \leq 10^9, i = 1, 2, \dots, n$)

Output

- Ghi ra giá nhỏ nhất tìm được

Example

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

Subtask

- Subtask 1 (60%): $1 \leq n \leq 1000$.
- Subtask 2 (40%): Không có giới hạn gì thêm.

Problem MTWALK

Input file: MTWALK.INP
Output file: MTWALK.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

Cho một bản đồ kích thước $N \times N$ ($2 \leq N \leq 100$), mỗi ô mang giá trị là độ cao của ô đó ($0 \leq \text{độ cao} \leq 110$). Bác John và bò Bessie đang ở ô trên trái (dòng 1, cột 1) và muốn đi đến cabin (dòng N , cột N). Họ có thể đi sang phải, trái, lên trên và xuống dưới nhưng không thể đi theo đường chéo. Hãy giúp bác John và bò Bessie tìm đường đi sao cho chênh lệch giữa điểm cao nhất và thấp nhất trên đường đi là nhỏ nhất.

Input

- Dòng đầu chứa số nguyên N .
- N dòng tiếp theo mỗi dòng chứa N số nguyên, mỗi số cho biết cao độ của một ô.

Output

- Một số nguyên là chênh lệch cao độ nhỏ nhất.

Example

MTWALK.INP	MTWALK.OUT
5 1 1 3 6 8 1 2 2 5 5 4 4 0 3 3 8 0 2 3 4 4 3 0 2 1	2

Subtask

- Subtask 1 (40%): $N \leq 10$
- Subtask 2 (60%): Không có giới hạn gì thêm.

Problem C11CAVE

Input file: C11CAVE.INP
Output file: C11CAVE.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

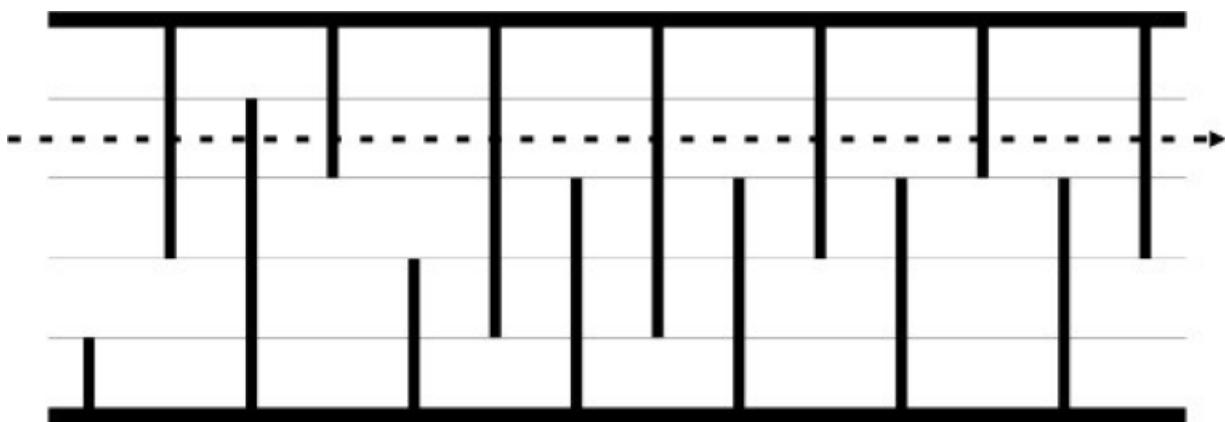
Một con đom đóm bay vào một cái hang đầy những chướng ngại vật gồm: mảng đá (nhô lên từ mặt đất) và nhũ đá (đâm xuống từ trần hang). Hang này dài N đơn vị (N chẵn) và cao H đơn vị. Khi vào hang, vật cản đầu tiên là mảng đá, sau đó là nhũ đá, rồi lại đến mảng đá, ... cứ thế thay phiên nhau.

Đây là một ví dụ về một hang dài 14 đơn vị và cao 5 đơn vị.



Con đom đóm này không phải là loài có thể bay quanh các chướng ngại vật. Thay vào đó, nó sẽ chọn một mức chiều cao bắt đầu rồi bay từ đầu đến cuối hang, phá hết tất cả các chướng ngại vật trên đường bay của nó.

Theo ví dụ trên, nếu chọn mức 4, con đom đóm sẽ phá tất cả 8 chướng ngại vật.



Đây không phải là lựa chọn tốt nhất vì con đom đóm sẽ ít mệt hơn nếu chọn mức 1 hoặc mức 5, lúc này nó chỉ cần phá 7 chướng ngại vật.

