

Mục lục

Bữa tiệc — PARTY	1
Lịch trực cho y tá — NURSE	2
Trò chơi đồ thị — GRAPHGAME	3
Nghịch thế — INVERSION	4
Do tín hiệu — SIGNAL	5
Đánh giá hiệu quả làm việc — CHECKPOINT	6
Xếp bò vào chuồng — COW	7
Chơi bi-a — BIA	8
Đặt cửa — BLADE	9
Chèn phép toán — OPERATOR	10
Cắt đá cẩm thạch — MARBLE	11
Gõ phím — KEYBOARDING	12

Bài A. Bữa tiệc

File dữ liệu vào: `party.inp`
File kết quả: `party.out`
Hạn chế thời gian: 0.1 giây
Hạn chế bộ nhớ: 512 Mb

Hôm nay Linh mời khách tới tham dự bữa tiệc sinh nhật của mình. Tham gia bữa tiệc có tất cả n người bao gồm cả Linh. Mọi người trong bữa tiệc ngồi xung quanh một chiếc bàn lớn gồm đúng n chỗ ngồi.

Linh rất muốn nói chuyện với nhiều khách nhưng không muốn to tiếng làm ảnh hưởng đến cả bàn. Vì vậy Linh thỉnh thoảng lại nhờ người bên trái hoặc bên phải đổi chỗ ngồi cho mình. Dĩ nhiên tất cả các khách mời đều sẵn sàng chấp nhận yêu cầu của Linh.

Sau khi tiễn khách về, Linh chợt nhớ ra mình để quên điện thoại ở chỗ cuối cùng mà Linh ngồi. Linh không nhớ chỗ Linh ngồi cuối cùng ở đâu, nhưng lại nhớ chỗ Linh ngồi đầu tiên và nhớ rằng Linh đã đổi chỗ đúng k lần với những người bên cạnh.

Bây giờ Linh muốn tính số lượng chỗ ngồi mà cô ấy có thể đã ngồi cuối cùng trong suốt bữa tiệc.

Yêu cầu: Cho biết n và k , hãy giúp Linh xác định số chỗ ngồi mà cô ấy có thể đã ngồi cuối cùng trong suốt bữa tiệc.

Dữ liệu vào

Gồm một dòng chứa hai số nguyên n và k ($3 \leq n \leq 10^9, 0 \leq k \leq 10^9$).

Kết quả

Ghi ra một số nguyên duy nhất là số lượng chỗ ngồi mà Linh có thể đã ngồi trong suốt bữa tiệc.

Ví dụ

<code>party.inp</code>	<code>party.out</code>
5 2	3
3 3	3

Giải thích

Trong ví dụ đầu tiên, Linh có hai lần đổi chỗ với người bên cạnh. Do đó Linh có thể đã ngồi lại chỗ ban đầu, hoặc là ngồi một trong hai vị trí cách chỗ đầu tiên đúng một chỗ ngồi. Trong ví dụ thứ hai, Linh có thể đã ngồi ở tất cả các chỗ xung quanh bàn.

Bài B. Lịch trực cho y tá

File dữ liệu vào: Nurse.inp
File kết quả: Nurse.out
Hạn chế thời gian: 0.1 s
Hạn chế bộ nhớ: 512 MB

Người quản lý bệnh viện H muốn xếp ca trực đêm cho một y tá trong một giai đoạn gồm N ngày liên tiếp được đánh số từ 1 đến N . Do chính sách của bệnh viện cũng như đặc thù của công việc, y tá không thể trực liên tục tất cả các ngày từ 1 đến N mà cần phải có ngày nghỉ giữa chừng. Một giai đoạn trực được định nghĩa là 1 dãy các ngày liên tiếp nhau trong đó ngày nào y tá cũng phải trực. Chính sách của bệnh viện yêu cầu y tá chỉ được nghỉ đúng 1 ngày giữa 2 giai đoạn trực liên tiếp (có nghĩa hôm nay y tá nghỉ thì hôm sau bắt buộc phải trực), đồng thời số ngày của mỗi giai đoạn trực phải lớn hơn hoặc bằng K_1 và nhỏ hơn hoặc bằng K_2 . Lịch trực cho y tá là một cách chia N ngày ra thành các giai đoạn thỏa mãn chính sách đã nêu.

