Механична кукла

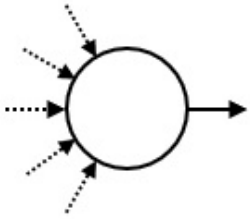
Механичните кукли автоматично повтарят предварително зададена последователност от движения. В Япония има традиции в изработката на механични кукли, датиращи от древни времена.

Движенията на всяка механична кукла се управляват от логическа **схема**, която се състои от **устройства**. Устройствата са свързани с тръбичка. Всяко устройство има произволен брой **входове** (допустими са нула входа) и един или два **изхода**. Всяка тръбичка свързва изход на устройство с вход на устройство (допустимо е устройство да се свързва посредством тръбичка със себе си). Точно една тръбичка е закачена към всеки вход и всеки изход.

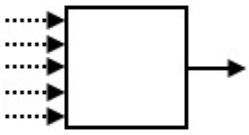
За да опишем как се управлява движението на куклата, нека си представим **топче**, което пускаме в едно от устройствата. Топчето започва да се движи по схемата. На всеки етап от пътуването си топчето напуска устройството, в което се намира, през един от изходите, търкулва се по съответната тръбичката и попада в устройството, свързано с другия ѝ край.

Съществуват три вида устройства: **начално**, **активатор** и **превключвач**. Във схемата има точно едно начално устройство, M активатора и S превключвача (S може да е равно на нула). Вашата програма трябва да вземе решение колко превключвача ще се използват. Всяко от устройствата има уникален сериен номер.

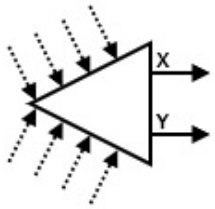
За да се стартира процеса, топчето се пуска в началното устройство. Началното устройство има точно един изход и неговият сериен номер е равен на 0.



Всеки активатор кара куклата да изпълни определено движение, когато топчето попадне в него. Всеки активатор има точно един изход. Серийните номера на активаторите са от 1 до M.



Всеки превключвач има точно два изхода, означени с 'X' и 'Y'. Всеки превключвач се намира в едно от две **състояния** - 'X' или 'Y'. Когато топчето попадне в превключвач, то го напуска през изхода, който е определен от текущото състояние на устройството.. След напускане на топчето състоянието на превключвача се сменя. В началото всички превключвачи се намират в състояние 'X'. Серийните номера на превключвачите са от (-1) до $(-S)$.



Известен е броят активатори M. Зададена е и последователност Aот серийни номера на активатори с дължина N. Всеки активатор може да се среща нула или повече пъти в последователността A. Вашата задача е да създадете схема от устройства, която удовлетворява следните условия:

- Топчето се връща в началното устройство след определен брой придвижвания между устройствата.
- При завръщане на топчето в началното устройство, всички превключвачи се намират в състояние 'X'.
- Топчето се завръща за първи път в началното устройство след като е преминало през точно N активатора. Серийните номера на посетените активатори (в реда на посещаването) образуват последователността $A_0, A_1, \ldots, A_{N-1}$.
- Нека P е общия брой промени на състояния на превключвачи, извършени по време на движение на топчето, преди завръщането му за пръв път в началното устройство. Стойността на P не трябва да надвишава $20\,000\,000$.

В същото време вашата схема не трябва да използва твърде много превключвачи.

Детайли за реализацията

Трябва да реализирате следната процедура:

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

- `M`: броя на активаторите.
- `A`: масив с дължина N, задаващ последователните серийни номера на активатори, които топчето трябва да посети.
- Процедурата се извиква точно един път.
- Как да получите дължината N на масива `A` е описано в таблицата на страница "Бележки".

Вашата програма трябва да извика следната процедура, за да подаде отговора към грейдъра:

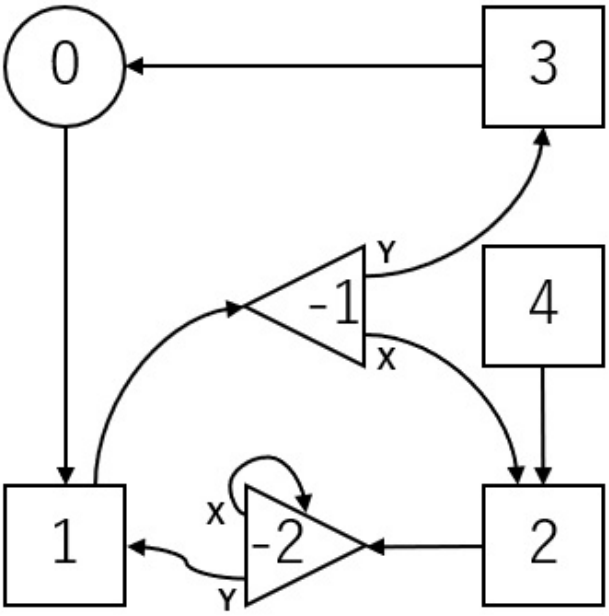
```
answer(int[] C, int[] X, int[] Y)
```