Bạn được cho chiều dài, chiều cao và kích thước của tất cả các chướng ngại vật. Hãy xác định số chướng ngại vật tối thiểu mà con đom đóm cần phá để thoát khỏi hang, và có bao nhiêu cách chọn khác nhau đưa đến kết quả đó.

Input

- Dòng 1: Hai số nguyên N và H ($1 \leq N \leq 2 \cdot 10^5$ và $1 \leq H \leq 5 \cdot 10^5$) là chiều dài và chiều cao của hang.
- Mỗi dòng trong N dòng tiếp theo là một số nguyên dương - kích thước của chướng ngại vật. Tất cả các kích thước đều nhỏ hơn H .

Output

- Gồm 2 số nguyên cách nhau là số chuồng ngai vật ít nhất cần phá và số cách chọn khác nhau để có được kết quả đó.

Example

C11CAVE.INP	C11CAVE.OUT
6 7 1 5 3 3 5 1	2 3
14 5 1 3 4 2 2 4 3 4 3 3 3 2 3 3	7 2

Subtask

- Subtask 1 (40%): $N \leq 1000$, $H \leq 1000$
- Subtask 2 (60%): Không có giới hạn gì thêm.

Problem MSTICK

Input file: **MSTICK.INP**
Output file: **MSTICK.OUT**
Time limit: 1 giây
Memory limit: 512 megabytes

Có n đoạn gỗ, mỗi đoạn có một chiều dài l và trọng lượng w . Để xử lý chúng cần thời gian chuẩn bị:

(a) Thời gian chuẩn bị cho đoạn gỗ đầu tiên là 1 phút.

(b) Sau khi xử lý xong đoạn gỗ có chiều dài l và trọng lượng w , không mất thời gian xử lý nếu đoạn gỗ tiếp theo có độ dài l' và trọng lượng w' thỏa $l \leq l'$ và $w \leq w'$. Ngược lại mất 1 phút để chuẩn bị.

Tìm thời gian chuẩn bị ít nhất cho n đoạn gỗ. Ví dụ có 5 đoạn $(9, 4), (2, 5), (1, 2), (5, 3)$, và $(4, 1)$, thì thời gian ít nhất để chuẩn bị là 2 vì có thể xử lý theo thứ tự sau $(4, 1), (5, 3), (9, 4), (1, 2), (2, 5)$.

Input

- Dòng đầu là số lượng test T ($1 \leq T \leq 100$). Mỗi test gồm 2 dòng:
 - Dòng đầu là số n ($1 \leq n \leq 5000$).
 - Dòng thứ hai gồm $2n$ số nguyên dương $l_1, w_1, l_2, w_2, \dots, l_n, w_n$ ($1 \leq l_i, w_i \leq 10000$).

Output

- Ghi ra thời gian ít nhất trên từng dòng.

Example

MSTICK.INP	MSTICK.OUT
3	2
5	1
4 9 5 2 2 1 3 5 1 4	3
3	
2 2 1 1 2 2	
3	
1 3 2 2 3 1	

Subtask

- Subtask 1 (30%): $T = 1, n \leq 10$
- Subtask 2 (40%): $T = 1$
- Subtask 3 (30%): Không có giới hạn gì thêm.

Problem SUSHI

Input file: SUSHI.INP
Output file: SUSHI.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

Arkady mời Anna đi ăn tối tại một nhà hàng sushi. Nhà hàng này có một chút đặc biệt: nó cung cấp n mẫu sushi xếp liền nhau và một khách hàng phải chọn một phân đoạn liên tục của những mẫu sushi này để mua.

Các mẫu sushi có hai loại: mẫu sushi cá ngừ hoặc mẫu sushi con lươn. Cho n sushi nằm trên một đoạn thẳng, loại của mẫu sushi thứ i tính từ bên trái sang là t_i , trong đó $t_i=1$ có nghĩa là sushi cá ngừ và $t_i=2$ có nghĩa là sushi con lươn.

Arkady không thích sushi cá ngừ, còn Anna không thích sushi con lươn. Arkady muốn chọn một phân đoạn liên tục của sushi sao cho nó có số lượng sushi của mỗi loại bằng nhau và mỗi nửa của phân đoạn chỉ chứa sushi của một loại. Ví dụ, phân đoạn $[2,2,2,1,1,1]$ là hợp lệ, nhưng phân đoạn $[1,2,1,2,1,2]$ không hợp lệ, vì cả hai nửa đều chứa cả hai loại sushi.

