



الدمية الميكانيكية

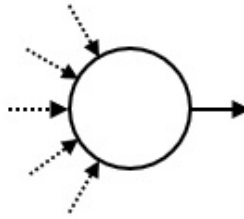
الدمية الميكانيكية هي لعبة تقوم بتكرار سلسلة معينة من الحركات بشكل أوتوماتيكي، وفي اليابان تم تصنيع العديد من الدمى الالكترونية منذ العصور القديمة

حركات الدمية الميكانيكية يتم التحكم بها عن طريق دائرة والتي تتألف من أجهزة. الجهاز يتصل بمجموعة من الأنابيب. كل جهاز له بعض المداخل (من الممكن ان لا يوجد مداخل له)، ومخرج واحد أو مخرجين. كل جهاز يمكن ان يكون له عدد اعتباطي من المداخل. كل انبوب يصل بين مخرج لجهاز ومدخل لجهاز آخر ممكن أن يكون نفس النوع أو غير نوع. انبوب واحد تماماً موصول إلى كل مدخل وواحد تماماً موصول إلى كل مخرج.

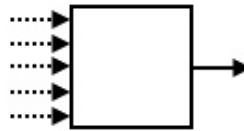
لتوصيف كيفية قيام الدمية بالحركات، لنفرض أن كرة موضوعة في أحد الأجهزة. تنتقل الكرة عبر الدارة. في كل خطوة من التنقل، تغادر الكرة جهازاً من خلال أحد مخارج، وتنتقل عبر الأنبوب المرتبط بهذا المخرج وتدخل إلى الجهاز في النهاية الأخرى للأنبوب. تستمر الكرة بالانتقال إلى اللانهاية.

يوجد ثلاث أنواع من الأجهزة: الأصل، القادح والمفتاح. يوجد واحد تماماً من نوع الأصل كما يوجد M قادح ويوجد S مفتاح (يمكن ل S أن تكون صفر). يجب عليك حساب قيمة S . كل جهاز له رقم تسلسلي فريد.

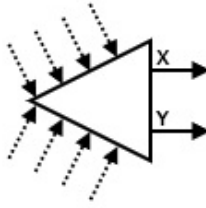
الأصل هو الجهاز الذي تكون فيه الكرة في البداية، له فقط مخرج واحد ورقمه التسلسلي هو 0.



القادح يسبب قيام الدمية بأحد أنواع الحركات، كلما دخلت كرة إليه. لكل قادح مخرج واحد فقط. الرقم التسلسلي للقادح هو من 1 وحتى M



كل مفتاح له مخرجان، يدعيان 'X' و 'Y'. حالة المفتاح هي إما 'X' أو 'Y'. بعد أن تدخل الكرة إلى المفتاح تغادره من خلال المخرج المحدد بالحالة الحالية للمفتاح. بعد ذلك يقوم المفتاح بتغيير حالته الحالية إلى الحالة المعاكسة. في البداية تكون حالة كل المفاتيح هي 'X'. الأرقام التسلسلية للمفاتيح تكون من 1- وحتى S .



سيتم إعطاؤك عدد القوادح M ، كما سيتم إعطاؤك سلسلة A طولها N . كل من عناصرها هو رقم تسلسلي لأحد القوادح. يمكن لكل قادح أن يظهر عدة مرات (وربما ولا مرة) ضمن A . مهمتك هي إنشاء دارة تحقق الشروط التالية:

- يجب على الكرة أن تعود إلى الأصل بعد عدد ما من الخطوات.
- عندما تعود الكرة أول مرة إلى الأصل يجب أن تكون حالة جميع المفاتيح هي 'X'.
- تعود الكرة إلى الأصل بعد أن تدخل إلى القوادح تماماً N مرة والأرقام التسلسلية المتتالية للقوادح التي تم الدخول إليها يجب أن يكون A_0, A_1, \dots, A_{N-1} .
- ليكن العدد الكلي لتغير حالات المفاتيح التي سببها عبور الكرات خلالها قبل أن تعود الكرة أول مرة إلى الأصل، يجب أن تكون قيمة P أن لا تتجاوز 20 000 000.

بنفس الوقت أنت لا تريد أن تستخدم الكثير من المفاتيح.

