

1. Hội chợ [MARKET.*]

Hải Dương tổ chức hội chợ nông sản. Dọc đại lộ Tôn Đức Thắng, ban tổ chức đã xây dựng m gian hàng liền nhau đánh số lần lượt $1, 2, \dots, m$. Tuy nhiên chỉ có n gian hàng trong số chúng được thuê. Gian hàng thứ i được thuê có số hiệu x_i . Không có hai gian hàng được thuê có cùng số hiệu. Để tiết kiệm chi phí, ban tổ chức chỉ che mưa cho những gian hàng được thuê bằng những tấm bạt. Một tấm bạt phủ được từ gian hàng số hiệu u đến gian hàng số hiệu v ($u \leq v$) được coi là có kích thước $v - u + 1$. Giá của một tấm bạt kích thước w là C_w . Chú ý rằng những tấm bạt kích thước lớn hơn không nhất thiết phải đắt hơn những tấm bạt kích thước nhỏ hơn.

Hãy giúp ban tổ chức tính số tiền ít nhất để có thể mua bạt che tất cả các gian hàng được thuê. Chú ý rằng trong phương án tối ưu các tấm bạt có thể phủ chồng lên nhau ở một số gian hàng.

Input:

- Dòng đầu ghi hai số nguyên dương n, m ($1 \leq n \leq 5000, 1 \leq m \leq 10^5$)
- Dòng thứ hai chứa n số nguyên dương $x_1, x_2, \dots, x_n$ ($1 \leq x_i \leq m, x_i \neq x_j \forall i \neq j$)
- Dòng thứ ba chứa m số nguyên $C_1, C_2, \dots, C_m$ ($1 \leq C_i \leq 10^6$) là giá của những tấm bạt kích thước $1, 2, \dots, m$

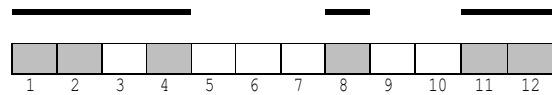
Các số liên tiếp trên cùng một dòng của file dữ liệu vào cách nhau ít nhất một dấu cách.

Output: Một số nguyên duy nhất là chi phí nhỏ nhất tìm được.

Example:

Input	Output
6 12 1 2 11 8 4 12 2 3 4 4 8 9 15 16 17 18 19 19	9

Hình minh họa ví dụ



Có 12 gian hàng, các gian màu xám là các gian được thuê. Ở trên sử dụng ba tấm bạt kích thước 4, 1 và 2

2. Nén dữ liệu [COMPRESS.*]

HD sáng tạo ra một phương pháp nén dữ liệu mới như sau:

- Cho dãy N phần tử bằng 1.
- Sử dụng các chữ số từ 1 đến A để nén dãy phần tử ban đầu sao cho tổng các phần tử sau khi nén bằng N

Ví dụ: Với $N = 5, A = 3$ ta có thể các cách nén như sau:

(1,1,1,1,1) thành (1,2,1,1); (1,2,2); (2,1,2); (3; 2)

Yêu cầu: Cho N và A . Hãy đếm số cách nén khác nhau?

Input: 1 dòng duy nhất chứa 2 số nguyên N và A ($1 \leq A \leq N \leq 50$)

Output: Một số duy nhất là số cách nén khác nhau?

Example:

input	output
2 1	1

3. Xếp pho mát [CHEESE.*]

Bác John muốn xếp các miếng Pho mát vào một cái thùng để tích trữ cho mùa đông giá rét. Độ cao của thùng chứa là T ($1 \leq T \leq 1000$) và các miếng pho mát được đặt chồng lên nhau sao cho tổng độ cao không vượt quá giá trị này. Trang trại của bác John có N ($1 \leq N \leq 100$) loại pho mát

khác nhau đánh số từ 1 đến N loại thứ i có chiều cao H_i ($5 \leq H_i \leq T$) và có giá trị V_i ($1 \leq V_i \leq 10^6$) chú ý rằng chiều cao của các miếng pho mát luôn là bội số của 5. Số lượng mỗi loại là không hạn chế. Khi các miếng pho mát được xếp chồng lên nhau thì độ cao của các miếng ở dưới bị co lại. Nếu một miếng pho mát ở dưới một miếng pho mát có độ cao lớn hơn hoặc bằng K ($1 \leq K \leq T$) thì độ cao của miếng pho mát này chỉ còn lại bằng $4/5$ độ cao ban đầu của nó. Tuy vậy, giá trị của miếng pho mát không thay đổi (vì độ cao của mỗi miếng pho mát luôn là bội của 5 nên sau khi nén lại độ cao của nó vẫn là số nguyên!!!).