Tìm độ dài của phân đoạn liên tục dài nhất mà Arkady có thể mua được.

Input

- Dòng đầu tiên chứa một số nguyên duy nhất n ($2 \leq n \leq 100000$) — số lượng mẫu sushi.
- Dòng thứ hai chứa n số nguyên t_1, t_2, \dots, t_n (với $t_i=1$, đại diện cho sushi cá ngừ hoặc $t_i=2$, đại diện cho sushi con lươn), biểu thị loại sushi từ trái qua phải.
- Đảm bảo rằng có ít nhất một mẫu sushi của mỗi loại. Lưu ý rằng điều này có nghĩa là có ít nhất một phân đoạn liên tục hợp lệ.

Output

- In ra một số nguyên duy nhất — độ dài tối đa của một phân đoạn liên tục hợp lệ.

Example

SUSHI.INP	SUSHI.OUT
7 2 2 2 1 1 2 2	4
6 1 2 1 2 1 2	2
9 2 2 1 1 1 2 2 2	6

Subtask

- Subtask 1 (40%): $n \leq 1000$
- Subtask 2 (60%): Không có giới hạn gì thêm.

Problem PRIMEMATRIX

Input file: PRIMEMATRIX.INP
Output file: PRIMEMATRIX.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

Bạn được cho một ma trận kích thước $n \times m$. Ma trận này bao gồm các số nguyên. Trong một lần thao tác, bạn có thể thực hiện một phép biến đổi duy nhất trên ma trận: chọn một phần tử bất kỳ trong ma trận và tăng giá trị của nó lên 1. Mỗi phần tử có thể được tăng lên một số lần tùy ý.

Bạn rất tò mò về các số nguyên tố. Nhắc lại rằng một số nguyên tố là một số nguyên dương có đúng hai ước số nguyên dương phân biệt: chính nó và số một. Ví dụ, các số 2, 3, 5 là số nguyên tố và các số 1, 4, 6 không phải là số nguyên tố.

Một ma trận được coi là ma trận số nguyên tố nếu ít nhất một trong hai điều kiện sau được thỏa mãn:

- Ma trận có một hàng chỉ chứa toàn các số nguyên tố;
- Ma trận có một cột chỉ chứa toàn các số nguyên tố.

Nhiệm vụ của bạn là đếm số lần thao tác tối thiểu cần thiết để có được một ma trận số nguyên tố từ ma trận bạn đã có.

Input

- Dòng đầu tiên chứa hai số nguyên n, m ($1 \leq n, m \leq 500$) — số hàng và số cột trong ma trận, tương ứng.
- Tiếp theo, mỗi trong n dòng sau chứa m số nguyên — ma trận ban đầu. Tất cả các phần tử của ma trận là số nguyên dương. Tất cả các số trong ma trận ban đầu không vượt quá 10^5 .
- Các số trong các dòng được phân tách bởi dấu cách.

Output

- In hấy in ra một số nguyên duy nhất — số lần thao tác tối thiểu cần thiết để có được một ma trận số nguyên tố từ ma trận ban đầu. Nếu ma trận được cho đã là một ma trận số nguyên tố, hãy in 0.

Example

PRIMEMATRIX.INP	PRIMEMATRIX.OUT
3 3 1 2 3 5 6 1 4 4 1	1
2 3 4 8 8 9 2 9	3
2 2 1 3 4 2	0

Problem COPY

Input file: COPY.INP
Output file: COPY.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

Ban đầu, bạn có một mảng a gồm một phần tử là 1 ($a = [1]$).

Trong một thao tác, bạn có thể thực hiện một trong hai điều sau:

- Tăng giá trị của một (duy nhất) phần tử trong a lên 1 (chọn một i từ 1 đến độ dài hiện tại của a và tăng a_i lên 1).
- Thêm một bản sao của một (duy nhất) phần tử trong a vào cuối mảng (chọn một i từ 1 đến độ dài hiện tại của a và thêm a_i vào cuối mảng).

Ví dụ, xem xét chuỗi năm thao tác:

- Ban đầu chuỗi $a = [1]$.
- Bạn lấy phần tử đầu tiên a_1 , thêm bản sao của nó vào cuối mảng và có $a = [1, 1]$.
- Bạn lấy phần tử đầu tiên a_1 , tăng nó lên 1 và có $a = [2, 1]$.
- Bạn lấy phần tử thứ hai a_2 , thêm bản sao của nó vào cuối mảng và có $a = [2, 1, 1]$.
- Bạn lấy phần tử đầu tiên a_1 , thêm bản sao của nó vào cuối mảng và có $a = [2, 1, 1, 2]$.
- Bạn lấy phần tử thứ tư a_4 , tăng nó lên 1 và có $a = [2, 1, 1, 3]$.

