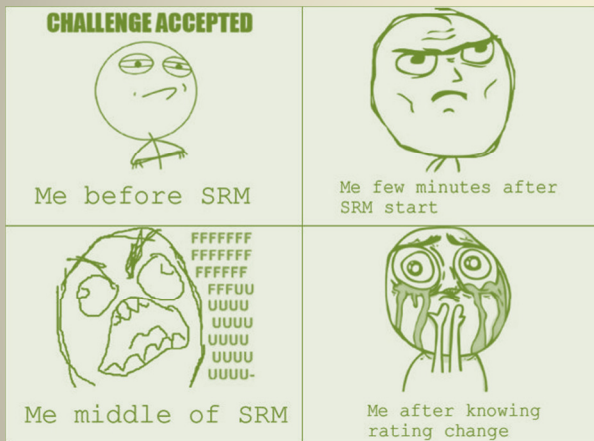


stdin – random.cpp - stdout

Luận Anh Hùng

Trận 05

Chuyên Tổng Hợp



[Thanh Trung Nguyen](#) : kiểm nỳ đi

Thanh Trung Nguyen : nếu chú ko học nó sẽ chữi là

Thanh Trung Nguyen : học hành thế à, sau này ai nuôi bà :-w

Thanh Trung Nguyen : thế là phải học =))

mr_invincible khuyên **huuthanh**

(Nguồn : fb RR)

1. LÁI XE XUYÊN SA MẠC

Giả sử xe đua của bạn chuẩn bị băng qua sa mạc. Phía trước là bao hiểm nguy và khó khăn. Mạng sống của bạn dựa vào chiếc jeep có đủ đủ xăng không. Nhưng cần đủ một lượng xăng tối thiểu là bao nhiêu để bạn băng được qua phía bên kia của sa mạc ?

Các sự kiện sau đây có thể xảy ra trong hành trình của bạn:

Fuel consumption n Có nghĩa là xe của bạn cần n lít nhiên liệu để đi 100km. n là số nguyên trong khoảng [1..30]. Giá trị n này có thể thay đổi trong suốt hành trình, ví dụ khi n tăng khi xuống dốc, n giảm khi lên dốc.

Leak Có nghĩa là thùng xăng của bạn va vào một vật cứng và bị rò rỉ. Thùng xăng sẽ rò rỉ 1 lít nhiên liệu trên 1 km. Nếu có k vết rò rỉ thì sẽ mất k lít nhiên liệu khi đi 1 km. Ch

Tuy nhiên không chỉ toàn những sự kiện đen đủi như vậy, có những sự kiện sau làm tăng khả năng sống sót trên sa mạc của bạn.

Gas station cho phép bạn nạp đầy nhiên liệu.

Mechanic sửa thùng xăng bị rò rỉ.

Và cuối cùng

Goal Bạn đã thành công đến đích!

INPUT

Input có nhiều test. Mỗi test có tối đa 50 sự kiện. Mỗi sự kiện được mô tả bằng một số nguyên, khoảng cách (theo đơn vị km, tính từ điểm khởi đầu) – khi sự kiện bắt đầu xảy ra, sau đó là kiểu sự kiện.

Trong mỗi test, sự kiện đầu tiên là '0 Fuel consumption n', và sự kiện cuối cùng là 'd Goal' (d là khoảng cách còn lại đến đích)

Các sự kiện được sắp xếp theo khoảng cách từ vị trí xảy ra sự kiện đến vị trí khởi đầu (theo thứ tự không giảm). Có thể tại cùng một vị trí xảy ra đồng thời nhiều sự kiện, khi đó hãy xử lý theo thứ tự xuất hiện các sự kiện

Dòng '0 Fuel consumption 0' đánh dấu kết thúc input và không phải xử lý.



OUTPUT

Với mỗi test, in ra dòng chứa dung tích thùng xăng bé nhất có thể (theo lít, có 3 chữ số sau dấu chấm thập phân) để đảm bảo đến đích thành công.

| Sample Input | Sample Output |
|-------------------------|---------------|
| 0 Fuel consumption 10 | 10.000 |
| 100 Goal | 35.000 |
| 0 Fuel consumption 5 | 81.000 |
| 100 Fuel consumption 30 | |
| 200 Goal | |
| 0 Fuel consumption 20 | |
| 10 Leak | |
| 25 Leak | |
| 25 Fuel consumption 30 | |
| 50 Gas station | |
| 70 Mechanic | |
| 100 Leak | |
| 120 Goal | |
| 0 Fuel consumption 0 | |

2. TÌM SỐ

Cho trước số nguyên i . Hãy xác định chữ số thứ i trong chuỗi hợp thành $S_1S_2...S_k$. Mỗi S_k là các số nguyên liên tiếp từ 1 đến k , viết liên tiếp nhau. Ví dụ 80 chữ số đầu tiên của chuỗi hợp thành như sau:

11212312341234512345612345671234567812345678912345678910123456789101112345678910

INPUT

Dòng đầu là số lượng test t ($1 \leq t \leq 25$), t dòng tiếp theo là các test, mỗi test là một số nguyên i ($1 \leq i \leq 2147483647$)

OUTPUT

Ứng với mỗi test là một kết quả tương ứng.

| Sample Input | Sample Output |
|--------------|---------------|
| 2 | 2 |
| 8 | 2 |
| 3 | |

3. MÊ CUNG SỐ

Không ai biết ai là người đầu tiên tạo ra mê cung, như tất cả các loại mê cung có thể tìm thấy tại hầu hết mọi nơi và một số được tạo ra từ một thời gian lâu dài về trước. Mê cung nổi tiếng nhất có thể là đang nằm trong Crete, được Daedalus vẽ và thiết kế dành cho vua Minos, một nơi mà người ta tin rằng Minotaur (quái thú đầu bò thân người) đã ở đó, ăn thịt người, đến khi bị Theseus tiêu diệt. Hình sau mô tả một kiểu mê cung.

Liệu bạn có thể tìm đường đưa bạn từ đầu đến cuối mê cung? Bạn chỉ được phép di chuyển dọc hoặc ngang, bạn không được di chuyển chéo. Đường đi phải đi qua các ô theo quy luật : 1; 1,2; 1,2,3; 1,2,3,4; Trên đường đi có thể có sự chuyển hướng bất kỳ.

Vấn đề bạn phải giải là xác định vị trí điểm đi vào và con đường đưa bạn ra đến điểm kết thúc. Điểm xuất phát luôn nằm tại một ô hàng đầu tiên của mê cung (với số 1!) và điểm kết thúc luôn là hàng cuối cùng trong mê cung.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 6 | 5 | 2 | 1 | 1 | 2 | 3 | 2 | 1 | 4 |
| 1 | 2 | 6 | 3 | 2 | 1 | 1 | 3 | 4 | 5 | 6 |
| 1 | 2 | 3 | 2 | 1 | 3 | 2 | 5 | 6 | 4 | 2 |
| 2 | 3 | 1 | 2 | 2 | 3 | 3 | 4 | 5 | 2 | 1 |
| 3 | 4 | 2 | 3 | 4 | 5 | 3 | 2 | 1 | 4 | 2 |
| 4 | 3 | 4 | 4 | 5 | 6 | 4 | 3 | 2 | 5 | 3 |
| 5 | 4 | 2 | 1 | 2 | 3 | 4 | 4 | 3 | 6 | 4 |
| 6 | 5 | 3 | 2 | 3 | 4 | 5 | 5 | 4 | 1 | 1 |
| 1 | 6 | 4 | 3 | 3 | 5 | 6 | 6 | 1 | 2 | 3 |
| 2 | 1 | 5 | 1 | 6 | 6 | 1 | 2 | 2 | 3 | 4 |

INPUT

Input bắt đầu với một số nguyên dương chỉ ra số lượng trường hợp, mỗi trường hợp sẽ được diễn giải ngay sau đó. Dòng này được nối tiếp bằng một dòng trống, và có một dòng trống nữa ở giữa hai trường hợp khác nhau.

Dòng đầu tiên của mỗi trường hợp là hai số nguyên N, M chỉ số hàng và số cột của từng mê cung. Mỗi N dòng liên tục chứa M số, ngăn cách bởi một khoảng trống. Giá trị của mỗi ô luôn lớn hơn hoặc bằng 1.

OUTPUT

Đối với mỗi trường hợp, output phải tuân theo chỉ dẫn sau đây. Giữa hai trường hợp liên nhau, phải có một dòng trống giữa chúng.

Mỗi trường hợp, có hai dòng, dòng đầu tiên là tọa độ (hàng và cột) của ô xuất phát, và dòng thứ hai chứa tọa độ của điểm kết thúc. Nếu có nhiều cách giải khác nhau, in ra một cách mà có điểm xuất phát nhỏ nhất, nếu vẫn có nhiều con đường, chỉ ra điểm kết thúc nhỏ nhất.

| Sample Input | Sample Output |
|--|---------------|
| 10 11 1 6 5 2 1 1 2 3 2 1 4 1 2 6 3 2 1 1 3 4 5 6 1 2 3 2 1 3 2 5 6 4 2 2 3 1 2 2 3 3 4 5 2 1 3 4 2 3 4 5 3 2 1 4 2 | 1 6 10 3 |

| | |
|-----------------------|--|
| 4 3 4 4 5 6 4 3 2 5 3 | |
| 5 4 2 1 2 3 4 4 3 6 4 | |
| 6 5 3 2 3 4 5 5 4 1 1 | |
| 1 6 4 3 5 5 6 6 1 2 3 | |
| 2 1 5 1 6 6 1 2 2 3 4 | |

4. MATRIOSHKAS

Vladimir làm matrioshkas trong nhiều năm, matrioshkas những con búp bê lồng vào nhau – là sản phẩm thủ công đặc sắc của Nga. Matrioshka là một con búp bê có thể được tách thành hai nửa, bên trong chứa một con búp bê. Sau đó, con búp bê này có thể được tách ra để bên trong lại chứa một con búp bê khác. Điều này có thể được lặp đi lặp lại nhiều lần, cho đến khi không thể tách ra con búp bê cuối cùng bé nhất.