Yêu cầu: Hãy giúp người quản lý bệnh viện đếm xem có bao nhiêu cách xếp lịch trực cho y tá thỏa mãn yêu cầu đặt ra.

Dữ liệu vào

Một dòng duy nhất ghi 3 số nguyên dương cách nhau ít nhất một dấu cách $N, K_1, K_2 (N \leq 1000, K_1 < K_2 \leq 400)$

Kết quả

Ghi ra một số duy nhất là số dư trong phép chia số cách xếp lịch trực thỏa mãn yêu cầu đặt ra cho $10^9 + 7$.

Ví dụ

Nurse.inp	Nurse.out
6 2 3	4

Giải thích

Giải thích: Với bộ dữ liệu này, $N = 6, K_1 = 2, K_2 = 3$, có tất cả 4 cách xếp lịch trực thỏa mãn yêu cầu đặt ra sau đây:

Cách 1	trực	trực	nghỉ	trực	trực	trực
Cách 2	trực	trực	nghỉ	trực	trực	off
Cách 3	nghỉ	trực	trực	nghỉ	trực	trực
Cách 4	trực	trực	trực	nghỉ	trực	trực

Bài C. Trò chơi đồ thị

File dữ liệu vào: `graphgame.inp`
File kết quả: `graphgame.out`
Hạn chế thời gian: 0.2 giây
Hạn chế bộ nhớ: 512 Mb

Khi học lý thuyết đồ thị, Ninh và Giang cùng nhau chơi một trò chơi trên đồ thị. Đầu tiên Ninh có một đơn đồ thị vô hướng G có N đỉnh được đánh số từ 1 đến N . Tiếp theo Ninh tiếp tục ghi ra giấy danh sách các đỉnh kề của từng đỉnh. Sau đó Giang thay đổi danh sách các đỉnh kề của 1 hoặc 2 đỉnh bằng cách thay đổi một số đỉnh kề với chúng. Cụ thể, nếu một đỉnh ban đầu có X đỉnh kề, Giang sẽ ghi ra đủ X đỉnh kề mà một số đỉnh trong đó khác với danh sách đỉnh kề ban đầu.

Yêu cầu: Cho biết số lượng đỉnh mà Giang đã thay đổi và danh sách các đỉnh kề của từng đỉnh sau khi đã thay đổi, hãy giúp Ninh tìm ra các đỉnh kề bị Giang thay đổi.

Dữ liệu vào

- Dòng thứ nhất chứa một số nguyên duy nhất $P \in \{1, 2\}$ là số lượng đỉnh mà Giang đã thay đổi danh sách các đỉnh kề của chúng.
- Dòng thứ hai ghi duy nhất một số nguyên dương N ($3 \leq N \leq 10^5$) là số lượng đỉnh của đồ thị G .
- Dòng thứ i trong số N dòng tiếp theo mô tả danh sách kề của đỉnh i đã bị Giang thay đổi:
 - đầu tiên là một số nguyên dương $K_i \leq N - 1$, là số lượng đỉnh kề của đỉnh i ;
 - K_i số nguyên tiếp theo là các đỉnh kề của đỉnh i .
- Biết rằng $K_1 + K_2 + \dots + K_N \leq 4 \times 10^5$.

Các số trên cùng một dòng cách nhau bởi dấu cách.

Kết quả

- Nếu $P = 1$, ghi ra duy nhất một đỉnh mà Giang đã thay đổi.
- Nếu $P = 2$, ghi ra trên cùng dòng 2 đỉnh mà Giang đã thay đổi theo thứ tự tăng dần.

Dữ liệu đảm bảo luôn có lời giải và có duy nhất một lời giải.

