

Problem A. DP LIS bản khó

Time Limit 1000 ms

Mem Limit 524288 kB

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?

Input

- 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 ($1 \leq A_i \leq 10000$).

Output

Ghi ra độ dài của dãy con tăng đơn điệu dài nhất.

Input	Output
6 1 2 5 4 6 2	4

Note

Dãy con dài nhất là dãy $A_1 = 1 < A_2 = 2 < A_4 = 4 < A_5 = 6$, độ dài dãy này là 4.

Download test và solution tại [đây](#).

Problem B. DP nhân ma trận

Time Limit 1000 ms

Mem Limit 262144 kB

Bờm chơi trò chơi điện tử Lucky Luke đến màn phải điều khiển Lucky leo lên một cầu thang gồm N bậc.

Các bậc thang được đánh số từ 1 đến N từ dưới lên trên. Lucky có thể đi lên một bậc thang, hoặc nhảy một bước lên hai bậc thang. Tuy nhiên một số bậc thang đã bị thủng do cũ kỹ và Lucky không thể bước chân lên được. Biết ban đầu, Lucky đứng ở bậc thang số 1 (bậc thang số 1 không bao giờ bị thủng).

Chơi đến đây, Bờm chợt nảy ra câu hỏi: có bao nhiêu cách để Lucky leo hết được cầu thang? (nghĩa là leo đến bậc thang thứ N). Bờm muốn nhờ bạn trả lời câu hỏi này.

Input

Dòng đầu tiên: gồm 2 số nguyên N và K , là số bậc của cầu thang và số bậc thang bị hỏng ($0 \leq K < N \leq 10^5$).

Dòng thứ hai: gồm K số nguyên cho biết chỉ số của các bậc thang bị hỏng theo thứ tự tăng dần.

Output

In ra phần dư của số cách Lucky leo hết cầu thang khi chia cho 14062008.

Input	Output
4 2 2 3	0
Input	Output
90000 1 49000	4108266

Problem C. DP chia nhóm 2

Time Limit 1000 ms

Mem Limit 262144 kB

Có N người sắp hàng mua vé dự buổi hoà 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 N ($1 \leq N \leq 60000$).
- Dòng thứ hai ghi N số nguyên dương t_1, t_2, \dots, t_N . ($1 \leq t_i \leq 30000$)
- Dòng thứ ba ghi $N - 1$ số nguyên dương r_1, r_2, \dots, r_{N-1} . ($1 \leq r_i \leq 30000$)

Output

In ra tổng thời gian phục vụ nhỏ nhất.

Input	Output
5 2 5 7 8 4 4 9 10 10	18

Dãy con tăng dài nhất

Cho dãy a_1, a_2, \dots, a_n gồm n phần tử (với $1 \leq n \leq 10^9$) được mã hóa thành mảng b_1, b_2, \dots, b_n gồm m phần tử như sau: Dãy b sẽ gồm các phần tử có dạng $x.y$ với ý nghĩa: Phần tử hiện đang có giá trị là x , y phần tử tiếp theo lập thành một dãy cấp số cộng với công sai là 1. Xét $a = \{1, 2, 3, 2, 9, 4, 5, 6\}$. Ta có thể mã hóa mảng a thành mảng $b = \{1.3, 2.1, 9.1, 4.3\}$ như sau:

- Tại vị trí $i = 1$, ta thấy từ $i = 1$ đến $j = 3$ lập thành một dãy cấp số cộng có công sai 1 với $u_0 = 1$, gồm 3 phần tử $\rightarrow b = \{1.3\}$.
- Tại vị trí $i = 4$, ta thấy từ $i = 4$ đến $j = 4$ lập thành một dãy cấp số cộng có công sai 1 với $u_0 = 2$, gồm 1 phần tử $\rightarrow b = \{1.3, 2.1\}$.
- Tại vị trí $i = 5$, ta thấy từ $i = 5$ đến $j = 5$ lập thành một dãy cấp số cộng có công sai 1 với $u_0 = 9$, gồm 1 phần tử $\rightarrow b = \{1.3, 2.1, 9.1\}$
- Tại vị trí $i = 6$, ta thấy từ $i = 6$ đến $j = 8$ lập thành một dãy cấp số cộng có công sai 1 với $u_0 = 4$, gồm 3 phần tử $\rightarrow b = \{1.3, 2.1, 9.1, 4.3\}$.

Sau khi được mã hóa, mảng b sẽ gồm m phần tử (với $1 \leq m \leq 10^5$). Từ dãy b cho trước, tìm độ dài của dãy con tăng dần dài nhất của dãy a ban đầu khi chưa được mã hóa sang dãy b .

Dữ liệu

Một số nguyên m , là số phần tử của mảng b_1, b_2, \dots, b_m ($1 \leq m \leq 10^5$).

m dòng tiếp theo, mỗi dòng gồm hai số nguyên dương x_i và y_i ($1 \leq x_i \leq 10^6$, $1 \leq y_i \leq 10^9$, $x_i + y_i \leq 10^9$).

Dữ liệu đầu vào đảm bảo số lượng phần tử của dãy a ban đầu có không quá 10^9 phần tử.

Kết quả

Một số nguyên duy nhất là độ dài của dãy con tăng dần dài nhất của dãy a .

Ví dụ

Sample Input	Sample Output
3 1 3 4 2 2 7	8
4 1 3 2 1 9 1 4 3	6

Chấm điểm

Free Contest Cup 2023

- Subtask 1 (5% số test): dãy a ban đầu có dạng $a_i = a_{i-1} + 1$ (với mọi i thoả $1 < i \leq n$).
 - Subtask 2 (10% số test): $1 \leq n \leq 10^3$.
 - Subtask 3 (15% số test): $1 \leq n \leq 10^5$.
 - Subtask 4 (25% số test): $1 \leq m \leq 10^3$.
 - Subtask 5 (45% số test): Không có ràng buộc gì thêm.
-

Problem E. DP at school

Time Limit 1000 ms

Mem Limit 524288 kB

Trường THPT chuyên Lê Quý Đôn, TP Đà Nẵng có một hệ thống xe buýt riêng. Hằng ngày xe buýt sẽ chạy dọc theo bờ sông Hàn thơ mộng để đón các học sinh đến trường. Trên hành trình, xe buýt sẽ tiếp nhận các học sinh đứng chờ ở các điểm hẹn nếu như xe còn chỗ trống. Xe buýt có thể đỗ lại để chờ những học sinh chưa kịp đến điểm hẹn. Tuy nhiên vì học sinh trường Lê Quý Đôn là những học sinh rất coi trọng việc giờ giấc nên họ có thể để xe buýt chờ chứ không bao giờ chịu chờ xe buýt. Điều đó có nghĩa là nếu họ đến điểm hẹn mà xe chưa đến thì họ sẽ tự đi bộ đến trường.

Cho biết thời điểm mà mỗi học sinh đến điểm hẹn của mình và thời điểm qua mỗi điểm hẹn của xe buýt. Giả thiết rằng xe buýt đến điểm hẹn đầu tiên tại thời điểm 0 và thời gian xếp khách lên xe được bằng 0.

Xe buýt cần phải chở một số lượng nhiều nhất các học sinh có thể được đến trường. Hãy xác định khoảng thời gian ngắn nhất để xe buýt thực hiện công việc.

Input

- Dòng đầu tiên chứa 2 số nguyên dương n, m theo thứ tự là số điểm hẹn và số chỗ ngồi của xe buýt.
- Dòng thứ i trong số n dòng tiếp theo chứa số nguyên t_i là thời gian cần thiết để xe buýt di chuyển từ điểm hẹn thứ i đến điểm hẹn thứ $i + 1$ (điểm hẹn thứ $n + 1$ sẽ là trường Lê Quý Đôn) và số nguyên k là số lượng học sinh đến điểm hẹn i , tiếp theo k số nguyên là các thời điểm đến điểm hẹn của k học sinh.
- $1 \leq n \leq 200000, 1 \leq m \leq 20000$
- Tổng số học sinh không vượt quá 200000.
- Kết quả không vượt quá $2^{31} - 1$.

Output

- Gồm một dòng duy nhất, là thời gian ngắn nhất tìm được.

Input	Output
3 5 1 2 0 1 1 1 2 1 4 0 2 3 4	5

Problem F. DP online

Time Limit 1000 ms

Mem Limit 524288 kB

Các học sinh khi đến thực tập trong phòng máy tính thường hay chơi trò chơi điện tử trên mạng. Để ngăn ngừa, người trực phòng máy đã ngắt tất cả các máy tính ra khỏi mạng và xếp chúng thành một dãy trên một cái bàn dài và gắn chặt máy xuống mặt bàn rồi đánh số thứ tự các máy từ 1 đến N theo chiều từ trái sang phải. Các học sinh tinh nghịch không chịu thua, họ đã quyết định tìm cách nối các máy trên bàn bởi các đoạn dây nối sao cho mỗi máy được nối với ít nhất một máy khác. Để tiến hành công việc này, họ đã đo khoảng cách giữa hai máy liên tiếp. Bạn hãy giúp các học sinh này tìm cách nối mạng thoả mãn yêu cầu đặt ra sao cho tổng độ dài cáp nối phải sử dụng là ít nhất.