Chiều cao của mỗi miếng pho mát hoặc giữ nguyên hoặc bằng $4/5$ chiều cao ban đầu (cho dù nó bị đè bởi bao nhiêu miếng pho mát bên trên nữa).

Hãy tìm tổng giá trị lớn nhất của chồng pho mát mà bác John có thể có.

Ví dụ, bác John muốn đặt pho mát vào thùng với độ cao tối đa 53, miếng pho mát sẽ bị co lại nếu như nó nằm dưới một miếng có chiều cao lớn hơn hoặc bằng 25. Trong trang trại của bác John có 3 loại pho mát như sau:

Type	Value	Height
1	100	25
2	20	5
3	40	10

Khi đó phương án tối ưu xếp pho mát của bác John là:

Type	Height	Value	
top -> [1]	25	100	
[2]	4	20	<- crushed by [1] above
[3]	8	40	<- crushed by [1] above
[3]	8	40	<- crushed by [1] above
bottom -> [3]	8	40	<- crushed by [1] above

Tổng chiều cao của các miếng pho mát sau khi xếp là:

$$25 + 4 + 8 + 8 + 8 = 53$$

Và tổng giá trị là:

$$100 + 20 + 40 + 40 + 40 = 240$$

Input:

- Dòng 1: Ghi ba số nguyên N, T và K
- Dòng 2 đến dòng N+1: Dòng $i+1$ ghi hai số nguyên V_i và H_i .

Output: Một số nguyên duy nhất là giá trị lớn nhất của chồng pho mát.

Example:

input	output
3 53 25 100 25 20 5 40 10	240

Ghi chú: Có 40% số test ứng với 40% số điểm có $n \leq 20$

4. Biến đổi đối xứng [CHEAPPAL.*]

Cho một xâu gồm M ($M \leq 2000$) chữ cái thường trong số N ($N \leq 26$) chữ cái thường cho trước. Bạn có hai thao tác biến đổi xâu như sau:

- Xóa một ký tự c ở một vị trí bất kì với chi phí không âm $d(c) \leq 10000$.
- Chèn thêm ký tự c ở một vị trí bất kì với chi phí không âm $i(c) \leq 10000$.

Hãy tìm cách biến đổi xâu đã cho thành xâu đối xứng với chi phí nhỏ nhất.

Input:

- Dòng 1: ghi 2 số nguyên N và M.

- Dòng 2: ghi một xâu gồm M kí tự.
- N dòng tiếp: mỗi dòng ghi một kí tự la tinh thường c và 2 số nguyên u, v tương ứng là chi phí khi chèn hoặc xóa một kí tự c.

Output: Một dòng duy nhất ghi ra kết quả của bài toán.

Example:

input	output
3 4 abcb a 1000 1100 b 350 700 c 200 800	900

5. Đếm mảng [CARRAYS.*]

Bài toán đơn giản: Hãy đếm xem có bao nhiêu mảng khác nhau $a_1, a_2, \dots, a_n$ trong đó a_i nhận các giá trị nguyên dương trong đoạn $[1, M]$ sao cho tồn tại ít nhất một đoạn K giá trị liên tiếp giống nhau?

Ở đây hai mảng được gọi là khác nhau nếu như tồn tại ít nhất một vị trí mà giá trị phần tử hai mảng ở vị trí này là khác nhau.