Ví dụ

<code>graphgame.inp</code>	<code>graphgame.out</code>
2 7 4 7 3 2 4 2 6 1 4 7 6 4 1 4 1 3 5 6 2 4 1 3 7 5 1 1 3	1 6

Bài D. Nghịch thế

File dữ liệu vào: `inversion.inp`
File kết quả: `inversion.out`
Hạn chế thời gian: 0.1 giây
Hạn chế bộ nhớ: 256 MB

Số lượng nghịch thế của một dãy số a_1, a_2, \dots, a_n là số lượng cặp (a_i, a_j) thỏa mãn $i < j$ và $a_i > a_j$.

Yêu cầu: Cho số n và số lượng nghịch thế m , hãy tìm hoán vị nhỏ nhất của tập $\{1, 2, \dots, n\}$ sao cho số lượng nghịch thế của nó đúng bằng m .

Nhắc lại: một hoán vị a_1, a_2, \dots, a_n gọi là nhỏ hơn hoán vị b_1, b_2, \dots, b_n khi và chỉ khi tồn tại một số nguyên k sao cho $a_j = b_j$ với $1 \leq j < k$ và $a_k < b_k$.

Dữ liệu vào

Dòng đầu tiên ghi một số T ($T \leq 50$) là số lượng bộ test.

Dòng thứ i trong số T dòng tiếp theo chứa hai số nguyên n và m ($m \leq 10^{12}$).

Kết quả

Ghi ra T dòng, mỗi dòng một số q là giá trị duy nhất được tính theo hoán vị a_1, a_2, \dots, a_n tìm được tương ứng với dòng trong dữ liệu đầu vào. Số q là phần dư của phép chia công thức sau cho $10^9 + 7$:

$$a_1 + a_2 * 2^1 + a_3 * 2^2 + \dots + a_n * 2^{n-1}.$$

Nếu không tồn tại hoán vị thỏa mãn thì ghi ra -1.

Ví dụ

<code>inversion.inp</code>	<code>inversion.out</code>
2	58
5 9	673
7 3	

Giải thích

- $n = 5, m = 9$, hoán vị tìm được: $4\ 5\ 3\ 2\ 1 \rightarrow q = 58$;
- $n = 7, m = 3$, hoán vị tìm được: $1\ 2\ 3\ 4\ 7\ 6\ 5 \rightarrow q = 673$.

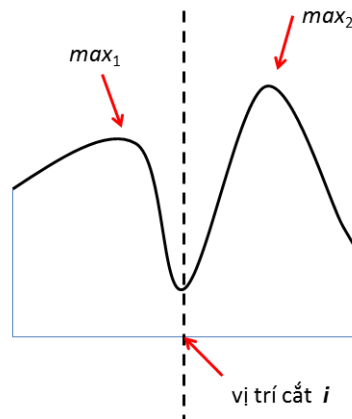
Hạn chế

- Subtask 1: $1 \leq n \leq 10$;
- Subtask 2: $1 \leq n \leq 1000$;
- Subtask 3: $1 \leq n \leq 10^5$;
- Subtask 4: $1 \leq n \leq 10^6$.

Bài E. Đo tín hiệu

File dữ liệu vào: `signal.inp`
File kết quả: `signal.out`
Hạn chế thời gian: 0.1 giây
Hạn chế bộ nhớ: 512 Mb