Input

- Dòng đầu tiên chứa số lượng máy N ($1 \leq N \leq 25000$).
- Dòng thứ i trong số $N - 1$ dòng tiếp theo chứa các khoảng cách từ máy i đến máy $i + 1$ ($i = 1, 2, \dots, N - 1$). Giả thiết rằng khoảng cách từ máy 1 đến máy N không vượt quá 10^6 .

Output

Ghi ra độ dài của cáp nối cần sử dụng.

Input	Output
6 2 2 3 2 2	7

Problem G. DP Hall

Time Limit 1000 ms

Mem Limit 524288 kB

Nhà trường có một phòng hội trường. Có những yêu cầu muốn sử dụng phòng hội trường này, mỗi yêu cầu cho biết thời điểm bắt đầu và thời điểm kết thúc. Nhà trường có thể chấp nhận hoặc từ chối đối với một yêu cầu.

Yêu cầu: hãy giúp nhà trường chọn các yêu cầu sử dụng hội trường sao cho tổng thời gian hội trường được sử dụng là lớn nhất.

Input

Dòng đầu tiên chứa một số nguyên dương n ($n \leq 10000$), số yêu cầu.

Mỗi dòng trong số n dòng tiếp theo chứa 2 số nguyên dương p và k ($0 \leq p < k \leq 30000$), mô tả một yêu cầu bắt đầu tại thời điểm p và kết thúc tại thời điểm k .

Output

Gồm một dòng duy nhất là tổng thời gian lớn nhất mà hội trường được sử dụng

Input	Output
12 1 2 3 5 0 4 6 8 7 13 4 6 9 10 9 12 11 14 15 19 14 16 18 20	16

Problem H. DP sequence

Time Limit 380 ms

Mem Limit 262144 kB

Một vụ tai nạn giao thông vừa xảy ra ngoài phố. Mọi người tò mò chen lấn vòng trong vòng ngoài để xem. Rất đông người xem vụ tai nạn như: **khanhcank7**, **nhungngoisao**, **chicuong123**, **harryporter7**, **mr_ntt**, **manhboyak6**. Một anh đến chậm không tài nào vào xem được. Tức quá, anh ta liền hét toáng lên:

- Tôi là bố kẻ bị nạn đây!

Mọi người kinh ngạc quay lại nhìn và vội vã giãn ra cho anh ta vào. "Kẻ bị nạn" là một... chú chó vừa bị xe cán chết.

Ngại ngùng, anh ta chạy về nhà, than khóc. Và ông tiên **hellosirius** hiện ra ban cho anh ta 1 điều ước. Anh ta ước rằng mọi người trên thế giới sẽ quên hết mọi chuyện trong hôm nay. Tuy nhiên, theo định luật bảo toàn năng lượng cũng như định luật bảo toàn tính mạng ông tiên buộc chú phải giải bài toán sau mới giúp chú thực hiện điều ước. Nhanh chóng chú giải được bài toán ông đưa ra tuy nhiên, lần này ông buộc chú phải chiến đấu với hiệp sĩ đẹp trai **n_cqt** và **blackstar**. Để chiến thắng hiệp sĩ cách duy nhất là giải được bài toán hiệp sĩ đưa ra (vì hiệp sĩ không những đẹp trai, học giỏi mà còn khỏe mạnh vô đối nữa). Đề bài của hiệp sĩ như sau:

Cho dãy số nguyên A gồm N phần tử A_1, A_2, \dots, A_n . Tìm cặp chỉ số i, j thỏa mãn:

$DSEQ = |(A_1 + A_2 + \dots + A_i) - (A_j + A_{j+1} + \dots + A_n)|$ đạt giá trị lớn nhất (với $1 \leq i < j \leq N$).

Hãy giúp anh bạn khốn khổ của chúng ta hoàn thành điều ước!

Input

Dòng đầu là số nguyên dương N ; ($2 \leq N \leq 10^6$).

Dòng tiếp theo chứa N số nguyên A_1, A_2, \dots, A_n ; ($|A_i| < 10^9$), các số cách nhau một dấu cách.

Output

Gồm một dòng chứa 2 số nguyên là DSEQ lớn nhất tìm được (do hiệp sĩ **n_cqt** yêu cầu) và số cặp chỉ số thỏa mãn (hai số cách nhau một dấu cách) (do hiệp sĩ **blackstar** yêu cầu)

Giới hạn

$\frac{1}{2}$ số test có $N \leq 5000$

Input	Output
5 1 -2 3 -4 -7	13 1

Problem I. DP build

Time Limit 380 ms

Mem Limit 524288 kB

Nông dân John muốn xây một cái hàng rào có 4 mặt vây lấy đàn bò. Ông ta có một thanh gỗ có độ dài là 1 số nguyên N ($4 \leq N \leq 2500$), ông ta muốn cắt thanh gỗ này tại 3 điểm để chia thành 4 miếng nhỏ hơn, mỗi miếng có độ dài là 1 số nguyên.

4 miếng này dài ngắn thế nào cũng được miễn là có thể giúp nông dân John đóng được 1 cái hàng rào hình tứ giác là được. Hỏi có bao nhiêu cách khác nhau cắt thanh gỗ ban đầu để tạo thành được hàng rào?

- Hai cách cắt gọi là khác nhau nếu một cách có 1 nhát cắt tại 1 điểm mà cách kia không có.
- Đảm bảo rằng hàng rào này xây dựng có diện tích lớn hơn 0.
- Chú ý đáp án luôn nằm trong phạm vi 1 số nguyên 32 bit có dấu.

Input

- Dòng 1: 1 số nguyên duy nhất: N

Output

- Một số nguyên duy nhất là số cách mà nông dân John có thể cắt thanh gỗ thành 4 miếng nhỏ hơn mà có thể tạo được 1 tứ giác.

Input	Output
6	6

Note

Nông dân John có thể cắt thanh gỗ theo 10 cách: (1, 1, 1, 3); (1, 1, 2, 2); (1, 1, 3, 1); (1, 2, 1, 2); (1, 2, 2, 1); (1, 3, 1, 1); (2, 1, 1, 2); (2, 1, 2, 1); (2, 2, 1, 1); or (3, 1, 1, 1). Trong đó 4 cách -- (1, 1, 1, 3), (1, 1, 3, 1), (1, 3, 1, 1), và (3, 1, 1, 1) -- không thể sử dụng để tạo thành 1 tứ giác.

Problem J. DP on DAG

Time Limit 1000 ms

Mem Limit 524288 kB

Cho một bảng A kích thước $m \times n$ ($1 \leq m, n \leq 100$), trên đó ghi các số nguyên a_{ij} ($|a_{ij}| \leq 100$). 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 1: 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

Input	Output
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

Problem K. DP on graph 1

Time Limit 380 ms

Mem Limit 524288 kB

Công viên Disneyland khai trương N điểm vui chơi mới. Một hôm Bờm và Cuội đến chơi Công viên. Qua việc hỏi thăm người hướng dẫn Bờm và Cuội đã biết thời gian cụ thể để đi từ điểm vui chơi i đến địa điểm vui chơi j . Bờm và Cuội muốn đi thăm hết tất cả các địa điểm theo nguyên tắc sau:

- Mỗi người đi từ điểm số 1, qua 1 số điểm và quay trở về điểm số 1.
- Các điểm đến trên đường đi của Bờm và Cuội là 1 dãy số tăng dần(trừ điểm số 1 khi quay về).
- Mỗi điểm vui chơi phải thuộc ít nhất 1 trong 2 đường đi của Bờm và Cuội.
- Thời gian đi là nhỏ nhất (tính cả thời gian quay về 1).

Input

- Dòng đầu là số N ($N < 201$)
- Tiếp theo là ma trận N dòng N cột: 1 số nguyên không âm (< 1001) trên hàng i cột j cho ta biết thời gian đi từ địa điểm i đến địa điểm j . ($A_{i,j} = A_{j,i}$, $A_{i,i} = 0$ với mọi i, j).

Output

Gồm 1 số duy nhất là thời gian nhỏ nhất để Bờm và Cuội thăm quan Công Viên theo nguyên tắc trên.

Input	Output
4 0 1 4 3 1 0 2 4 4 2 0 4 3 4 4 0	10

Note

(Giải thích : cách đi tốt nhất là Bờm đi $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1$ và Cuội đứng yên).