تفاصيل البرمجة

يجب عليك برمجة الإجراء التالي

```
create_circuit(int M, int[] A)
```

- M : عدد القوادح.
- A : مصفوفة طولها N ، تعطي الأرقام التسلسلية المتتالية للقوادح التي يجب على الكرة أن تدخلها..
- يتم استدعاء هذا الإجراء مرة واحدة تماماً.
- لاحظ أن قيمة N هي طول المصفوفة A ، ويمكن الحصول عليها كما هو موضح في ملاحظات البرمجة.

يجب على برنامجك أن يستدعي الإجراء التالي لإعطاء الجواب

```
answer(int[] C, int[] X, int[] Y)
```

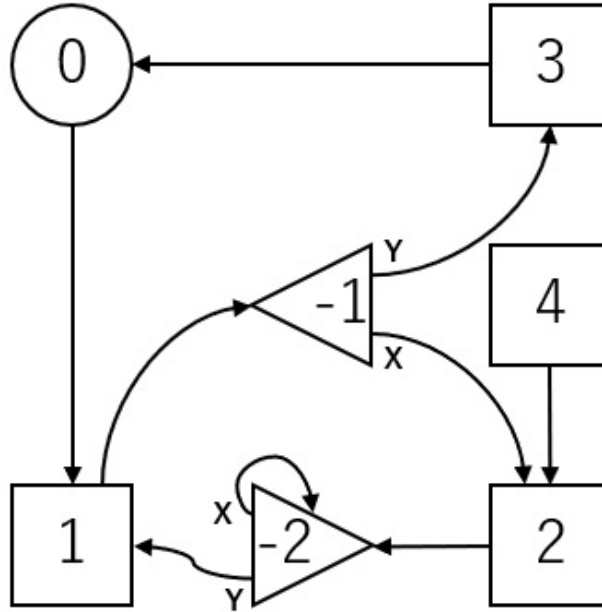
- C : مصفوفة طولها $M + 1$. بحيث مخرج الجهاز i ($0 \leq i \leq M$) موصول بالجهاز $C[i]$.
- X, Y : مصفوفتان من نفس الطول، طول هاتين المصفوفتين S هو عدد المفاتيح، من أجل المفتاح $j -$ ($1 \leq j \leq S$)، مخرج 'X' خاصته متصل مع الجهاز $X[j - 1]$ ومخرج 'Y' خاصته متصل بالجهاز $Y[j - 1]$.
- كل عنصر من C و X و Y يجب أن يكون عددين صحيحاً بين $-S$ و M ، متضمنهما.
- S يجب أن تكون على الأكثر 400 000.
- يجب استدعاء هذا الإجراء مرة واحدة تماماً.
- الدارة الممثلة بـ C و X و Y يجب أن تتوافق مع الشروط المحددة بنص المسألة.

إذا تم مخالفة أحد الشروط السابقة ستكون نتيجة تقييم برنامج هي **Wrong Answer**. وإلا، ستكون نتيجة تقييم

برنامج هي **Accepted** وسيتم تحديد علامتك بناءً على قيمة S (انظر المسائل الجزئية).

مثال

لتكن $M = 4$, $N = 4$ و $A = [1, 2, 1, 3]$. سيقوم المقيم بطلب `create_circuit(4, [1, 2, 1, 3])`



الشكل أعلاه يعرض دائرة، والتي يمكن تمثيلها من خلال الاستدعاء `answer([1, -1, -2, 0, 2], [2, 1], [-2, 3])`. الأعداد في الشكل هي الأرقام التسلسلية للأجهزة.

تم استخدام مفاتيح ، وهكذا $S = 2$.

في البداية تكون حالة المفتاح -1 و -2 كلاهما 'X'.

تنتقل الكرة كما يلي:

$$0 \rightarrow 1 \rightarrow -1 \xrightarrow{X} 2 \rightarrow -2 \xrightarrow{X} -2 \xrightarrow{Y} 1 \rightarrow -1 \xrightarrow{Y} 3 \rightarrow 0$$

- عندما تدخل الكرة لأول مرة إلى المفتاح -1 ، تكون حالته هي 'X'. وهكذا تنتقل الكرة نحو القادح 2. بعدها تتغير حالة المفتاح -1 إلى 'Y'.
- عندما تدخل الكرة إلى المفتاح -1 مرة ثانية، تكون حالته هي 'Y'. وهكذا تنتقل الكرة إلى القادح 3. بعدها تتغير حالة المفتاح -1 إلى 'X'.