Một thiết bị cảm biến có nhiệm vụ thu nhận dữ liệu về các đối tượng trong 1 khu vực để truyền về cho trung tâm xử lý. Mỗi đối tượng sẽ được biểu diễn bởi 1 dãy số nguyên dương. Như vậy, cảm biến sẽ truyền các dãy số về cho trung tâm xử lý. Tuy nhiên, do các đối tượng ở gần nhau và có tín hiệu nhiễu nên một dãy số gửi về cho trung tâm có thể là dữ liệu của 2 đối tượng. Dãy a_1, \dots, a_n sẽ là dữ liệu của 2 đối tượng nếu có 1 vị trí i ($1 < i < n$) sao cho $\max_1 - a_i \geq b$ và $\max_2 - a_i \geq b$ với $\max_1 = \max\{a_1, \dots, a_{i-1}\}$, $\max_2 = \max\{a_{i+1}, \dots, a_n\}$, và b là hằng số cho trước (xem minh hoạ trong Hình 1). Khi phát hiện một dãy số a_1, \dots, a_n là dữ liệu của 2 đối tượng thì cần phải tiến hành cắt dãy số đó thành 2 dãy, mỗi dãy là dữ liệu của một đối tượng. Khi đó vị trí cắt sẽ là vị trí i sao cho $\max\{a_1, \dots, a_{i-1}\} - a_i + \max\{a_{i+1}, \dots, a_n\} - a_i$ đạt giá trị lớn nhất (giá trị đó gọi là độ đo cắt tín hiệu).



Hình 1: Minh hoạ dữ liệu biểu diễn 2 đối tượng

Ví dụ: với giá trị $b = 5$ thì dãy số 3, 5, 4, 7, 2, 5, 4, 6, 9, 8 là dữ liệu của 2 đối tượng vì tìm thấy vị trí $i = 5$ tại đó $\max\{3, 5, 4, 7\} - 2 \geq 5$ và $\max\{5, 4, 6, 9, 8\} - 2 \geq 5$ và vị trí $i = 5$ cũng chính là vị trí cắt.

Yêu cầu: cho trước giá trị n , b và dãy số nguyên dương a_1, \dots, a_n . Hãy lập trình kiểm tra xem dãy số a_1, \dots, a_n có phải là dữ liệu biểu diễn 2 đối tượng hay không và tính độ đo cắt tín hiệu nếu câu trả lời là có.

Dữ liệu vào

Dữ liệu đầu vào bao gồm các dòng sau:

- Dòng thứ nhất chứa 2 số nguyên dương n và b ($3 \leq n \leq 200000, 1 \leq b \leq 50$)
- Dòng thứ 2 chứa n tự nhiên a_1, \dots, a_n

Kết quả

Ghi ra độ đo cắt tín hiệu nếu dãy đầu vào là dữ liệu biểu diễn 2 đối tượng và giá trị -1 nếu ngược lại.

Ví dụ

signal.inp	signal.out
10 5 3 5 4 7 2 5 4 6 9 8	12

Bài F. Đánh giá hiệu quả làm việc

File dữ liệu vào: `checkpoint.inp`
File kết quả: `checkpoint.out`
Hạn chế thời gian: 0.1 giây
Hạn chế bộ nhớ: 512 Mb

Một công ty kinh doanh tiến hành đánh giá hiệu quả làm việc của nhân viên A (check point) dựa trên tình hình doanh thu và chi phí của họ theo thời gian trong 1 giai đoạn công tác. Một giai đoạn công tác bao gồm một chuỗi các ngày làm việc được đánh số $1, 2, \dots, N$. Ngày làm việc thứ i của nhân viên A được đặc trưng bởi 1 con số (doanh thu hoặc chi phí) a_i trong đó $a_i > 0$ có nghĩa ngày đó nhân viên thu được lượng tiền là a_i (doanh thu), $a_i < 0$ có nghĩa nhân viên cần chi tiêu một khoản $-a_i$ (chi phí). Hiệu quả làm việc của nhân viên A được đo bằng tổng lợi nhuận (tổng doanh thu - tổng chi phí) lớn nhất thu được trong số các chuỗi gồm một số liên tiếp các ngày mà trong đó có không quá K ngày nhân viên phải chi tiêu.

Ví dụ: giai đoạn kinh doanh gồm 9 ngày, trong đó lượng thu, chi của nhân viên được mô tả bởi dãy gồm 9 số: 100, -1, -2, -3, 60, -4, -5, 50, -10. Việc thu chi được thể hiện trong Bảng 1

Ngày	1	2	3	4	5	6	7	8	9
Thu	100				60			50	
Chi		1	2	3		4	5		10

Bảng 1: Tình hình thu và chi của nhân viên A

Yêu cầu: cho giá trị N , K và dãy a_1, \dots, a_N . Hãy tính hiệu quả công việc của nhân viên A.