Problem L. DP on graph 2

Time Limit 380 ms

Mem Limit 524288 kB

Công viên Disneyland khai trương N điểm vui chơi mới. Một hôm Bờm và Cuội đến chơi Công viên. Qua việc hỏi thăm người hướng dẫn Bờm và Cuội đã biết thời gian cụ thể để đi từ điểm vui chơi i đến địa điểm vui chơi j . Bờm và Cuội muốn đi thăm hết tất cả các địa điểm theo nguyên tắc sau:

- Mỗi người đi từ điểm số 1, qua 1 số điểm và quay trở về điểm số 1.
- Các điểm đến trên đường đi của Bờm và Cuội là 1 dãy số tăng dần(trừ điểm số 1 khi quay về).
- Mỗi điểm vui chơi chỉ thuộc 1 trong 2 đường đi của Bờm hoặc Cuội (tất nhiên là trừ điểm số 1).
- Thời gian đi là nhỏ nhất (tính cả thời gian quay về 1).

Input

- Dòng đầu là số N ($N < 201$)
- Tiếp theo là ma trận N dòng N cột: 1 số nguyên không âm (< 1001) trên hàng i cột j cho ta biết thời gian đi từ địa điểm i đến địa điểm j . ($A_{i,j} = A_{j,i}$, $A_{i,i} = 0$ với mọi i, j).

Output

Gồm 1 số duy nhất là thời gian nhỏ nhất để Bờm và Cuội thăm quan Công Viên theo nguyên tắc trên.

Input	Output
<pre> 4 0 1 4 3 1 0 2 4 4 2 0 4 3 4 4 0 </pre>	10

Problem M. DP trên cây

Time Limit 2000 ms

Mem Limit 1024 kB

Có N bông hoa được xếp thành một hàng trong vườn hoa của Taro. Với mỗi i ($1 \leq i \leq N$), chiều cao và vẻ đẹp của bông hoa thứ i từ trái sang phải lần lượt là h_i và a_i . Trong vườn hoa, chiều cao của các bông hoa h_1, h_2, \dots, h_N phân biệt đôi một với nhau.

Taro đang nhổ một số bông hoa đi để những bông hoa còn lại trong vườn thỏa mãn điều kiện sau đây:

- Chiều cao của các bông hoa còn lại tăng dần từ trái sang phải.

Taro muốn vườn hoa của mình là đẹp nhất có thể (dĩ nhiên, ai mà chẳng muốn như vậy), vậy nên Taro muốn bỏ đi các bông hoa sao cho tổng vẻ đẹp của những bông hoa còn lại là lớn nhất. Tuy nhiên, Taro đang bận làm bài tập nên đã nhờ bạn làm giúp việc này. Hãy giúp Taro và in ra tổng vẻ đẹp lớn nhất có thể của các bông hoa còn lại.

Input

Dòng thứ nhất chứa một số nguyên N ($1 \leq N \leq 2 \times 10^5$).

Dòng thứ hai chứa N số nguyên h_i ($1 \leq h_i \leq N$) phân biệt đôi một.

Dòng thứ ba chứa N số nguyên a_i ($1 \leq a_i \leq 10^9$).

Output

In ra tổng vẻ đẹp lớn nhất có thể của những bông hoa còn lại.

Sample 1

Input	Output
4 3 1 4 2 10 20 30 40	60

Giải thích

Tổng vẻ đẹp lớn nhất của vườn hoa có thể là 60, bằng cách giữ lại các bông hoa thứ hai và thứ tư từ trái qua phải.

Sample 2

Input	Output
1 1 10	10

Giải thích

Vườn hoa đã thỏa mãn điều kiện đề bài từ ban đầu.

Sample 3

Input	Output
5 1 2 3 4 5 1000000000 1000000000 1000000000 1000000000 1000000000	5000000000

Sample 4

Input	Output
9 4 2 5 8 3 6 1 7 9 6 8 8 4 6 3 5 7 5	31

Giải thích

Tổng vẻ đẹp lớn nhất của vườn hoa có thể là 31, bằng cách giữ lại các bông hoa thứ hai, thứ ba, thứ sáu, thứ tám và thứ chín từ trái qua phải.

Lưu ý

Đáp án có thể vượt quá giới hạn của kiểu biến số nguyên 32-bit.

Problem N. DP chia n nhóm

Time Limit 1000 ms

Mem Limit 524288 kB

Cho N số nguyên dương A_1, A_2, \dots, A_N . Mỗi số có giá trị không vượt quá N .

Yêu cầu:

Hãy chia N số này thành một số nhóm sao cho:

- Mỗi nhóm là một dãy các số liên tiếp.
- Trọng số của mỗi nhóm được tính theo công thức: Bình phương của số giá trị khác nhau trong nhóm đó.

Ví dụ:

- 1 2 3 \rightarrow số giá trị khác nhau bằng 3, trọng số = 9.
- 1 2 1 \rightarrow số giá trị khác nhau bằng 2, trọng số = 4.
- 1 1 1 \rightarrow số giá trị khác nhau bằng 1, trọng số = 1.

Trọng số của dãy số bằng Tổng trọng số của tất cả các nhóm. Hãy tìm cách chia sao cho Tổng trọng số đạt giá trị nhỏ nhất.

Input

Dòng 1: Gồm 2 số N và M - Số lượng số và số giá trị khác nhau trong dãy.

Dòng 2 ... $N + 1$: Mỗi dòng chứa một số nguyên.

Output

Trọng số nhỏ nhất của dãy số.

Input	Output
13 4 1 2 1 3 2 2 3 4 3 4 3 1 4	11

Note

Giải thích: Chúng ta sẽ chia dãy số thành 8 nhóm

- 4 nhóm đầu tiên mỗi nhóm chứa một số duy nhất.
- nhóm thứ 5 chứa hai số nguyên đều có giá trị là 2.
- nhóm thứ 6 gồm các số từ 7 \rightarrow 12: (3, 4, 3, 4, 3)
- 2 nhóm cuối cùng mỗi nhóm gồm 1 số duy nhất.

Kết quả là $1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 2^2 + 1^2 + 1^2 = 11$.

Giới hạn: $N \leq 40000$.

Problem O. DP on sets

Time Limit 2000 ms

Code Length Limit 50000 B

OS Linux

Read problems statements in [Mandarin Chinese](#) and [Russian](#) as well.

Let's consider a set S of N different elements numbered from 0 to $N - 1$. It's a well-known fact that there are exactly 2^N subsets of S (including the empty subset). Each subset S' of S can be encoded as a bitmask $B(S')$ containing N bits, where the i 'th bit of $B(S')$ is equal to 1 if the i 'th element of S belongs to S' , and 0 otherwise. Each bitmask can also be considered as a non-negative integer represented in binary.

For example, suppose N is equal to 5 . Then $S = \{0, 1, 2, 3, 4\}$. Let's assume $S' = \{0, 3, 4\}$.

Then $B(S') = 1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 + 1 \times 2^4 = 11001_2 = 25_{10}$.

Let's say that a triple (A, B, C) of subsets of S covers a subset D of S , if D is a subset of the union of subsets A , B and C . In other words, every element of D is an element of at least one of A , B , or C .

Let's consider four functions F , G , H and R . Each takes a subset of S as the only parameter and returns a non-negative integer. You are given the values of $F(i)$, $G(i)$, and $H(i)$ for each $0 \leq i < 2^N$.

The value of the function R of a subset X of S is equal to the sum of $F(A) \times G(B) \times H(C)$ for all triples (A, B, C) of subsets of S that cover X .

Your task is to calculate $R(0) + R(1) + \dots + R(2^N - 1)$ modulo $1000000007(10^9 + 7)$.

Input

The first line of the input contains one integer N .

The second line of the input contains 2^N integers, denoting the values of $F(0)$, $F(1)$, ..., $F(2^N - 1)$. The third and the fourth lines of the input contains the values of G and H in the same format.

Output

The output should contain one integer: $R(0) + R(1) + \dots + R(2^N - 1)$ modulo $1000000007(10^9 + 7)$.

Constraints

$$1 \leq N \leq 20;$$

$$0 \leq F_i, G_i, H_i < 1,000,000,007(10^9 + 7).$$

Example

Input:

```
2
1 3 9 12
0 5 1 2
2 3 4 1
```

Output:

```
7680
```

Explanations

Here's the table of what sets are covered by all the possible triples for $N = 1$:

A	B	C	D
0	0	0	{0}
0	0	1	{0, 1}
0	1	0	{0, 1}
0	1	1	{0, 1}
1	0	0	{0, 1}
1	0	1	{0, 1}
1	1	0	{0, 1}
1	1	1	{0, 1}

Problem P. DP IOI

Time Limit 1000 ms

Mem Limit 524288 kB

Có hai mỏ than, mỗi mỏ có một nhóm thợ mỏ làm việc. Khai thác than là công việc vất vả, do đó các thợ mỏ cần thực phẩm để hoạt động. Mỗi khi một đợt vận chuyển thực phẩm đến mỏ, các thợ mỏ sẽ khai thác được một lượng than nào đó. Có 3 loại thực phẩm được vận chuyển: thịt, cá và bánh mì.