Nhiệm vụ của bạn là tìm số lần di chuyển tối thiểu cần thiết để có được mảng a mà có tổng ít nhất là n.

Bạn cần trả lời t test độc lập.

Input

- Dòng đầu tiên của đầu vào chứa một số nguyên t ($1 \leq t \leq 1000$) — số lượng bộ test. Mỗi bộ test có dạng như sau:
- Dòng duy nhất của mỗi bài kiểm tra chứa một số nguyên n ($1 \leq n \leq 10^9$) — giới hạn dưới của tổng các phần tử trong mảng.

Output

- Đối với mỗi test, in ra câu trả lời: số thao tác tối thiểu cần thiết để có được mảng a có tổng ít nhất là n.

Example

COPY.INP	COPY.OUT
5	0
1	3
5	11
42	72
1337	63244
1000000000	

Subtask

- Subtask 1 (40%): $n \leq 10^4$
- Subtask 2 (60%): Không có giới hạn gì thêm.

Problem PTRANG

Input file: PTRANG.INP
Output file: PTRANG.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

Văn bản là một dãy gồm N từ đánh số từ 1 đến N . Từ thứ i có độ dài là w_i ($i = 1, 2, \dots, N$). Phân trang là một cách xếp lần lượt các từ của văn bản vào dãy các dòng, mỗi dòng có độ dài L , sao cho tổng độ dài của các từ trên cùng một dòng không vượt quá L . Ta gọi hệ số phạt của mỗi dòng trong cách phân trang là hiệu số $(L - S)$, trong đó S là tổng độ dài của các từ xếp trên dòng đó. Hệ số phạt của cách phân trang là giá trị lớn nhất trong số các hệ số phạt của các dòng. Tìm cách phân trang với hệ số phạt nhỏ nhất.

Input

- Dòng 1 chứa 2 số nguyên dương N, L ($N \leq 6000, L \leq 1000$)
- Dòng thứ i trong số N dòng tiếp theo chứa số nguyên dương w_i ($w_i \leq L$), $i = 1, 2, \dots, N$

Output

- In ra hệ số phạt nhỏ nhất

Example

PTRANG.INP	PTRANG.OUT
4 5 3 2 2 4	2

Subtask

- Subtask 1 (40%): $N \leq 10, L \leq 10$
- Subtask 2 (60%): Không có giới hạn gì thêm.

Problem QBMAX

Input file: QBMAX.INP
Output file: QBMAX.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

Cho một bảng A kích thước $m \times n$ ($1 \leq m, n \leq 100$), trên đó ghi các số nguyên a_{ij} ($|a_{i,j}| \leq 100$). Một người xuất phát tại ô nào đó của cột 1, cần sang cột n (tại ô nào cũng được).

Quy tắc đi: Từ ô (i, j) chỉ được quyền sang một trong 3 ô $(i, j+1)$; $(i-1, j+1)$; $(i+1, j+1)$

Input

- Dòng đầu tiên ghi hai số m, n là số hàng và số cột của bảng.
- M dòng tiếp theo, dòng thứ i ghi đủ n số trên hàng i của bảng theo đúng thứ tự từ trái qua phải.

Output

- Gồm 1 dòng duy nhất ghi tổng lớn nhất tìm được.

Example

QBMAX.INP	QBMAX.OUT
5 7 9 -2 6 2 1 3 4 0 -1 6 7 1 3 3 8 -2 8 2 5 3 2 1 -1 6 2 1 6 1 7 -2 6 2 1 3 7	41

Subtask

- Subtask 1 (40%): $n, m \leq 5$
- Subtask 2 (60%): Không có giới hạn gì thêm.

Problem QBSTR

Input file: QBSTR.INP
Output file: QBSTR.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

Xâu ký tự X được gọi là xâu con của xâu ký tự Y nếu ta có thể xoá đi một số ký tự trong xâu Y để được xâu X .

Cho biết hai xâu ký tự A và B , hãy tìm xâu ký tự C có độ dài lớn nhất và là con của cả A và B .