Input: Một dòng duy nhất chứa ba số nguyên dương n, M, K ($1 \leq n, M, K \leq 10^6$)

Output: Một số nguyên duy nhất là số lượng mảng khác nhau tìm được. Con số này có thể rất lớn nên bạn chỉ cần lấy phần dư của nó khi chia cho 10^9+7

Example:

Input	Output
3 2 2	6

Giải thích: Các mảng tìm được là $(1, 1, 1), (1, 1, 2), (1, 2, 2), (2, 1, 1), (2, 2, 1), (2, 2, 2)$

6. Xóa đơn điệu - K gián đoạn [DAYK.*]

Cho một dãy số nguyên gồm N số $a_1, a_2, \dots, a_n$, $N \leq 1000$ các số hạng có giá trị tuyệt đối không vượt quá 30000 và một số nguyên dương K. Hãy tìm cách xóa đi một số ít nhất số hạng sao cho dãy còn lại là một dãy đơn điệu tăng ngoại trừ không quá K cặp số liền nhau vi phạm điều kiện này.

Input:

- Dòng thứ nhất ghi hai số nguyên dương N, K ($K \leq 10$).
- N dòng tiếp theo, mỗi dòng ghi một số hạng của dãy bắt đầu từ a_1 và cuối cùng là a_n .

Output:

Dòng thứ nhất ghi B là số lượng số hạng cần xóa đi. Nếu $B > 0$, trong B dòng tiếp theo, mỗi dòng ghi chỉ số của một số hạng bị xóa đi theo thứ tự tăng dần.

Example:

Input	Output
5 2 1 2 3 2 4	0

7. Bầu cử [ELECTION.*]

Người dân Byteland đã đi bầu cử quốc hội. Khi kết quả được công bố, các đảng phái lên kế hoạch liên minh để thành lập chính phủ.

Mỗi chính đảng có một số ghế nhất định trong quốc hội. Một liên minh bao gồm một số chính đảng được quyền thành lập chính phủ nếu có tổng số ghế của các đảng trong liên minh lớn hơn nữa số ghế quốc hội.

Một liên minh được gọi là dư nếu có thể loại bỏ một đảng nào đó ra khỏi liên minh mà số ghế còn lại vẫn quá bán. Việc loại bỏ như vậy cho phép các thành viên còn lại thông qua luật không phụ thuộc vào ý kiến của các đảng ngoài liên minh.

Các đảng được đánh số từ 1 đến n ($1 \leq n \leq 300$). Đảng thứ i giành được a_i ghế. Tổng số lượng ghế trong quốc hội không quá 100 000.

Yêu cầu: Cho biết $n, a_1, a_2, \dots, a_n$. Hãy xác định một liên minh quá bán không dư có tổng số ghế lớn nhất, chỉ ra số lượng đảng tham gia liên minh và số thứ tự của các đảng thuộc liên minh.

Input:

- Dòng đầu tiên chứa số nguyên n ,
- Dòng thứ 2 chứa n số nguyên $a_1, a_2, \dots, a_n$.

Output:

- Dòng đầu tiên chứa số nguyên p – số lượng đảng tham gia liên minh,
- Dòng thứ 2 chứa p số nguyên xác định các đảng thuộc liên minh.

Example:

Input	Output
4	2
1 3 2 4	2 4

Ghi chú: Có 60% số test có $N \leq 20$

8. Khôi phục biểu thức đúng [EXPRESS.*]

Cho một biểu thức chỉ gồm các kí tự (,), [,], {, }, hãy cho biết biểu thức đó có đúng đắn hay không. Nếu không hãy tìm cách thêm một số ít nhất các kí tự thuộc một trong 6 loại trên để nhận được biểu thức đúng.