Các thợ mỏ muốn thực đơn đa dạng. Họ sẽ làm việc với năng suất cao hơn nếu được cung cấp nguồn thực phẩm đa dạng. Chính xác hơn, mỗi khi một đợt vận chuyển đến mỏ, họ sẽ so sánh đợt vận chuyển này với hai đợt vận chuyển liền trước (hoặc ít hơn nếu chưa đủ):

- Nếu các đợt vận chuyển cùng một loại thực phẩm, họ sẽ sản xuất được 1 đơn vị than.
- Nếu có 2 loại thực phẩm khác nhau trong các đợt vận chuyển, họ sẽ sản xuất được 2 đơn vị than.
- Nếu có 3 loại thực phẩm khác nhau trong các đợt vận chuyển, họ sẽ sản xuất được 3 đơn vị than.

Biết trước các loại thực phẩm và thứ tự chúng được vận chuyển, ta có thể tác động lên lượng than sản xuất được bằng cách chỉ định đợt vận chuyển nào sẽ đến mỏ than nào.

Các đợt vận chuyển không thể được chia nhỏ, lượng thực phẩm của mỗi đợt phải được gửi toàn bộ đến một trong hai mỏ.

Hai mỏ than không nhất thiết phải nhận số đợt vận chuyển như nhau (thậm chí có thể gửi tất cả các đợt vận chuyển đến một mỏ).

Biết các loại thực phẩm theo thứ tự chúng được vận chuyển, hãy tìm tổng lượng than lớn nhất có thể sản xuất (trong cả hai mỏ) bằng cách quyết định đợt vận chuyển nào sẽ được gửi đến mỏ 1, đợt vận chuyển nào sẽ được gửi đến mỏ 2.

Input

- Dòng đầu tiên chứa 1 số nguyên dương N ($1 \leq N \leq 100000$), số đợt vận chuyển thực phẩm.
- Dòng thứ hai chứa một chuỗi gồm N ký tự, cho biết các loại thực phẩm theo thứ tự chúng được phân phối. Mỗi ký tự sẽ có dạng một trong 3 chữ cái in hoa: M (thịt), F (cá) hoặc B (bánh mì).

Output

In ra một số nguyên duy nhất, là tổng lượng than lớn nhất có thể sản xuất được.

Input	Output
6 MBMFFB	12
Input	Output
16 MMBMBBBBMMMMBMB	29

Note

Trong ví dụ đầu tiên, bằng cách phân phối các chuyến hàng theo thứ tự: mỏ 1, mỏ 1, mỏ 2, mỏ 2, mỏ 1, mỏ 2, lượng than sản xuất được sẽ lần lượt là 1, 2, 1, 2, 3, 3. Tổng lượng than là 12 đơn vị. Có thể phân phối theo cách khác để đạt được tổng lượng than này.

Problem Q. DP on numbers

Time Limit 7500 ms

Mem Limit 524288 kB

Cho một lưới ô vuông gồm m dòng và n cột. Các dòng được đánh số từ 1 đến m từ trên xuống dưới, các cột được đánh số từ 1 đến n từ trái qua phải. Ô nằm ở vị trí dòng i và cột j của lưới được gọi là ô (i, j) và khi đó, i được gọi là tọa độ dòng còn j được gọi là tọa độ cột của ô này. Trên ô (i, j) của lưới ghi số nguyên dương a_{ij} , $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$. Trên lưới đã cho, từ ô (i, j) ta có thể di chuyển đến ô (p, q) nếu các điều kiện sau đây được thỏa mãn:

- $j < n$; $i \leq p$; $j \leq q$ và $i + j < p + q$;
- a_{ij} và a_{pq} có ước số chung lớn hơn 1.

Ta gọi một cách di chuyển từ mép trái sang mép phải của lưới là cách di chuyển bắt đầu từ một ô có tọa độ cột bằng 1 qua các ô của lưới theo qui tắc di chuyển đã nêu và kết thúc ở một ô có tọa độ cột bằng n .

Yêu cầu: Tính số cách di chuyển từ mép trái lưới sang mép phải lưới.

Input

- Dòng đầu tiên ghi 2 số nguyên dương m, n .
- Dòng thứ i trong số m dòng tiếp theo ghi n số nguyên dương $a_{i1}, a_{i2}, \dots, a_{in}$ là các số trên dòng thứ i của lưới, $i = 1, 2, \dots, m$.

Hai số liên tiếp trên cùng một dòng được ghi cách bởi ít nhất một dấu cách.

Output

Ghi ra 1 số nguyên là phần dư của số lượng cách di chuyển tìm được cho 10^9 .

Giới hạn

- Trong tất cả các test: $1 < m, n \leq 100$; $a_{ij} \leq 30000$, $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$.
- Có 50% số lượng test với $m, n \leq 50$.

Input	Output
2 2 2 4 6 8	4
Input	Output
2 2 2 5 6 7	0

TỔNG QUAN VỀ ĐỀ BÀI

STT	Tên file bài làm	Tên file dữ liệu	Tên file kết quả	Giới hạn mỗi test	Điểm
1	SPELL.*	SPELL.INP	SPELL.OUT	1 giây/1 GB	100
2	KIOSKS.*	KIOSKS.INP	KIOSKS.OUT	1 giây/1 GB	100
3	BITSTR.*	BITSTR.INP	BITSTR.OUT	1 giây/1 GB	100

Dấu * được thay bằng phần mở rộng PAS, C hay CPP của ngôn ngữ lập trình được sử dụng.

Đề thi có 4 trang.

Hãy lập chương trình giải các bài toán sau đây

Bài 1. GHÉP CHỮ

Các bé trường mầm non SuperKids đang chơi trò chơi ghép chữ trên một sân hình chữ nhật kích thước $m \times n$ được chia thành lưới ô vuông đơn vị. Các hàng ô đánh số từ 1 tới m từ trên xuống và các cột ô được đánh số từ 1 tới n từ trái qua phải. Ô nằm trên giao của hàng i và cột j được gọi là ô (i, j) và trên đó chứa đúng một chữ cái hoa a_{ij} .

Mỗi bé khi chơi được cho trước xâu ký tự S độ dài k chỉ gồm các chữ cái hoa. Bé được chọn một ô để xuất phát và thực hiện trò chơi qua nhiều lượt. Tại mỗi lượt, bé **bắt buộc** phải di chuyển sang một trong bốn ô kề cạnh với ô đang đứng, sau đó bé được **viết ra đúng một chữ cái** bằng với chữ cái tại ô vừa tới **nếu muốn**. Mục đích của bé là viết ra được xâu ký tự S đã cho. Chú ý rằng các chữ cái phải được viết ra lần lượt theo đúng thứ tự trong xâu S và khi tới một ô chỉ được viết ra đúng một chữ cái.

Yêu cầu: Hãy giúp bé thực hiện được mục đích của mình trong trò chơi với số lần di chuyển ít nhất. Cho biết số lần di chuyển theo phương án tìm được

Dữ liệu: Vào từ file văn bản SPELL.INP

- ✿ Dòng 1 chứa ba số nguyên dương m, n, k ($2 \leq m, n, k \leq 300$) cách nhau bởi dấu cách
- ✿ Dòng 2 chứa xâu ký tự S gồm đúng k chữ cái hoa viết liền.
- ✿ m dòng tiếp theo, dòng thứ i chứa n chữ cái hoa liền nhau, chữ cái thứ j là a_{ij}

Dữ liệu đảm bảo rằng mọi ký tự của xâu S đều có mặt trong ít nhất một ô của sân

Kết quả: Ghi ra file văn bản SPELL.OUT một số nguyên duy nhất là số lần di chuyển ít nhất mà bé cần thực hiện để đạt được mục đích của trò chơi.

Ví dụ

SPELL.INP	SPELL.OUT
2 4 6 DHDBBB DHAB ABBD	7

Giải thích

Bé có thể chọn ô (2, 1) để xuất phát. Sau đó:

(2, 1)

↓

(1, 1) → D

↓

(1, 2) → H

↓

(1, 1) → D

↓

(2, 1)

↓

(2, 2) → B

↓

(2, 3) → B

↓

(2, 2) → B

Bộ test chia làm 3 subtasks:

Subtask 1 (30% số điểm): $m, n, k \leq 4$

Subtask 2 (30% số điểm): $m, n, k \leq 100$

Subtask 3 (40% số điểm): Không có ràng buộc bổ sung ngoài các ràng buộc đã nêu trong đề bài.