Input

- Dòng 1: chứa xâu A ($|A| \leq 5000$)
- Dòng 2: chứa xâu B ($|B| \leq 5000$)

Output

- Chỉ gồm một dòng ghi độ dài xâu C tìm được

Example

QBSTR.INP	QBSTR.OUT
abc1def2ghi3 abcdefghi123	10

Subtask

- Subtask 1 (40%): $|A|, |B| \leq 10$
- Subtask 2 (60%): Không có giới hạn gì thêm.

Problem QBSEQ

Input file: QBSEQ.INP
Output file: QBSEQ.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

Cho một dãy gồm n ($n \leq 1000$) số nguyên dương A_1, A_2, \dots, A_n và một số nguyên dương k ($k \leq 50$). Hãy tìm dãy con gồm nhiều phần tử nhất của dãy đã cho sao cho tổng các phần tử của dãy con này chia hết cho k .

Input

- Dòng đầu tiên chứa hai số nguyên n và k , được ghi cách nhau bởi ít nhất một dấu trống.
- Các dòng tiếp theo chứa các số A_1, A_2, \dots, A_n ($1 \leq A_i \leq 1000$), được ghi theo đúng thứ tự cách nhau ít nhất một dấu cách.

Output

- Gồm 1 dòng duy nhất ghi số lượng phần tử của dãy con dài nhất thỏa mãn.

Example

QBSEQ.INP	QBSEQ.OUT
10 3 2 3 5 7 9 6 12 7 11 15	9

Subtask

- Subtask 1 (40%): $n \leq 20$
- Subtask 2 (60%): Không có giới hạn gì thêm.

Problem NKTICK

Input file: NKTICK.INP
Output file: NKTICK.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

Có N người sắp hàng mua vé dự buổi hòa nhạc. Ta đánh số họ từ 1 đến N theo thứ tự đứng trong hàng. Mỗi người cần mua một vé, song người bán vé được phép bán cho mỗi người tối đa hai vé. Vì thế, một số người có thể rời hàng và nhờ người đứng trước mình mua hộ vé. Biết t_i là thời gian cần thiết để người i mua xong vé cho mình. Nếu người $i + 1$ rời khỏi hàng và nhờ người i mua hộ vé thì thời gian để người thứ i mua được vé cho cả hai người là r_i .

Yêu cầu: Xác định xem những người nào cần rời khỏi hàng và nhờ người đứng trước mua hộ vé để tổng thời gian phục vụ bán vé là nhỏ nhất.

Input

- Dòng đầu tiên chứa số nguyên N ($1 \leq N \leq 60000$), thể hiện số người trong hàng.
- Dòng thứ hai ghi N số nguyên dương t_1, t_2, \dots, t_N ($1 \leq t_i \leq 30000$), thể hiện thời gian cần thiết để người i mua vé cho chính họ.
- Dòng thứ ba ghi $N - 1$ số nguyên dương r_1, r_2, \dots, r_{N-1} ($1 \leq r_i \leq 30000$), thể hiện thời gian để mua vé cho người khác (nếu có).

Output

- In ra tổng thời gian phục vụ nhỏ nhất.

Example

NKTICK.INP	NKTICK.OUT
5 2 5 7 8 4 4 9 10 10	18
4 5 7 8 4 50 50 50	24

Subtask

- Subtask 1 (40%): $N \leq 20$
- Subtask 2 (60%): Không có giới hạn gì thêm.

Problem QBSELECT

Input file: QBSELECT.INP
Output file: QBSELECT.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

Cho một bảng hình chữ nhật kích thước $4 \times n$ ô vuông. Các dòng được đánh số từ 1 đến 4, từ trên xuống dưới, các cột được đánh số từ 1 đến n từ trái qua phải.

Ô nằm trên giao của dòng i và cột j được gọi là ô (i, j) . Trên mỗi ô (i, j) có ghi một số nguyên a_{ij} , $i = 1, 2, 3, 4$; $j = 1, 2, \dots, n$. Một cách chọn ô là việc xác định một tập con khác rỗng S của tập tất cả các ô của bảng sao cho không có hai ô nào trong S có chung cạnh. Các ô trong tập S được gọi là ô được chọn, tổng các số trong các ô được chọn được gọi là trọng lượng của cách chọn. Tìm cách chọn sao cho trọng lượng là lớn nhất.

	1	2	3
1	-1	9	3
2	-4	5	-6
3	7	8	9
4	9	7	2

Một trong các cách chọn cần tìm là tập các ô $S = \{(3, 1), (1, 2), (4, 2), (3, 3)\}$ với trọng lượng 32.