عندما تعود الكرة إلى الأصل، تكون قد عبرت عبر القوادح 1, 2, 1, 3 كما أن حالة المفتاحين -1 و -2 هي في كلاهما 'X'. قيمة P هي 4. لذلك هذه الدائرة تتوافق مع الشروط.

الملف `sample-01-in.txt` في الحزمة المرفقة المضغوطة يتوافق مع هذا المثال. كما أن هناك أمثلة أخرى على الدخل ضمن الحزمة.

القيود

- $1 \leq M \leq 100\,000$
- $1 \leq N \leq 200\,000$
- $1 \leq A_k \leq M$ ($0 \leq k \leq N - 1$)

المسائل الجزئية

العلامة والقيود لكل حالة اختبار هي كالتالي:

1. (2 نقطتان) من أجل i ($1 \leq i \leq M$)، العدد i يظهر مرة واحدة على الأكثر في السلسلة A_0, A_1, \dots, A_{N-1} .
2. (4 نقاط) من أجل i ($1 \leq i \leq M$)، العدد i يظهر مرتين على الأكثر في السلسلة A_0, A_1, \dots, A_{N-1} .
3. (10 نقاط) من أجل i ($1 \leq i \leq M$)، العدد i يظهر 4 مرات على الأكثر في السلسلة A_0, A_1, \dots, A_{N-1} .
4. (10 نقاط) $N = 16$
5. (18 نقاط) $M = 1$
6. (56 نقاط) لا يوجد قيود إضافية

من أجل كل حالة اختبار إذا تم تقييم برنامجك بأنه **Accepted** سيتم حساب نتيجتك بحسب قيمة S :

- إذا كان $S \leq N + \log_2 N$ ، تنال العلامة الكاملة على حالة الاختبار.
- من أجل كل حالة اختبار في المسألتين الجزئيتين 5 و 6، إذا كان $N + \log_2 N < S \leq 2N$ ، تحصل على علامة جزئية، العلامة على حالة الاختبار هي $0.5 + 0.4 \times \left(\frac{2N - S}{N - \log_2 N} \right)^2$ ، مضروبة بالعلامة المخصصة للمسألة الجزئية.
- غير ذلك ستكون علامتك صفر 0.

لاحظ ان علامتك في كل مسألة جزئية هي أصغر علامة من بين علامات حالات الاختبار في هذه المسألة الجزئية.

مقيم الاختبار

The sample grader reads the input from the standard input in the following format

- 1 line: $N \ M$
- 2 line: $A_{N-1} \dots A_1 \ A_0$

The sample grader produces three outputs

First, the sample grader outputs your answer to a file named out.txt in the following format

- 1 line: S
- $2 + i$ ($0 \leq i \leq M$): $C[i]$
- $2 + M + j$ ($1 \leq j \leq S$): $X[j - 1] \ Y[j - 1]$

Second, the sample grader simulates the moves of the ball. It outputs the serial numbers of the devices the ball entered in order to a file named `log.txt`

Third, the sample grader prints the evaluation of your answer to the standard output

If your program is judged as **Accepted**, the sample grader prints S and P in the following format Accepted: S P

If your program is judged as **Wrong Answer**, it prints Wrong Answer: MSG. The meaning of MSG is as follows

answered not exactly once: The procedure answer is called not exactly once

wrong array length: The length of C is not $M + 1$, or the lengths of X and Y are different

400 000 over 400000 switches: S is larger than

wrong serial number: There is an element of C , X , or Y which is smaller than $-S$ or larger than M

over 20000000 inversions: The ball doesn't return to the origin within 20 000 000 state changes of the switches

state 'Y': There is a switch whose state is 'Y' when the ball first returns to the origin

wrong motion: The triggers which cause motions are different from the sequence A

Note that the sample grader might not create `out.txt` and/or `log.txt` when your program is judged as Wrong Answer