Bài 2. TRUNG TÂM MUA SẮM

Một trung tâm mua sắm có n kiốt (kiosk) đánh số từ 1 tới n , kiốt thứ i bán mặt hàng c_i . Siêu thị có $n - 1$ con đường hai chiều đánh số từ 1 tới $n - 1$, con đường thứ i nối giữa kiốt u_i và kiốt v_i . Hệ thống các con đường đi đảm bảo sự đi lại giữa hai kiốt bất kỳ.

Trong thời kỳ dịch bệnh, người ta muốn ngưng hoạt động một số kiốt để dễ dàng kiểm soát các hoạt động mua bán cũng như giao tiếp với khách hàng. Khi một kiốt bị ngưng hoạt động, tất cả các con đường nối tới kiốt đó đều bị chặn để đảm bảo an ninh. Ngoài ra vì không muốn ảnh hưởng nhiều tới khách hàng, siêu thị muốn lập phương án sao cho các kiốt vẫn còn hoạt động phải thỏa mãn hai điều kiện sau:

- ⚙ Các kiốt còn hoạt động phải liên thông với nhau: Tức là giữa hai kiốt bất kỳ vẫn được mở cửa phải tồn tại đường đi (qua các con đường không bị chặn)
- ⚙ Tất cả các mặt hàng mang số hiệu từ 1 tới k (là những mặt hàng thiết yếu) đều phải có bán ở ít nhất một kiốt còn hoạt động.

Hai phương án được gọi là khác nhau nếu có một kiốt bị ngưng hoạt động trong một phương án nhưng được phép hoạt động trong phương án còn lại.

Yêu cầu: Hãy cho biết có bao nhiêu phương án khác nhau thỏa mãn điều kiện nêu trên.

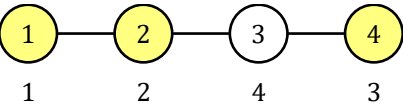
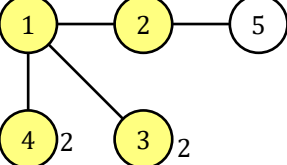
Dữ liệu: Vào từ file văn bản KIOSKS.INP

- ✿ Dòng 1 chứa số nguyên dương $n \leq 10^4, k \leq 10$
- ✿ Dòng 2 chứa n số nguyên dương c_1, c_2, \dots, c_n ($\forall i: c_i \leq n$)
- ✿ $n - 1$ dòng tiếp theo, dòng thứ i chứa hai số nguyên dương u_i, v_i

Các số trên cùng một dòng của input được ghi cách nhau bởi dấu cách

Kết quả: Ghi ra file văn bản KIOSKS.OUT một số nguyên duy nhất là số dư trong phép chia: số phương án thỏa mãn điều kiện đề bài cho 1000000007 ($10^9 + 7$)

Ví dụ

KIOSKS.INP	KIOSKS.OUT	Giải thích
4 3 1 2 4 3 1 2 2 3 3 4	1	 1 2 4 3 Không thể đóng cửa kiosk nào cả
5 2 1 2 2 2 3 1 2 1 3 1 4 2 5	11	 Các phương án có thể: {1, 2}; {1, 3}; {1, 4}; {1, 2, 3}; {1, 2, 4}, {1, 3, 4}, {1, 2, 3, 4} {1, 2, 5}; {1, 2, 3, 5}; {1, 2, 4, 5}; {1, 2, 3, 4, 5} (Các số trong {} là những kiosk được hoạt động trong phương án)

Bộ test chia làm 3 subtasks

Subtask 1 (30% số điểm): $k = 1$

Subtask 2 (30% số điểm): Mỗi kiốt có không quá 2 con đường nối tới nó.

Subtask 3 (40% số điểm): Không có ràng buộc bổ sung ngoài các ràng buộc đã nêu trong đề bài.

Bài 3. XÂU NHỊ PHÂN

Cho xâu S gồm n ký tự $\in \{0,1\}$ và số tự nhiên k . Hãy tìm cách đảo một số ít nhất các ký tự của chuỗi S (đảo ký tự 0 thành ký tự 1 hoặc ngược lại) sao cho chuỗi kết quả có thể được phân tách thành không quá k chuỗi con mà mỗi chuỗi con chỉ chứa các ký tự 0 hoặc chỉ chứa các ký tự 1.

Yêu cầu: Cho biết số ký tự ít nhất trong xâu S cần đảo.

Dữ liệu: Vào từ file văn bản BITSTR.INP

- ✿ Dòng 1 chứa hai số nguyên dương $n, k \leq 2 \cdot 10^5$ cách nhau bởi dấu cách
- ✿ Dòng 2 ghi xâu S (gồm n ký tự $\in \{0,1\}$ viết liền nhau)

Kết quả: Ghi ra file văn bản BITSTR.OUT một số nguyên duy nhất là số ký tự ít nhất trong xâu S cần đảo.

Ví dụ

BITSTR.INP	BITSTR.OUT	Giải thích
10 1 1000100011	4	Biến đổi thành xâu gồm toàn ký tự 0 0000000000
6 2 010110	2	Biến đổi thành: 000111 hoặc 111110 hoặc 011111

Bộ test chia làm 5 subtasks:

Subtask 1 (20% số điểm): $n \leq 20$

Subtask 2 (20% số điểm): $n, k \leq 400$

Subtask 3 (20% số điểm): $n \leq 2 \cdot 10^5; k \leq 400$

Subtask 4 (20% số điểm): $n \leq 2 \cdot 10^5; k \leq 5000$

Subtask 5 (20% số điểm): Không có ràng buộc bổ sung ngoài các ràng buộc đã nêu trong đề bài.

🌀 Hết 🌀

Problem S. DP on computer

Time Limit 1000 ms

Mem Limit 524288 kB

Công ty phần mềm XYZ mới mua x máy tính để bàn và y máy tính xách tay. Giá một chiếc máy tính để bàn là a đôla còn giá một chiếc máy tính xách tay là b đôla. Để tránh sự thất lạc giữa các phòng ban, Tổng giám đốc đã đưa ra cách phân bổ các máy tính này về n phòng ban như sau:

- Sắp xếp n phòng ban theo thứ tự về mức độ quan trọng của các phòng ban.
- Tiến hành phân bổ các máy tính cho các phòng ban bảo đảm nếu phòng ban i có mức độ quan trọng nhỏ hơn mức độ quan trọng của phòng ban j thì tổng giá trị máy tính được phân bổ cho phòng ban i không được vượt quá tổng giá trị máy tính được phân bổ cho phòng ban j .
- Phòng ban nhận được tổng giá trị máy tính nhỏ nhất là lớn nhất.

Là một lập trình viên giỏi nhưng lại thuộc phòng ban có mức độ quan trọng nhỏ nhất, Khanh muốn chứng tỏ tay nghề của mình với đồng nghiệp nên đã lập trình tính ra ngay được tổng giá trị máy tính mà phòng ban mình nhận được rồi mời bạn tính lại thử xem! Yêu cầu Cho x, a, y, b, n . Hãy tính tổng giá trị máy tính mà phòng Khanh nhận được.

Input

Gồm 2 bộ dữ liệu, mỗi bộ trên một dòng, mỗi dòng chứa 5 số nguyên dương x, a, y, b, n (các số có giá trị không vượt quá 1000). Giới hạn 50% số test có x, a, y, b, n không vượt quá 100.

Output

Gồm hai dòng là mỗi dòng là đáp án tương ứng với bộ dữ liệu vào.

Input	Output
3 300 2 500 2	900
4 300 3 500 2	1300

Problem T. DP at home

Time Limit 1000 ms

Mem Limit 262144 kB

Trò chơi Đấu Trường Chân Lý do VNGGames phát hành đạt giải thưởng Game của năm tại đêm trao giải Vietnam Game Awards 2024.

Để ăn mừng cho những thành công đạt được, VNGGames tổ chức các trò chơi giải trí cho thành viên trong team. Ban Tổ Chức sẽ giao nhiệm vụ cho các đội chơi, mỗi đội chơi gồm 2 người. Nhiệm vụ cụ thể như sau:

Có n nhiệm vụ, được đánh số từ 1 đến n . Nhiệm vụ i thuộc loại $t[i]$.

- Nếu nhiệm vụ i được giao cho người đã hoàn thành nhiệm vụ ngay trước đó cùng loại thì thời gian thực hiện là $b[t[i]]$.
- Nếu không, thời gian thực hiện là $a[t[i]]$.

