Механічна лялька

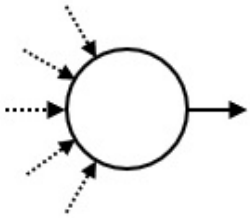
Механічна лялька - це лялька, яка автоматично повторює певну послідовність рухів. У Японії з давніх часів створювали багато механічних ляльок.

Рухи механічної ляльки котролюються **схемою**, що складається з **пристроїв**. Ці пристрої з'єднані між собою трубами. Кожен пристрій має один або два **виходи**, та може мати скільки завгодно (можливо нуль) **входів**. Кожна труба з'єднує вихід пристрою з входом цього або іншого пристрою. Рівно одна труба приєднана до кожного входу та рівно одна труба приєднана до кожного виходу.

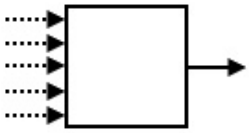
Щоб описати рухи ляльки, розглянемо **кульку**, яка розташована на одному з пристроїв. Кулька рухається по схемі. На кожному кроці подорожі кулька залишає пристрій через один з його виходів, проходить по трубі, приєднаній до виходу, і потряпляє у пристрій на іншому кінці труби.

Є три типи пристроїв: **старт**, **тригер** та **перемикач**. Є тільки один старт, M тригерів та S перемикачів (S може бути рівне нулю). Вам необхідно знайти S. Кожен пристрій має унікальний серійний номер.

Старт - це пристрій в якому спочатку знаходиться кулька. Він має один вихід. Його серійний номер 0.

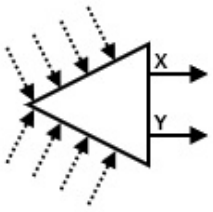


Тригер змушує ляльку робити певний рух кожного разу, коли кулька входить до нього. Кожен тригер має один вихід. Серійні номери усіх тригерів від 1 до M.



Кожен перемикач має два виходи, які називаються 'X' та 'Y'. Перемикач може знаходитись в **стані** 'X' або 'Y'. Після того, як кулька потрапляє до перемикача, вона виходить з перемикача через вихід, заданий поточним станом перемикача. Після цього перемикач змінює свій стан на протилежний. Спочатку стан кожного

перемикача 'X'. Серійні номери усіх перемикачів від -1 до $-S$.



Вам дано кількість тригерів M. Вам також дано послідовність A довжини N, кожен з елементів якої є серійним номером тригера. Кожен тригер може з'явитись декілька (можливо нуль) разів у послідовності A. Ваше завдання полягає у створенні схеми, що задовольняє наступним умовам:

- Після кількох кроків кулька повертається до старту.
- Коли кулька повертається до старту, стан кожного перемикача 'X'.
- Кулька перший раз повертається до старту, коли відвідає рівно N тригерів. Серійні номери цих тригерів в порядку потрапляння до них кульки є $A_0, A_1, \ldots, A_{N-1}$.
- Нехай P - загальна кількість змін станів усіх перемикачів, викликаних кулькою, перш ніж кулька повернеться до старту. Значення P не перевищує $20\,000\,000$.

До того ж, ви не бажаєте використовувати надто багато перемикачів.

Деталі реалізації

Ви повинні реалізувати наступну процедуру.

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

- `M`: кількість тригерів.
- `A`: масив довжини N, який містить серійні номери тригерів, які має відвідати кулька в порядку потрапляння.
- Ця процедура викликається рівно один раз.
- Зверніть увагу, що N це довжина масиву `A`, і його значення можна отримати способом зазначеним в Зауваженнях до реалізації.

Ваша програма повинна викликати таку процедуру, щоб надати відповідь.

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

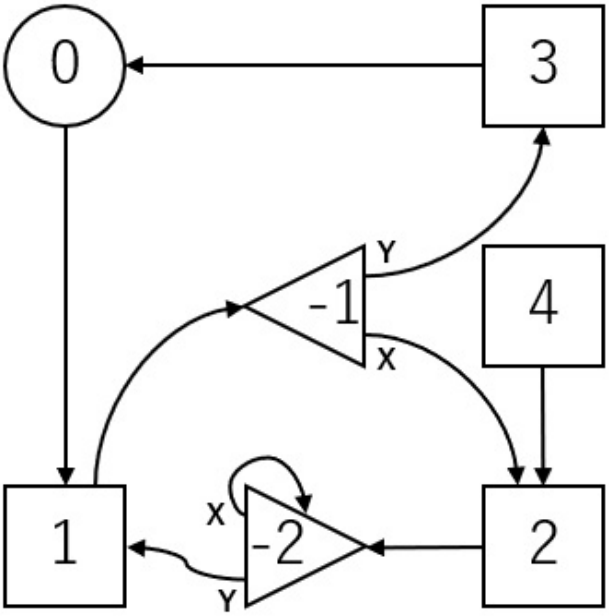
- `C`: масив довжини $M+1$. Вихід пристрою i $(0 \leq i \leq M)$ підключено до пристрою `C[i]`.
- `X`, `Y`: масиви однакової довжини. Довжина S цих масивів - кількість перемикачів. Для перемикача $-j$ $(1 \leq j \leq S)$, його вихід 'X' підключено до

пристрою `X[j - 1]` і вихід 'Y' підключено до пристрою `Y[j - 1]`.
- Кожен елемент `C`, `X`, і `Y` має бути цілим числом в межах від $-S$ до M, включно.
- S має бути не більше $400\,000$.
- Цю процедуру потрібно викликати один раз.
- Схема, представлена символами `C`, `X`, та `Y` повинна задовольняти умовам задачі.