Dữ liệu vào

Dữ liệu đầu vào bao gồm các dòng sau:

- Dòng thứ nhất chứa 2 số nguyên dương N và K ($1 \leq N \leq 10^5, 0 \leq K \leq 100$)
- Dòng thứ 2 chứa N số nguyên a_1, \dots, a_N

Kết quả

Ghi ra một số nguyên là hiệu quả làm việc của nhân viên A

Ví dụ

<code>checkpoint.inp</code>	<code>checkpoint.out</code>
9 2 100 -1 -2 -3 60 -4 -5 50 -10	101

Giải thích

Hiệu quả công việc là tổng lợi nhuận thu được trong giai đoạn 4 ngày: 5, 6, 7, 8 và bằng $60 - 4 - 5 + 50 = 101$.

Bài G. Xếp bò vào chuồng

File dữ liệu vào: `cow.inp`
File kết quả: `cow.out`
Hạn chế thời gian: 0.5s
Hạn chế bộ nhớ: 512 MB

Anh nông dân Giang mới xây dựng một cái chuồng nuôi bò rất dài mới, với N ($2 \leq N \leq 100,000$) gian. Các gian được xếp dọc theo một đường thẳng tại các vị trí x_1, \dots, x_N ($0 \leq x_i \leq 1,000,000,000$).

C ($2 \leq C \leq N$) con bò của anh ta không thích cách bố trí chuồng nuôi này và trở nên hung hăng với nhau khi đưa vào cùng một gian chuồng. Để ngăn chặn các con bò làm tổn thương nhau, Giang muốn xếp các con bò vào các gian chuồng, sao cho khoảng cách tối thiểu giữa hai con càng lớn càng tốt.

Yêu cầu: Hãy giúp Giang tìm cách bố trí để có khoảng cách tối thiểu là lớn nhất.

Dữ liệu vào

Dòng đầu chứa một số nguyên $t \leq 30$ là số lượng bộ test, tiếp theo mỗi bộ test có cấu trúc như sau:

- Dòng 1 chứa hai số nguyên cách nhau bởi dấu cách: N và C
- Dòng $2 \dots N + 1$: Dòng $i + 1$ chứa một số nguyên, x_i là vị trí của một ngăn chuồng.

Kết quả

Mỗi bộ test in ra trên một dòng một số nguyên là khoảng cách tối thiểu lớn nhất tìm được.

Ví dụ

cow.inp	cow.out
1	3
5 3	
1	
2	
8	
4	
9	

Giải thích

Giang có thể xếp 3 con bò của anh ta vào các ngăn chuồng ở các vị trí 1, 4 và 8, với khoảng cách tối thiểu thu được là 3.

Bài H. Chơi bi-a

File dữ liệu vào: `bia.inp`
File kết quả: `bia.out`
Hạn chế thời gian: 0.1 giây
Hạn chế bộ nhớ: 512 MB

Trên một bàn bi-a kích thước chiều ngang là a inch và kích thước chiều dọc là b , một viên bi được bắn đi từ chính giữa bàn. Sau khoảng thời gian $s > 0$ giây, viên bi trở lại vị trí mà nó được bắn đi sau khi va chạm m lần theo chiều dọc và n lần theo chiều ngang của bàn. Tìm góc bắn A ban đầu của viên bi (so với chiều cạnh chiều ngang) với giá trị nằm trong khoảng 0° và 90° và vận tốc bắn ban đầu của quả viên bi.

Giả thiết là va chạm với cạnh bàn là va chạm đàn hồi (không mất năng lượng), và vận tốc của quả viên bi so với hai cạnh của bàn là không đổi. Và cũng giả thiết rằng quả viên bi có bán kính là 0 và bàn bi-a không có lỗ.

Dữ liệu vào

Bao gồm nhiều dòng, mỗi dòng chứa 5 số nguyên không âm cách nhau bởi dấu cách. Năm số đó là a, b, s, m và n . Tất cả các số đều là số nguyên dương không lớn hơn 10 000.