Mục tiêu của đội chơi là phân chia công việc sao cho phải hoàn thành tất cả nhiệm vụ trong thời gian sớm nhất. Các bạn hãy tính toán xem thời gian tốt nhất để một đội chơi có thể hoàn thành tất cả nhiệm vụ là bao nhiêu.

Lưu ý:

- Nhiệm vụ phải được hoàn thành theo thứ tự để cho.
- Trong quá trình thực hiện nhiệm vụ, không ai được nghỉ quá d nhiệm vụ liên tiếp.
- Mỗi nhiệm vụ chỉ cần một trong hai người hoàn thành.

Input

Gồm nhiều test cases. Dòng đầu tiên là một số nguyên T , số lượng test cases ($T \leq 10$).

Dòng đầu tiên của mỗi test case là n, k và d ($1 \leq n, k, d \leq 5000$) lần lượt là số nhiệm vụ, số loại nhiệm vụ và khoảng nghỉ tối đa của một người.

Dòng thứ 2 trong mỗi test case là n số nguyên t_1, t_2, \dots, t_n ($1 \leq t_i \leq k$) là loại của nhiệm vụ thứ i .

Dòng thứ 3 trong mỗi test case là k số nguyên a_1, a_2, \dots, a_k ($1 \leq a_i \leq 10^9$) là thời gian để người đó hoàn thành công việc loại i trong trường hợp thứ 2.

Dòng thứ 4 trong mỗi test case là k số nguyên b_1, b_2, \dots, b_k ($1 \leq b_i \leq a_i$) là thời gian để người đó hoàn thành công việc loại i trong trường hợp đầu tiên.

Output

Mỗi dữ liệu được ghi trên một dòng là thời gian ngắn nhất mà một đội chơi có thể hoàn thành xong nhiệm vụ của mình.

Input	Output
1 4 3 4 1 3 1 3 4 7 8 2 7 7	21

Subtask

- 20% số test có $n \leq 20, T = 1$.
- 50% số test tiếp theo có $d = n$.
- 30% giới hạn đề bài.

Giải thích

2 người A và B sẽ làm thứ tự như sau: $ABAB$ thì thời gian để hoàn thành là: $4 + 8 + 2 + 7 = 21$.

Problem U. DP at leaves

Time Limit 1000 ms

Mem Limit 524288 kB

Một ngày thu đẹp trời, Radu và Mars nhận ra rằng khu vườn của họ chứa đầy lá rụng. Họ quyết định gom lá thành đúng K đồng lá.

Biết rằng khu vườn có dạng một đường thẳng. 2 người đã thiết lập một hệ tọa độ với gốc ở điểm đầu của khu vườn.

Có N chiếc lá nằm thẳng hàng với trọng lượng khác nhau, khoảng cách giữa 2 chiếc lá liên tiếp là 1. Nghĩa là, chiếc lá đầu tiên có tọa độ 1, chiếc lá thứ 2 có tọa độ 2, ..., chiếc lá thứ N có tọa độ N . Ban đầu, 2 người đang đứng ở tọa độ N .

Radu và Mars thực hiện việc gom lá trong khi rời khỏi khu vườn, do đó những chiếc lá chỉ có thể di chuyển về bên trái. Chi phí di chuyển một chiếc lá bằng tích của trọng lượng chiếc lá và khoảng cách di chuyển. Hiển nhiên, một trong K đồng lá sẽ nằm ở tọa độ 1, tuy nhiên những đồng còn lại có thể nằm ở bất kỳ vị trí nào.

Yêu cầu: tìm chi phí nhỏ nhất để gom N chiếc lá thành đúng K đồng lá.

Input

- Dòng đầu tiên chứa 2 số nguyên dương N và K , cách nhau bởi 1 khoảng trắng.
- Dòng thứ i trong số N dòng tiếp theo chứa 1 số nguyên dương cho biết trọng lượng của chiếc lá thứ i .

Output

In ra 1 số nguyên là chi phí nhỏ nhất để gom N chiếc lá lại thành đúng K đồng lá.

Giới hạn

- $0 < N \leq 100000$
- $0 < K \leq 10, K < N$
- Trọng lượng của mỗi chiếc lá không vượt quá 1000.

Input	Output
5 2 1 2 3 4 5	13

Note

Cách tốt nhất là đặt 2 đồng lá ở vị trí 1 và 4.

Problem V. DP bao hàm loại trừ

Time Limit 1000 ms

Mem Limit 262144 kB

Có n cái kẹo trên bàn; người ta muốn cất kẹo vào trong một số túi sao cho mỗi túi chứa một số các viên kẹo nằm liên tiếp nhau trên bàn.

Đồng thời, mỗi túi có không quá m viên kẹo, và số kẹo trong hai túi liên tiếp hơn kém nhau không quá d viên.

Yêu cầu: Bạn hãy tính số cách chia kẹo vào túi thỏa mãn nhé. Hai cách chia được coi là khác nhau nếu số túi sử dụng là khác nhau, hoặc tồn tại một túi sao cho trong hai cách, số lượng kẹo tại túi đó khác nhau.

Input

Một dòng chứa ba số nguyên dương n, m, d ($n \leq 10^{18}, d \leq m \leq 10$).

Output

In ra một số nguyên là số cách chia kẹo. Vì đáp án có thể rất lớn nên hãy in ra số dư khi chia cho $10^9 + 7$.

Scoring

Subtask	Điểm	Giới hạn
1	20	$n \leq 10^5$
2	30	$m = 2$
3	50	Không có ràng buộc gì thêm

Input	Output
5 3 1	10

Notes

Ta có 10 cách chia:

- (1, 1, 1, 1, 1)
- (1, 1, 1, 2)
- (1, 1, 2, 1)
- (1, 2, 1, 1)
- (2, 1, 1, 1)
- (1, 2, 2)
- (2, 2, 1)
- (2, 1, 2)
- (2, 3)
- (3, 2)

Chú thích: Phần tử thứ i là số viên kẹo trong túi thứ i của cách chia.

Problem W. DP SOS

Time Limit 1000 ms

Mem Limit 262144 kB

You are given two integers l and r ($l \leq r$). Your task is to calculate the sum of numbers from l to r (including l and r) such that each number contains **at most** k different digits, and print this sum modulo 998244353.

For example, if $k = 1$ then you have to calculate all numbers from l to r such that each number is formed using only one digit. For $l = 10, r = 50$ the answer is $11 + 22 + 33 + 44 = 110$.

Input

The only line of the input contains three integers l, r and k ($1 \leq l \leq r < 10^{18}, 1 \leq k \leq 10$) — the borders of the segment and the maximum number of different digits.

Output

Print one integer — the sum of numbers from l to r such that each number contains at most k different digits, modulo 998244353.

Examples

Input	Output
10 50 2	1230

Input	Output
1 2345 10	2750685

Input	Output
101 154 2	2189

Note

For the first example the answer is just the sum of numbers from l to r which equals to $\frac{50 \cdot 51}{2} - \frac{9 \cdot 10}{2} = 1230$. This example also explained in the problem statement but for $k = 1$.

For the second example the answer is just the sum of numbers from l to r which equals to $\frac{2345 \cdot 2346}{2} = 2750685$.

For the third example the answer is $101 + 110 + 111 + 112 + 113 + 114 + 115 + 116 + 117 + 118 + 119 + 121 + 122 + 131 + 133 + 141 + 144 + 151 = 2189$.

Problem X. DP cut tree

Time Limit 1000 ms

Mem Limit 524288 kB

Hùng đang làm việc trong Công ty cao su X . Công ty có rừng cao su rất rộng, với những hàng cây cao su trồng cách đều thẳng tắp. Theo định kỳ, người ta thường phải chặt hạ cả hàng cây cao su đã hết hạn khai thác để trồng thay thế bằng hàng cây mới.

Hùng phát hiện ra một bài toán tin học liên quan đến vấn đề này: Một nhóm công nhân được giao nhiệm vụ chặt hạ hàng cây gồm n cây được trồng dọc theo một đường thẳng với khoảng cách cố định giữa hai cây liên tiếp. Nếu các công nhân cưa đổ một cây, họ có thể cho nó đổ về phía bên trái hoặc bên phải dọc theo hàng cây. Một cây khi đổ có thể lật đổ cây khác bị nó rơi vào và có thể làm đổ nhiều cây khác, theo hiệu ứng lan truyền domino. Sau khi khảo sát kỹ, Hùng đã mô tả được hiệu ứng lan truyền domino như sau: Giả sử các cây trên hàng cây được đánh số từ 1 đến n , từ trái qua phải và chiều cao của cây i là h_i ($1 \leq i \leq n$).