Input

- Đòng đầu tiên chứa số nguyên dương n , là số cột của bảng.
- Cột thứ j trong số n cột tiếp theo chứa 4 số nguyên $a_{1j}, a_{2j}, a_{3j}, a_{4j}$, hai số liên tiếp cách nhau ít nhất một dấu cách, là 4 số trên cột j của bảng.

Output

- Gồm 1 dòng duy nhất là trọng lượng của cách chọn tìm được.

Constraints

- Trong tất cả các test: $n \leq 10,000$, $|a_{ij}| \leq 30,000$.
- Có 50% số lượng test với $n \leq 5$.

Example

QBSELECT.INP	QBSELECT.OUT
3 -1 9 3 -4 5 -6 7 8 9 9 7 2	32

Problem QBGAME

Input file: QBGAME.INP
Output file: QBGAME.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

Ngày nay các nhà khoa học đã nghĩ ra 1 trò chơi trên ma trận rất thú vị. Thông qua đó có thể đo IQ một cách khá hiệu quả. Trò chơi được mô tả như sau:

Bạn có 1 ma trận A kích thước $8 \times N$ trên đó gồm các số nguyên là điểm của các ô đó. Người ta sẽ yêu cầu bạn chọn 1 tập khác rỗng các ô trên ma trận này sau đó tính tổng điểm trên những ô này. Trong những ô được chọn không có hai ô nào kề cạnh. IQ của người chơi sẽ tỉ lệ thuận với số điểm nhận được. Sherry tham gia trò chơi và đạt kết quả khá tốt. Và bây giờ Sherry muốn biết tổng điểm lớn nhất nhận được trong trò chơi này là bao nhiêu. Bạn hãy giúp Sherry nhé !!!

Input

- Dòng đầu tiên chứa số nguyên N ($1 \leq N \leq 10,000$), là số cột của ma trận.
- 8 dòng tiếp theo, mỗi dòng chứa N số nguyên. Số nguyên ở hàng i , cột j là A_{ij} ($-10^8 \leq A_{ij} \leq 10^8$).

Output

- Gồm 1 dòng duy nhất là số điểm lớn nhất tìm được

Example

QBGAME.INP	QBGAME.OUT
2 -22 2 -33 45 56 -60 -8 -38 79 66 -10 -23 99 46 1 -55	279

Explanation

Chọn các ô (3,1) (5,1) (7,1) (2,2)

Subtask

- Subtask 1 (40%): $N \leq 10$
- Subtask 2 (60%): Không có giới hạn gì thêm.

Problem LEM3

Input file: LEM3.INP
Output file: LEM3.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

Trong kì nghỉ hè năm nay, Sherry được bố thưởng cho 1 tour du lịch quanh N đất nước tươi đẹp với nhiều thắng cảnh nổi tiếng. Sherry sẽ di chuyển bằng máy bay.

Giá vé máy bay từ đất nước i đến đất nước j là C_{ij} (chú ý rằng C_{ij} có thể khác C_{ji}). Tuy được bố thưởng nhiều tiền để đi du lịch, Sherry vẫn muốn tìm cho mình 1 hành trình với chi phí thấp nhất có thể để dành tiền mua quà về tặng mọi người.

Bạn hãy giúp Sherry tìm 1 hành trình đi qua tất cả các nước, mỗi nước đúng 1 lần sao cho chi phí là bé nhất.

Input

- Dòng đầu tiên chứa số nguyên N ($5 < N < 16$), thể hiện số đất nước.
- Dòng thứ i trong N dòng tiếp theo chứa N số nguyên, số thứ j là C_{ij} ($0 < C_{ij} < 10,001$), thể hiện giá vé máy bay từ đất nước i đến đất nước j .

Output

- Gồm 1 dòng duy nhất ghi chi phí bé nhất tìm được

Example

LEM3.INP	LEM3.OUT
6 0 1 2 1 3 4 5 0 3 2 3 4 4 1 0 2 1 2 4 2 5 0 4 3 2 5 3 5 0 2 5 4 3 3 1 0	8

Subtask

- Subtask 1 (40%): $N \leq 10$
- Subtask 2 (60%): Không có giới hạn gì thêm.

Problem MIXUP2

Input file: MIXUP2.INP
Output file: MIXUP2.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

Mỗi trong N cô bò ($4 \leq N \leq 16$) của bác John có một số seri phân biệt S_i ($1 \leq S_i \leq 25,000$). Các cô bò tự hào đến nỗi mỗi cô đều đeo một chiếc vòng vàng có khắc số seri của mình trên cổ theo kiểu các băng dảng giang hồ.