Dòng cuối cùng chứa 5 số 0.

Kết quả

Đối với mỗi dòng trong dữ liệu vào ngoại trừ dòng cuối, ghi ra một dòng gồm có hai số thực (được làm tròn chính xác đến hai chữ số sau dấu phẩy) cách với nhau bằng dấu cách. Số đầu tiên là độ lớn của góc bắn A ban đầu và số thứ hai là vận tốc của viên bi theo đơn vị inch trên giây, theo mô tả phía trên.

Ví dụ

<code>bia.inp</code>	<code>bia.out</code>
100 100 1 1 1	45.00 141.42
200 100 5 3 4	33.69 144.22
201 132 48 1900 156	3.09 7967.81
0 0 0 0 0	

Bài I. Đặt cửa

File dữ liệu vào: `blade.inp`
File kết quả: `blade.out`
Hạn chế thời gian: 0.2s
Hạn chế bộ nhớ: 512 MB

Minh cần phải cắt giảm M mét gỗ. Đó là một công việc dễ dàng kể từ khi anh ta có một máy cưa mới tốt đến mức có thể chặt phá rừng như cháy rừng. Tuy nhiên, Mirko chỉ được phép cắt một hàng cây.

Máy cưa hoạt động như sau: Mirko thiết lập một tham số chiều cao H (tính bằng mét), và máy nâng một cây cưa khổng lồ đến chiều cao đó và cắt tất cả các phần cây trên hàng cao hơn H (tất nhiên, cây không cao hơn H mét vẫn còn nguyên vẹn). Minh sau đó lấy tất cả những phần cây bị cắt. Ví dụ: nếu hàng cây chứa cây có độ cao 20, 15, 10 và 17 mét, và Minh nâng cưa của mình lên 15 mét, phần cây còn lại sau khi cắt sẽ là 15, 15, 10 và 15 mét tương ứng, và Minh sẽ cắt được 5 mét từ cây đầu tiên và 2 mét từ cây thứ tư (tổng cộng 7 mét).

Minh thấy cần thiết bảo vệ sinh thái, vì vậy anh ta không muốn cắt gỗ nhiều hơn mức cần thiết. Đó là lý do tại sao anh ta muốn đặt cây cưa của mình càng cao càng tốt.

Yêu cầu: Hãy tìm cách giúp Minh tìm được chiều cao số nguyên tối đa của cưa mà vẫn cho phép anh ta lấy được ít nhất M mét gỗ.

Dữ liệu vào

Dòng đầu tiên chứa hai số nguyên dương cách nhau bởi dấu cách, N (số cây, $1 \leq N \leq 1000000$) và M (lượng gỗ tối thiểu Minh cần, $1 \leq M \leq 2000000000$).

Dòng thứ hai chứa N số nguyên dương cách nhau bởi dấu cách nhỏ hơn 1 000 000 000, là chiều cao của mỗi cây (tính bằng mét). Dữ liệu đảm bảo tổng của tất cả các chiều cao sẽ lớn hơn M , do đó Minh sẽ luôn luôn có thể có được số lượng gỗ yêu cầu.

Kết quả

Ghi trên một dòng duy nhất một số nguyên là chiều cao đặt cưa tìm được.

Ví dụ

<code>blade.inp</code>	<code>blade.out</code>
4 7 20 15 10 17	15
5 20 4 42 40 26 46	36

Bài J. Chèn phép toán

File dữ liệu vào: `operator.inp`
File kết quả: `operator.out`
Hạn chế thời gian: 0.1s
Hạn chế bộ nhớ: 256 MB

Cho tập N số nguyên $A = \{1, 2, 3, \dots, N\}$ và một số nguyên S .

Yêu cầu: Tìm cách điền mỗi phép toán ‘+’ hoặc ‘-’ vào giữa các cặp liên tiếp nhau của A hoặc trước số 1 sao cho kết quả của biểu thức thu được sau khi đã điền hết các phép toán là bằng đúng S .