- Nếu cây i đổ về bên trái thì tất cả các cây j với $i - h_i < j < i$ cũng sẽ đổ;
- Nếu cây i đổ về bên phải thì tất cả các cây j với $i < j < i + h_i$ cũng sẽ đổ;
- Mỗi cây chỉ đổ một lần về bên trái hoặc bên phải.

Do đó bài toán đặt ra đối với Hùng là: Xác định số lượng nhỏ nhất các cây mà các công nhân cần cưa đổ đảm bảo hạ đổ toàn bộ hàng cây.

Yêu cầu: Giúp Hùng giải quyết bài toán đặt ra.

Input

- Dòng đầu tiên ghi số nguyên dương n ;
- Dòng thứ hai chứa n số nguyên dương h_1, h_2, \dots, h_n được ghi cách nhau bởi dấu cách, mỗi số không vượt quá 10^6 .

Output

- Dòng đầu tiên ghi số nguyên dương k là số lượng cây mà các công nhân cần cưa đổ;
- Dòng thứ hai ghi dãy số nguyên c_1, c_2, \dots, c_k trong đó $|c_j|$ ($1 \leq j \leq k$) là dãy chỉ số của các cây theo thứ tự các công nhân phải lần lượt cưa đổ, c_j là số dương nếu cây cần

cho đổ về bên phải và là số âm nếu cây cần cho đổ về bên trái. Nếu có nhiều cách thì chỉ cần đưa ra một cách tùy ý.

Giới hạn

- Có 40% số test ứng với 40% số điểm của bài có $1 \leq n \leq 10^4$.
- Có 40% số test khác ứng với 40% số điểm của bài có $1 \leq n \leq 10^5$.
- Có 20% số test còn lại ứng với 20% số điểm của bài có $1 \leq n \leq 4 * 10^6$.

Input	Output
5 1 2 3 1 1	2 2 -1

Note

Còn một cách đốn cây khác: Cưa hai cây 1 và 2 và đều cho đổ về bên phải.

Problem Y. DP on functions

Time Limit 1000 ms

Code Length Limit 50000 B

OS Linux

While playing with arrays, Ira came across this problem. You are given an array $A[0..N-1]$. You are needed to compute 2 functions F and G . They are defined as follows.

- $F[i] = A[i]^2 + (\sum G[j])^2$ where $(i \& j) = j$ and $j < i$
- $G[i] = (\sum F[j]^2)$ where $(i \& j) = j$ and $j \leq i$

You need to print the summation $(\sum i \times F[i] \times G[i]) \bmod 10^9+7$.

Note : Here $\&$ denotes [bitwise and operator](#).

Input

- First line contains the integer N .
- N integers follow on the second line denoting the array $A[0..N-1]$.

Output

Output one integer: the answer to the problem.

Constraints

- $1 \leq N \leq 6 \cdot 10^5$
- $1 \leq A[i] \leq 10^9$

Example

Input:

5
1 1 1 2 3

Output:

5866820

Explanation

$$F[0] = A[0] * A[0] = 1$$

$$G[0] = F[0] * F[0] = 1 * 1 = 1$$

$$F[1] = A[1] * A[1] + G[0]^2 = 1 + 1^2 = 2$$

$$G[1] = F[0] * F[0] + F[1] * F[1] = 1 * 1 + 2 * 2 = 5$$

$$F[2] = A[2] * A[2] + G[0]^2 = 1 + (1)^2 = 2$$

$$G[2] = F[0] * F[0] + F[2] * F[2] = 1 * 1 + 2 * 2 = 5$$

$$F[3] = A[3] * A[3] + (G[0] + G[1] + G[2])^2 = 4 + (1 + 5 + 5)^2 = 125$$

$$G[3] = F[0] * F[0] + F[1] * F[1] + F[2] * F[2] + F[3] * F[3] = 1 * 1 + 2 * 2 + 2 * 2 + 125 * 125 = 15634$$

$$F[4] = A[4] * A[4] + G[0]^2 = 9 + 1^2 = 10$$

$$G[4] = F[0] * F[0] + F[4] * F[4] = 1 * 1 + 10 * 10 = 101$$

$$\text{ans} = 0 * 1 * 1 + 1 * 2 * 5 + 2 * 2 * 5 + 3 * 125 * 15634 + 4 * 10 * 101 = 5866820$$

Problem Z. DP party

Time Limit 1000 ms

Mem Limit 524288 kB

Marin là người tốt, vì thế anh quyết định tổ chức Q bữa tiệc cho N người bạn của anh ấy, và đương nhiên, mọi người đều là những cp-er "siu vjpro". Đồ uống duy nhất được phục vụ trong mỗi buổi tiệc sẽ là dzumbus - hỗn hợp của cola và nước gừng.

Và vì là người tốt nên Marin biết từng người bạn của anh ấy muốn uống bao nhiêu dzumbus trong bữa tiệc đó để họ được "thoải mái". Anh ấy cũng biết chuyện sẽ có M cặp những người bạn sẽ bắt đầu trao đổi về kì thi COCI vừa qua trong suốt mỗi buổi nếu cả hai người trong cặp đó đều "thoải mái".

Khi một người A chia sẻ cách làm của họ cho B thì B cũng chia sẻ lại cách làm của họ cho A . Từ đó M cặp được hình thành sao cho: trong suốt buổi tiệc, với thứ tự bất kì của các cuộc trao đổi, không xảy ra trường hợp cách làm của A được chia sẻ ngược lại với anh ấy. Tại mỗi buổi tiệc Marin sẽ chọn cách chia đồ uống sao cho tối đa hóa được số người trao đổi ít nhất một lần trong buổi tiệc đó. Nhiệm vụ của bạn là tính số người sẽ trao đổi cách làm của họ ít nhất một lần trong mỗi bữa tiệc trong Q bữa tiệc.

Input

Dòng đầu tiên gồm hai số nguyên N, M lần lượt là số người bạn của Marin và số cặp sẽ thảo luận khi họ thấy "thoải mái".

Dòng thứ hai gồm N số nguyên D_i , số lượng dzumbus mà người bạn thứ i cần uống để "thoải mái".

Ở dòng thứ i trong M dòng tiếp theo, chứa hai số nguyên A_i và B_i ($A_i \neq B_i$).

Dòng tiếp theo chứa một số nguyên Q là số bữa tiệc của Marin.

Q dòng tiếp theo, mỗi dòng chứa một số nguyên S_i là tổng số dzumbus sẽ có trong bữa tiệc thứ i .

Output

In ra Q dòng, mỗi dòng là số người sẽ trao đổi được cách làm của mình với người khác.

Phân bố điểm

Ở mọi subtask điều kiện sau luôn được đảm bảo: $0 \leq M < N \leq 1000$, $1 \leq Q \leq 2 \cdot 10^5$, $1 \leq D_i, S_i \leq 10^9$.

Subtask 1 gồm 2 test: $M = N - 1$, $1 \leq S_i \leq 1000$ và mỗi người bạn của Marin sẽ ghép đôi để trao đổi với tối đa hai người khác.

Subtask 2 gồm 3 test: $M = N - 1$ và mỗi người bạn của Marin sẽ ghép đôi để trao đổi với tối đa hai người khác.

Subtask 3 gồm 3 test: $N \leq 100$.

Subtask 4 gồm 3 test: Không có ràng buộc gì thêm.

Examples

Input	Output
1 0 1000 1 1000	0
Input	Output
3 2 1 2 3 1 2 1 3 3 2 3 5	0 2 2

Sample output 3

1	14 13
2	2 3 4 19 20 21 5 22 6 7 23 8 10 14
3	1 2
4	1 3
5	1 4
6	2 5
7	2 6
8	3 7
9	3 8