Các cô bò giang hồ này thích nỗi loạn nên đứng xếp hàng chờ vết sưa theo một thứ tự gọi được gọi là 'hỗn loạn'.

Một thứ tự bò là 'hỗn loạn' nếu trong dãy số seri tạo bởi hàng bò, hai số liên tiếp khác biệt nhau nhiều hơn K ($1 \leq K \leq 3,400$). Ví dụ, nếu $N = 6$ và $K = 1$ thì $1, 3, 5, 2, 6, 4$ là một thứ tự 'hỗn loạn' nhưng $1, 3, 6, 5, 2, 4$ thì không (vì hai số liên tiếp 5 và 6 chỉ chênh lệch 1).

Hỏi có bao nhiêu cách khác nhau để N cô bò sắp thành thứ tự 'hỗn loạn'?

Input

- Dòng đầu tiên chứa hai số nguyên N và K , cách nhau bởi khoảng trắng.
- N dòng tiếp theo, mỗi dòng chứa một số nguyên duy nhất là số seri của cô bò thứ i : S_i .

Output

- Một số nguyên duy nhất là số cách để N cô bò sắp thành thứ tự 'hỗn loạn'. Kết quả đảm bảo nằm trong phạm vi kiểu số nguyên 64-bit.

Example

MIXUP2.INP	MIXUP2.OUT
4 1 3 4 2 1	2

Subtask

- Subtask 1 (40%): $N \leq 10$
- Subtask 2 (60%): Không có giới hạn gì thêm.

Problem QBRECT

Input file: QBRECT.INP
Output file: QBRECT.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

Cho một bảng kích thước $M \times N$, được chia thành lưới ô vuông đơn vị M dòng N cột ($1 \leq M, N \leq 1,000$).

Trên các ô của bảng ghi số 0 hoặc 1. Các dòng của bảng được đánh số từ 1 đến M theo thứ tự từ trên xuống dưới và các cột của bảng được đánh số từ 1 đến N theo thứ tự từ trái qua phải.

Yêu cầu:

Hãy tìm một hình chữ nhật gồm các ô của bảng thỏa mãn các điều kiện sau:

1. Hình chữ nhật đó chỉ gồm các số 1.
2. Cạnh hình chữ nhật song song với cạnh bảng.
3. Diện tích hình chữ nhật là lớn nhất có thể.

Input

- Dòng đầu tiên ghi hai số nguyên M và N , thể hiện kích thước của bảng ($1 \leq M, N \leq 1,000$).
- M dòng tiếp theo, dòng thứ i ghi N số nguyên, số thứ j là số ghi trên ô (i, j) của bảng.

Output

- Gồm 1 dòng duy nhất ghi diện tích của hình chữ nhật tìm được

Example

QBRECT.INP	QBRECT.OUT
11 13 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1	49

Subtask

- Subtask 1 (40%): $N, M \leq 100$
- Subtask 2 (60%): Không có giới hạn gì thêm.

Problem DTGAME

Input file: DTGAME.INP
Output file: DTGAME.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

Tiền bạc vẫn luôn là thứ có giá trị đối với mỗi con người, kể cả với pirate. Vì vậy, khi hòn đảo xinh đẹp của tên cướp biển khét tiếng này sắp bị... giải tỏa, hắn đã tranh thủ vơ vét cho đến đồng tiền vàng cuối cùng. Trên hòn đảo có N mỏ vàng nằm cạnh nhau trên một đường thẳng, đánh số từ 1 đến N . Mỏ vàng i có giá trị là p_i . Đội thi công sẽ có nhiệm vụ san lấp hết N mỏ vàng này. Nhưng tại mỗi thời điểm, pirate cố gắng ngăn cản việc san lấp bằng cách xây một bức tường ngăn cách hai mỏ vàng liên tiếp nào đó, vậy là cụm mỏ còn lại được chia làm 2 phần. Đội thi công sẽ chọn một trong hai phần đó để san lấp hết, thế là pirate giữ lại được phần còn lại và hắn sẽ nhận được số đồng tiền vàng đúng bằng tổng giá trị của những mỏ còn lại đó. Công việc cứ tiếp tục cho đến khi chỉ còn lại một mỏ vàng duy nhất. Đội thi công biết điều đó, cho nên họ đã có chiến thuật san lấp để cực tiểu hóa số tiền của pirate. Với lòng tham không đáy, pirate quyết tâm lấy càng nhiều vàng càng tốt. Hãy giúp con người khốn khổ vì tiền này!