Dữ liệu vào

Dòng đầu chứa một số nguyên $T \leq 20$ là số lượng bộ test, mỗi dòng tiếp theo mô tả một bộ test bao gồm hai số nguyên N và S ($1 \leq N \leq 500, |S| \leq 125250$).

Kết quả

Mỗi dòng trong số T dòng ghi ra 1 nếu có cách chèn cho bộ test tương ứng, còn ghi ra 0 nếu ngược lại.

Ví dụ

<code>operator.inp</code>	<code>operator.out</code>
2	1
9 5	0
5 6	

Giải thích

Có một cách chèn các phép toán đối với bộ test thứ nhất: $1-2+3-4+5-6+7-8+9=5$

Bài K. Cắt đá cẩm thạch

File dữ liệu vào: `marble.inp`
File kết quả: `marble.out`
Hạn chế thời gian: 0.5s
Hạn chế bộ nhớ: 512 MB

Phong là một nhà điêu khắc, ông có một tấm đá cẩm thạch hình chữ nhật kích thước $W \times H$. Ông ta muốn cắt tấm đá thành các miếng hình chữ nhật kích thước $W_1 \times H_1, W_2 \times H_2, \dots, W_N \times H_N$. Ông ta muốn cắt đến tối đa các mẫu kích thước có thể. Tấm đá có những vân đá cho nên không thể xoay khi sử dụng, có nghĩa là không thể cắt ra miếng $B \times A$ thay cho miếng $A \times B$ trừ khi $A = B$. Các miếng phải được cắt tại các điểm nguyên trên hàng cột và mỗi nhát cắt phải cắt đến hết hàng hoặc hết cột. Sau khi cắt sẽ còn lại những mẫu đá còn thừa bỏ đi, nghĩa là những mẫu đá không thể cắt thành miếng kích thước cho trước nào.

Yêu cầu: Hãy tìm cách cắt sao cho còn ít nhất diện tích đá thừa bỏ đi.

Dữ liệu vào

Dòng đầu tiên chứa hai số nguyên: W và H . Dòng thứ hai chứa một số nguyên N . N dòng tiếp theo mỗi dòng chứa hai số nguyên W_i và H_i .

Kết quả

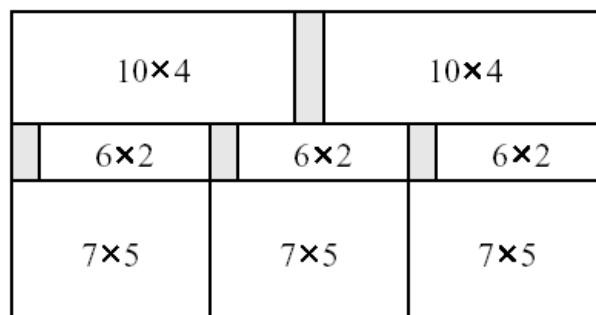
Kết quả ghi ra duy nhất một số nguyên là tổng diện tích nhỏ nhất các miếng thừa bỏ đi.

Ví dụ

marble.inp	marble.out
21 11 4 10 4 6 2 7 5 15 10	10

Giải thích

Hình dưới minh họa cách cắt các phiên đá trong ví dụ với diện tích thừa ít nhất tìm được là 10.



Hạn chế

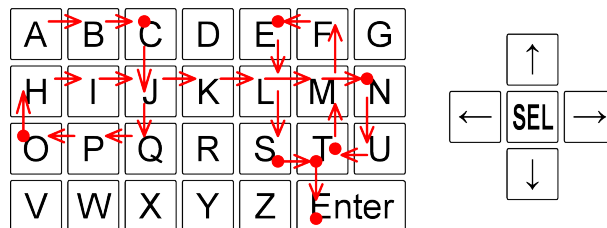
- $1 \leq W \leq 600, 1 \leq H \leq 600, 0 < N \leq 200, 1 \leq W_i \leq W$, and $1 \leq H_i \leq H$.
- Có 50% số test ứng với $W \leq 20, H \leq 20$ và $N \leq 5$.