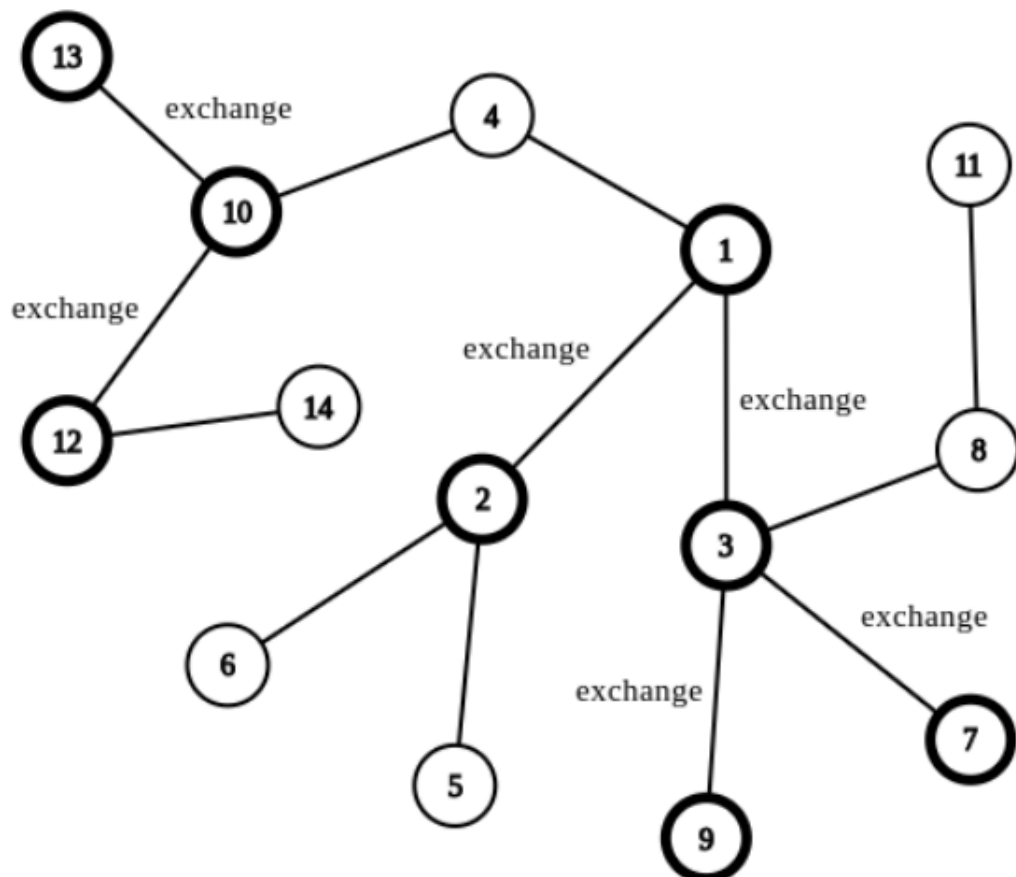
10	3 9
11	4 10
12	8 11
13	10 13
14	10 12
15	12 14
16	3
17	45
18	44
19	23

Sample output 3

1	8
2	7
3	5

Giải thích cho ví dụ thứ ba:

- Ở bữa tiệc đầu tiên, Marin quyết định cho những người bạn với chỉ số 1, 2, 3, 7, 9, 10, 12 và 13 được thoải mái. và họ uống tổng cộng 45 đơn vị dzumbus.



Problem AA. DP on loan

Time Limit 2000 ms

Mem Limit 262144 kB

Ivica và tập đoàn (gồm N công ty anh ấy sở hữu) đang gặp khó khăn. Các công ty đang trong tình cảnh nợ nần chồng chất nên các luật sư đang đến để tước các công ty từ anh ấy. May mắn là Ivica đã bàn bạc nhằm giữ lại cho anh ấy một số công ty bất chấp nợ nần.

Các luật sư đã đặt N tờ giấy kê nợ của các công ty của Ivica trên bàn. Số nợ của công ty thứ nhất được viết trên tờ giấy thứ nhất A_1 , nợ của công ty thứ hai được viết trên A_2 , ... và nợ của công ty cuối cùng được viết trên tờ giấy A_N . Ivica đã đàm phán để cho anh ấy giữ lại các công ty A_L, A_{L+1}, \dots, A_R , (L và R thể hiện vị trí của các tờ giấy được đặt trên bàn). May mắn cho Ivica, các luật sư cũng tham nhũng. Họ bắt anh ấy lấy đoạn con những tờ giấy liên tiếp đã được đàm phán trước đó (từ tờ thứ L tới tờ thứ R), nhưng cho phép anh ấy trao 2 tờ giấy trên bàn bất kì với một chi phí nhất định (số lượng lần trao là không giới hạn). Nói rõ hơn, thao tác trao tờ giấy ở vị trí i và vị trí j tốn $|i - j|$ đồng. Ivica chỉ còn K đồng trong túi. Anh ấy dùng số tiền đang có để trao các tờ giấy sao mà tổng số nợ của các công ty của anh còn sở hữu ấy là bé nhất có thể.

Input

Dòng đầu tiên chứa các số nguyên N, L, R ($1 \leq L \leq R \leq N \leq 100$) và K ($0 \leq K \leq 10\,000$).

Dòng thứ hai chứa N số nguyên A_i ($0 \leq A_i \leq 10^6$).

Output

In ra số nguyên duy nhất thể hiện tổng lượng nợ bé nhất mà Ivica có thể có nếu anh ấy tiêu dùng K đồng của anh ấy hợp lí.

Ví dụ

Input	Output
3 2 2 1 1 2 3	1

Input	Output
5 2 3 3 21 54 12 2 0	12
Input	Output
6 4 6 100 1 2 3 4 5 6	6

Ràng buộc

- 7 test đầu có $N \leq 13$ và $R = N$.
- 6 test sau có $N \leq 50$ và $R = N$.
- 6 test sau thỏa $N \leq 50$.
- 6 test còn lại không có ràng buộc gì thêm.

Problem AB. DP in combat

Time Limit 2000 ms

Mem Limit 1024 kB

Soon enough, the VNOI token airdrop campaign will take place over a duration of n seconds. Any user with an account on the VNOJ system can participate. The rewards distributed to each participating user will be based on the number of points the user earns during the campaign. Upon registering for the event initially, the user will receive s points per second from the bonus system.

Users can use their points to invest in **three** categories: 1) upgrade the VNOJ judging system, 2) upgrade the quality of problems on VNOJ, and 3) upgrade the number of contests on VNOJ. The i -th investment category has a maximum of l_i investment levels. For each category, users must **upgrade the levels from the lowest to the highest**. To reach level j of the i -th investment category, users need to spend $c_{i,j}$ points. In return, after upgrading, users will receive an additional $r_{i,j}$ points per second.

Formally, users start at second 0 with 0 points. For each second i from 0 to n , the following happens in order:

- If $i > 0$, users receive additional points from the bonus system.
- Users can now choose to upgrade **any number of levels** for each investment category, provided that they do not exceed their current points. This decreases their points (which happens immediately) and increases their bonus reward per second (which comes into effect at the start of the next second).

As a wise investor, you participate in the campaign to maximize your points. Calculate the maximum number of points you can receive after the campaign ends.

Input

Each test consists of multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 100$). The description of each test case is as follows.

The first line contains two integers n, s ($1 \leq n, s \leq 10^9$) — the duration of the campaign and the initial points the user receives each second.

Following this are three groups of lines corresponding to the three investment categories, with the i -th group of lines for the 3 investment categories as follows.

- The first line contains a positive integer l_i ($1 \leq l_i \leq 200$) — the number of investment levels for the i -th category.
- The second line contains l_i integers $c_{i,1}, c_{i,2}, \dots, c_{i,l_i}$ ($0 \leq c_{i,j} \leq 10^6$) — the points required to upgrade the levels of the i -th investment category.
- The third line contains l_i integers $r_{i,1}, r_{i,2}, \dots, r_{i,l_i}$ ($0 \leq r_{i,j} \leq 10^6$) — the points received per second after upgrading the levels of the i -th investment category.

It is guaranteed that $L_i \leq 200$ for all $1 \leq i \leq 3$, with L_i being the sum of l_i over all test cases.

Output

For each test case, output a single integer — the maximum points that can be received after the campaign.

Scoring

Subtask	Points	Constraints
1	750	$n \leq 200, L_i \leq 50$
2	1500	No additional constraints
Total	2250	

Input	Output
2 10 5 2 7 4 2 6 1 13 8 3 10 14 23 7 15 11 4 1 7 1 1 1 1 0 1 0 1 1 1 1 0 0 1 1 0 1 7 1 0 1 1 1 0 1 0 1 1 0 1 1 0	276 16

Notes

The optimal investment strategy for the first test case is as follows:

1. In second 0, you do nothing. By the end, you have 0 points with a bonus rate of 5 points per second.
2. In second 1, you get an additional 5 points to get to 5 points, and you do not upgrade any category. By the end, you have 5 points with a bonus rate of 5 points per second.
3. In second 2, you get an additional 5 points to get to 10 points. You upgrade the first level of the third category, which costs you 10 points. By the end, you have 0 points with a bonus rate of 12 points per second.
4. In second 3, you get an additional 12 points to get to 12 points. You upgrade the first and second levels of the first category, which costs you 11 points. By the end, you have 1 point with a bonus rate of 20 points per second.
5. In second 4, you get an additional 20 points to get to 21 points. You upgrade the second level of the third category, which costs you 14 points. By the end, you have 7 points with a bonus rate of 35 points per second.
6. In second 5, you get an additional 35 points to get to 42 points. You upgrade the first level of the second category and the third level of the third category, which costs you 36 points. By the end, you have 6 points with a bonus rate of 54 points per second.

In the remaining time, you do not upgrade any further categories. In the end, you receive $(10 - 5) \times 54 + 6 = 276$ points.