Input

- Dòng đầu tiên chứa một số nguyên dương N ($1 \leq N \leq 2,000$) - số lượng mỏ vàng trên hòn đảo.
- N dòng tiếp theo, mỗi dòng chứa một số nguyên p_i ($1 \leq p_i \leq 1,000$) - giá trị của từng mỏ vàng theo thứ tự từ 1 đến N .

Output

- Dữ liệu ra gồm 1 dòng duy nhất ghi ra số đồng tiền vàng lớn nhất có thể vơ vét được.

Example

DTGAME.INP	DTGAME.OUT
5	
8	
6	
2	
4	
2	

Explanation

Đầu tiên, pirate chia mỏ vàng thành hai phần: [1,2] và [3,4,5]. Xe ủi sẽ san lấp phần [1,2], và pirate nhận được tổng giá trị của các mỏ vàng trong phần này: $p_3 + p_4 + p_5 = 8$ đồng tiền vàng.

Tiếp theo, pirate chia phần [3,4,5] thành [3] và [4,5]. Xe ủi san lấp phần [4,5], và pirate thu thêm giá trị của mỏ vàng $p_3 = 2$ đồng tiền vàng.

Công việc kết thúc vì chỉ còn lại 1 mỏ vàng duy nhất.

Vậy tổng cộng số tiền vàng mà pirate thu được là $8 + 2 = 10$ đồng tiền vàng. Không có cách nào giúp hắn thu thêm tiền.

Subtask

- Subtask 1 (40%): $N \leq 100$
- Subtask 2 (60%): Không có giới hạn gì thêm.

Problem QBTICKET

Input file: QBTICKET.INP
Output file: QBTICKET.OUT
Time limit: 1 giây
Memory limit: 512 megabytes

Tuyến đường sắt từ thành phố A đến thành phố B đi qua một số nhà ga. Tuyến đường có thể biểu diễn bởi một đoạn thẳng, các nhà ga là các điểm trên đó. Tuyến đường bắt đầu từ A và kết thúc ở B, vì thế các nhà ga sẽ được đánh số bắt đầu từ A (có số hiệu là 1) và B là nhà ga cuối cùng.

Giá vé đi lại giữa hai nhà ga chỉ phụ thuộc vào khoảng cách giữa chúng. Cách tính giá vé như sau:

- Khoảng cách $0 < X \leq L_1 \rightarrow$ Giá vé C1
- Khoảng cách $0 < X \leq L_2 \rightarrow$ Giá vé C2
- Khoảng cách $0 < X \leq L_3 \rightarrow$ Giá vé C3

Nghĩa là với các giá vé C1, C2, C3 tương ứng bạn sẽ đi quảng đường tối đa là L1, L2, L3.

Vé để đi thẳng từ nhà ga này đến nhà ga khác chỉ có thể đặt mua nếu khoảng cách giữa chúng không vượt quá L3. Vì thế nhiều khi để đi từ nhà ga này đến nhà ga khác ta phải đặt mua một số vé. Hơn thế nữa, nhân viên đường sắt yêu cầu hành khách chỉ được giữ đúng một vé khi đi trên tàu và vé đó sẽ bị huỷ khi hành khách xuống tàu.

Yêu cầu: Tìm cách đặt mua vé để đi lại giữa hai nhà ga cho trước với chi phí mua vé là nhỏ nhất.

Input

- Dòng đầu tiên: Ghi các số nguyên $L_1, L_2, L_3, C_1, C_2, C_3$ ($1 \leq L_1 \leq L_2 \leq L_3 \leq 10^9; 1 \leq C_1 \leq C_2 \leq C_3 \leq 10^9$) theo đúng thứ tự liệt kê ở trên.
- Dòng thứ hai: Chứa số lượng nhà ga N ($2 \leq N \leq 100000$).
- Dòng thứ ba: Ghi hai số nguyên s, f là các chỉ số của hai nhà ga cần tìm cách đặt mua vé với chi phí nhỏ nhất để đi lại giữa chúng.
- Dòng thứ i trong số $N - 1$ dòng tiếp theo: Ghi số nguyên là khoảng cách từ nhà ga A (ga 1) đến nhà ga thứ $i + 1$.

Output

- Gồm 1 dòng duy nhất ghi chi phí nhỏ nhất tìm được

Example

QBTICKET.INP	QBTICKET.OUT
3 6 8 20 30 40 7 2 6 3 7 8 13 15 23	70