Gần đây, Vladimir nhận ra rằng có thể tổng quát ý tưởng của con búp bê lồng nhau thành món đồ chơi lồng nhau. Thật vậy, ông đã thiết kế những đồ chơi có chứa nhiều đồ chơi hơn. Một trong những món đồ chơi có thể được tách thành hai nửa và bên trong có thể có nhiều món đồ chơi bên trong. Đó là tính năng mới mà Vladimir muốn giới thiệu trong dòng sản phẩm đồ chơi của mình.

Vladimir đã phát triển một ký pháp để mô tả các đồ chơi lồng nhau. Một món đồ chơi được biểu diễn bằng một số nguyên dương, thể hiện kích thước của nó. Chính xác hơn: nếu khi mở món đồ chơi m , chúng ta thấy những đồ chơi n_1, n_2, \dots, n_r , và $n_1 + n_2 + \dots + n_r < m$. Khi đó, chúng ta nói rằng đồ chơi m trực tiếp chứa đồ chơi n_1, n_2, \dots, n_r . Các đồ chơi nằm bên trong các đồ chơi n_1, n_2, \dots, n_r không được coi là trực tiếp nằm trong đồ chơi m .

Matrioshka tổng quát được thể hiện bằng một chuỗi các số nguyên dương có dạng:

$$a_1 a_2 \dots a_N$$

trong đó đồ chơi k được biểu diễn trong chuỗi với hai số nguyên $-k$ và k , với số âm xuất hiện trong chuỗi trước số dương – thể hiện 2 mảnh của một món đồ chơi k .

Ví dụ, chuỗi $-9 -7 -2 2 -3 -2 -1 1 2 3 7 9$ biểu diễn một đồ chơi matrioshka tổng quát, cụ thể là có 6 món đồ chơi hai món đồ chơi 2; 1 3, 7 và 9 mỗi món có 1 đồ chơi. Lưu ý rằng: Đồ chơi 9 chứa trực tiếp đồ chơi 7, đồ chơi 7 chứa trực tiếp một món đồ chơi 2 và món 3; món đồ chơi ba chứa món đồ chơi 2, món đồ chơi 2 này chứa món đồ chơi 1. Chú ý rằng 2 mảnh -2 đầu tiên và 2 cuối cùng không phải hai mảnh của cùng một đồ chơi.

Mặt khác, các chuỗi sau đây không mô tả matrioshkas tổng quát:

-9 -7 -2 2 -3 -1 -2 2 1 3 7 9

bởi vì đồ chơi 2 lớn hơn so với đồ chơi 1 và không thể được nhét vào bên trong.

-9 -7 -2 2 -3 -2 -1 1 2 3 7 -2 2 9

bởi vì 7 và 2 không thể cùng nhét vào bên trong đồ chơi 9.

-9 -7 -2 2 -3 -1 -2 3 2 1 7 9

bởi vì có trục trặc trong việc nhét đồ vào đồ chơi 3.

Vấn đề của bạn là viết một chương trình để giúp đỡ Vladimir phân biệt một thiết kế tốt một thiết kế xấu.

INPUT

Các tập tin đầu vào có chứa nhiều test, mỗi test nằm trên một dòng riêng biệt. Mỗi test là một chuỗi các số nguyên khác không, có giá trị tuyệt đối nhỏ hơn 10^7 .

OUTPUT

Với mỗi test, in ra một dòng theo quy tắc sau

- **:-) Matrioshka!** nếu thiết kế mô tả một matrioshka tổng quát.
- **:-(Try again.** Trong trường hợp khác, câu trả lời nên có dạng

| Sample Input | Sample Output |
|-------------------------------|-----------------|
| -9 -7 -2 2 -3 -2 -1 1 2 3 7 9 | :-) Matrioshka! |
| -9 -7 -2 2 -3 -1 -2 2 1 3 7 9 | :-(Try again. |
| -9 -7 -2 2 -3 -1 -2 3 2 1 7 9 | :-(Try again. |
| -100 -50 -6 6 50 100 | :-) Matrioshka! |
| -100 -50 -6 6 45 100 | :-(Try again. |
| -10 -5 -2 2 5 -4 -3 3 4 10 | :-) Matrioshka! |
| -9 -5 -2 2 5 -4 -3 3 4 9 | :-(Try again. |

Về thôi !