Bài L. Gõ phím

File dữ liệu vào: `keyboarding.inp`
File kết quả: `keyboarding.out`
Hạn chế thời gian: 0.5 giây
Hạn chế bộ nhớ: 512 MB

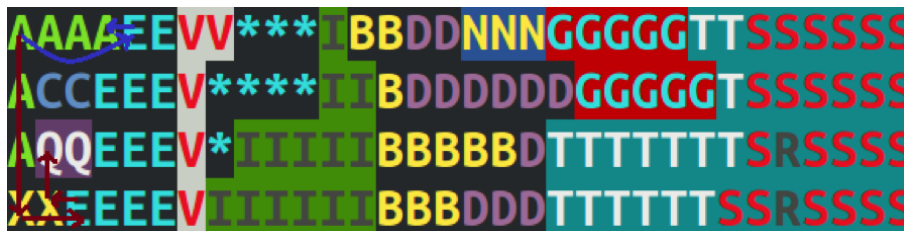
Đối với máy tính và đa số các thiết bị, các phím trên bàn phím được gõ trực tiếp để có được các ký tự mong muốn. Nhưng đối với các thiết bị nhỏ gọn như điện thoại, số lượng phím không đủ để thể hiện tất cả các ký tự, nên nhiều phương pháp đã được áp dụng. Một trong các giải pháp là dùng bàn phím ảo, có thể mô tả như sau:

Bàn phím ảo được hiển thị trên màn hình với đầy đủ các ký tự cần gõ, với một con trỏ phím di chuyển để chọn phím. Để di chuyển con trỏ, ta có 5 nút bấm, gồm 4 nút di chuyển theo hướng mũi tên, và nút select để xác nhận chọn ký tự. Ví dụ như hình vẽ:



Ta quy ước con trỏ sẽ bắt đầu từ phím góc trên trái của bàn phím ảo. Vậy để gõ chữ `CONTEST` ta phải thực hiện di chuyển và chọn ký tự như trên. Trong cách làm trên, cần 30 lượt bấm để gõ được chữ `CONTEST` (kết thúc bằng gõ phím `enter`). Chú ý là trên bàn phím ảo, mỗi ký tự có thể nằm trên nhiều ô vuông đơn vị liền nhau (vd như phím `Enter` trong hình).

Khi bấm một nút di chuyển, chưa chắc ô vuông liền kề đã là đích, mà là ô vuông của ký tự khác nó trên hướng đi. Nếu không thể chuyển sang phím khác ký tự theo hướng di chuyển thì con trỏ sẽ nằm nguyên mà không di chuyển. Ví dụ:



Yêu cầu: cho một đoạn văn bản là một chuỗi ký tự, hãy xác định số lần bấm ít nhất để gõ ra đoạn văn bản đó, bao gồm cả phím `Enter` ở cuối. Dữ liệu đảm bảo đoạn văn bản luôn có thể gõ được.

Dữ liệu vào

- Dòng đầu tiên gồm r và c là số dòng và số cột của bàn phím ($1 \leq r, c \leq 50$);
- R dòng tiếp theo là các ký tự của bàn phím, chỉ gồm các chữ cái hoa, chữ số, gạch ngang và `*` thể hiện cho ký tự `enter`. Mỗi ký tự có thể trải dài trên nhiều ô, nhưng phải đi liền nhau.
- Dòng cuối cùng là dòng chữ cần gõ, độ dài không quá 10000.

Kết quả

Ghi ra duy nhất một số là số lần bấm ít nhất để gõ ra đoạn văn bản cho trong input bao gồm cả phím `Enter` ở cuối.

Ví dụ

keyboarding.inp	keyboarding.out
4 7 ABCDEFGG HIJKLMN OPQRSTU VWXYZ** CONTEST	30
2 19 ABCDEFGHIJKLMNOPQZY X*****Y AZAZ	19
6 4 AXYB BBBB KLMB OPQB DEFB GHI* AB	7