VD: () ({ () }) (

Là biểu thức không đúng, biểu thức cần đưa ra là

([]) ({ () }) ()

Input: Một xâu thể hiện biểu thức gồm n kí tự ($n \leq 200$)

Output:

- Dòng đầu ghi một số nguyên là số lượng ít nhất các ký tự cần phải thêm vào để được một biểu thức đúng
- Dòng thứ hai ghi xâu ký tự là biểu thức đúng (Từ đầu dòng)

Example:

Input	Output
() ({ () } (	3
	([]) ({ () }) ()

9. Hộp thư điện tử [EMAIL.*]

Một người sử dụng INTERNET đặt yêu cầu nhận thông tin về một số chủ đề khác nhau từ một số địa chỉ truy nhập. Chủ của các địa chỉ truy nhập này sẽ gửi thông tin yêu cầu vào hòm thư của người đặt hàng. Mỗi thông tin nhận được từ địa chỉ truy nhập sẽ được ghi vào một danh mục trong máy của người sử dụng dưới dạng một file mà để ngắn gọn ta sẽ gọi là một thông báo. Để thuận tiện cho việc tra cứu, người sử dụng quyết định xây dựng các cặp tài liệu, mỗi cặp chứa

thông tin về cùng một chủ đề. Trước khi đọc tài liệu người sử dụng sẽ sao chép chúng từ danh mục các thông báo nhận được vào các cặp tương ứng.

Chương trình hộp thư điện tử gắn trên máy của người sử dụng cho phép sau "*một thao tác*" chuyển từ danh mục thông báo vào cặp tài liệu:

- Một thông báo từ danh mục hoặc
- Một dãy các thông báo liên tiếp nhau trong danh mục về cùng một chủ đề

Việc chuyển thông báo không nhất thiết phải bắt đầu từ đầu danh mục.

Cần tìm cách chuyển các thông báo trong danh mục vào các cặp tương ứng đòi hỏi số thao tác phải thực hiện là ít nhất.

Ví dụ: Giả sử người sử dụng muốn thu thập thông tin về các chủ đề A, B, C, D. Giả sử danh mục các thông báo nhận được theo trình tự thuộc về các chủ đề (A, C, D, C, B, B, C). Việc di chuyển vào cặp tài liệu có thể thực hiện như sau: Đầu tiên di chuyển hai thông báo B, khi đó danh mục còn lại là (A, C, D, C, C). Tiếp theo thực hiện việc di chuyển thông báo D, rồi thông báo A và cuối cùng di chuyển nốt 3 thông báo C liên nhau. Cách làm này đòi hỏi 4 thao tác.

Input: gồm một dòng chứa số nguyên dương N ($0 < N \leq 200$) là số thông báo trong danh mục, tiếp đến là N số nguyên là dãy số cấu các chủ đề của dãy các thông báo trong danh mục cần truyền.

Output: số thao tác ít nhất cần thực hiện.

Example:

Input	Output
7 1 3 4 3 2 2 3	4

10. Dãy con chung tăng dài nhất [LCIS.*]

Một dãy số $a_1, a_2, \dots, a_n$ được gọi là dãy tăng dần nếu như $a_i < a_{i+1}$ với mọi $i < n$.

Một dãy $s_1, s_2, \dots, s_k$ được gọi là dãy con của dãy $a_1, a_2, \dots, a_n$ nếu tồn tại một tập các chỉ số $1 \leq i_1 < i_2 < \dots < i_k \leq n$ thỏa mãn $a_{i_j} = s_j$. Nói cách khác, dãy số s nhận được từ việc loại bỏ đi một số phần tử của dãy a .

Cho hai dãy số nguyên, bạn hãy tìm dãy con chung tăng dài nhất của chúng.

Input:

- Dòng đầu tiên ghi số nguyên N là số phần tử của dãy thứ nhất ($1 \leq N \leq 5000$)
- Dòng thứ hai ghi N số nguyên thể hiện các phần tử trong dãy số thứ nhất. Các số nằm trong khoảng $[1, 10^9]$.
- Dòng thứ ba ghi số nguyên M là số phần tử của dãy thứ hai ($1 \leq M \leq 5000$)
- Dòng thứ tư ghi M số nguyên thể hiện các phần tử trong dãy thứ hai. Các số nằm trong khoảng $[1, 10^6]$.

Output: Một số nguyên duy nhất là số phần tử của dãy con chung tăng dài nhất tìm được

Example:

input	output
7 2 3 1 6 5 4 6 4 1 3 5 6	3
5 1 2 0 2 1 1 0 1	2