Якщо деякі з перелічених вище умов не виконуються, ваша програма отримує **`Wrong Answer`**. В іншому випадку ваша програма отримує **`Accepted`** і ви отримаєте оцінку, що залежить від S (див. підзадачі).

Приклад

Нехай $M = 4$, $N = 4$, і $A = [1, 2, 1, 3]$. Модуль перевірки викликає `create_circuit(4, [1, 2, 1, 3])`.



Наведений вище малюнок показує схему, яка описується викликом `answer([1, -1, -2, 0, 2], [2, -2], [3, 1])`. Числа на малюнку - це серійні номери пристроїв.

Використовується два перемикачі. Отже, $S = 2$.

Напочатку стани перемикачів -1 та -2 обидва 'X'.

Кулька рухається таким чином:

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

- Коли кулька спочатку потрапляє на перемикач -1, його стан 'X'. Тому кулька рухається до тригера 2. Стан перемикача -1 змінюється на 'Y'.
- Коли кулька вдруге потрапляє на перемикач -1 його стан 'Y'. Тому кулька

рухається до тригера 3. Стан перемикача -1 змінюється на 'X'.

Кулька перший раз повертається до старту, відвідавши тригери $1, 2, 1, 3$. Статуси перемикачів -1 і -2 - 'X'. Значення P дорівнює 4. Тому ця схема задовольняє умовам.

Файл `sample-01-in.txt`, що знаходиться в zip-архіві, відповідає цьому прикладу. Інші приклади також знаходяться в цьому архіві.

Обмеження

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

Підзадачі

Оцінка та обмеження для кожного тесту наступні:

1. (2 бала) Для усіх i $(1 \le i \le M)$, число i зустрічається не більше одного разу у послідовності $A_0, A_1, \ldots, A_{N-1}$.
2. (4 бала) Для усіх i $(1 \le i \le M)$, число i зустрічається максимально двічі у послідовності $A_0, A_1, \ldots, A_{N-1}$.
3. (10 балів) Для усіх i $(1 \le i \le M)$, число i зустрічається не більше 4 разів у послідовності $A_0, A_1, \ldots, A_{N-1}$.
4. (10 балів) $N = 16$
5. (18 балів) $M = 1$
6. (56 балів) Без додаткових обмежень.

Для кожного тесту, якщо ваша програма отримає **`Accepted`**, то ваша оцінка обчислюється в залежності від значення S:

- Якщо $S \le N + \log_2 N$, то ви отримаєте за тест повний бал.
- Для кожного тесту з підзадач 5 та 6, якщо $N + \log_2 N < S \le 2N$, то ви отримаєте частковий бал. Оцінка за тест буде $0.5 + 0.4 \times \left(\dfrac{2N - S}{N - \log_2 N}\right)^2$, помножене на оцінку підзадачі.
- Інакше, ви отримаєте 0.

Зверніть увагу, що ваша оцінка за кожну підзадачу буде обчислюватись як мінімум з оцінок для кожного теста з цієї підзадачі.

Модуль перевірки з прикладу

Модуль перевірки з прикладу читає вхідні дані зі стандартного вводу у наступному

форматі.

- рядок 1: M N
- рядок 2: A_0 A_1 $\ldots$ A_{N-1}

Модуль перевірки з прикладу видає результати у три способи.

По перше, модуль перевірки з прикладу виводить вашу відповідь у файл `out.txt` у такому форматі.

- рядок 1: S
- рядок $2 + i$ ($0 \le i \le M$): `C[i]`
- рядок $2 + M + j$ ($1 \le j \le S$): `X[j - 1] Y[j - 1]`

По друге, модуль перевірки з прикладу симулює рухи кульки. Він виводить номери пристроїв, в які вона послідовно потрапляла у файл `log.txt`.

По третє, модуль перевірки з прикладу виводить оцінку вашої відповіді у стандартний вивід.

- Якщо ваша програма отримує **`Accepted`**, модуль перевірки з прикладу виводить S та P у наступному форматі `Accepted: S P`.

- Якщо ваша програма отримує **`Wrong Answer`**, він виводить `Wrong Answer: MSG`. `MSG` означає наступне:

    - `answered not exactly once`: Процедуру `answer` було викликано не рівно один раз.
    - `wrong array length`: Довжина `C` не є $M + 1$, або довжини `X` та `Y` є різними.
    - `over 400000 switches`: S більше ніж $400\,000$.
    - `wrong serial number`: Існує елемент `C`, `X`, або `Y` який менше ніж $-S$ або більше ніж M.
    - `over 20000000 inversions`: Кулька не повертається на старт після $20\,000\,000$ змінів стану перемикачів.
    - `state 'Y'`: Є перемикач у стані 'Y' коли кулька вперше повертається на старт.
    - `wrong motion`: Тригери, що спричинили рухи, відмінні від послідовності A.

Зверніть увагу, що модуль перевірки з прикладу може не створити файл `out.txt` та/або файл `log.txt` якщо ваша програма отримає `Wrong Answer`.