- C: масив с дължина $M + 1$. Изходът на устройство i ($0 \leq i \leq M$) е свързан с вход на устройство `C[i]`.
- X, Y: масиви с еднаква дължина. Дължината на тези масиви S е броя на използваните превключвачи. За превключвач с номер $(-j)$ ($1 \leq j \leq S$) имаме - изходът му 'X' е свързан с устройство `X[j - 1]`, а изходът му 'Y' е свързан с устройство `Y[j - 1]`.
- Всеки елемент на масивите C, X и Y трябва да е цяло число между $-S$ и M включително.
- S трябва да бъде максимум $400\,000$.
- Процедурата трябва да бъде извикана само един път.
- Схемата представена в масивите C, X и Y трябва да удовлетворява условията, на задачата.

Ако някое от условията не е изпълнено, вашата програма ще получи от оценяващата система **`Wrong Answer`**. В противен случай програмата ще получи **`Accepted`** и резултатът ще бъде изчислен според стойността на S (за подробности вижте в Подзадачи).

Пример

Нека $M = 4$, $N = 4$, and $A = [1, 2, 1, 3]$. Грейдърът извиква `create_circuit(4, [1, 2, 1, 3])`.



Горната фигура показва схема, която е описана с извикване на `answer([1, -1,`

`-2, 0, 2], [2, -2], [3, 1])`. Числата на фигурата показват серийните номера на устройствата.

Използвани са два превключвача, т.е. $S = 2$.

В началото състоянията на превключвачите (-1) и (-2) са 'X'.

Топчето се движи по следния маршрут:

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

- Когато топчето за първи път влезе в превключвач (-1), той е в състояние 'X'. Поради тази причина топчето отива в активатор 2, след което състоянието на превключвача се сменя на 'Y'.
- Когато топчето повторно попадне в превключвач (-1), състоянието на превключвача вече е 'Y' и топчето се придвижва в активатор 3. След това състоянието на превключвач -1 се сменя на 'X'.

Топчето се връща за първи път в началното устройство, след като е обходило активатори $1, 2, 1, 3$. Състоянията на превключвачи -1 и -2 са 'X'. Стойността на P е 4. Построената схема отговаря на всички условия.

Файлът `sample-01-in.txt` в прикачения архив съответства на описания пример. В архива има и други примерни тестове.

Ограничения

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

Подзадачи

1. (2 точки) За всяко i $(1 \leq i \leq M)$, числото i се среща най-много един път в редицата $A_0, A_1, \ldots, A_{N-1}$.
2. (4 точки) За всяко i $(1 \leq i \leq M)$, числото i се среща най-много два пъти в редицата $A_0, A_1, \ldots, A_{N-1}$.
3. (10 точки) За всяко i $(1 \leq i \leq M)$, числото i е среща най-много 4 пъти в редицата $A_0, A_1, \ldots, A_{N-1}$.
4. (10 точки) $N = 16$
5. (18 точки) $M = 1$
6. (56 точки) Няма допълнителни ограничения

За всеки тест, ако вашата програма е оценена като **`Accepted`**, точките, които получавате, се определят спрямо стойността на S:

- Ако $S \leq N + \log_2 N$, получавате всички точки за теста.
- За всеки тест в Подзадачи 5 и 6, ако $N + \log_2 N < S \leq 2N$, получавате частичен резултат. Стойността му е $0.5 + 0.4 \times \left(\frac{2N - S}{N - \log_2 N} \right)^2$, умножено по точките, определени за подзадачата.
- В останалите случаи получавате 0 точки.

Забележка: Общите точки за всяка подзадача, са равни на минималните точки измежду всички тестове, включени в подзадачата.

Примерен грейдър

Примерният грейдър чете входните данни в следния формат:

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

Примерният грейдър извежда три неща:

Първо - вашият отговор се записва във файл с име `out.txt`, форматът е следния:

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

Второ - грейдърът симулира движението на топчето. Серийниите номера на устройствата, които топчето посещава, се записват поред във файл с име `log.txt`.

Трето - грейдърът извежда на стандартния изход оценката на вашия отговор:

- Ако отговорът е оценен като **`Accepted`**, грейдърът извежда S и P във формат `Accepted: S P`.
- Ако отговорът е оценен като **`Wrong Answer`**, грейдърът извежда `Wrong Answer: MSG`. Съобщението `MSG` може да е едно от следните:
  - `answered not exactly once`: Процедурата `answer` не е извикана точно веднъж.
  - `wrong array length`: Дължината на масива `C` не е равна на $M + 1$ или дължините на масивите `X` и `Y` са различни.
  - `over 400000 switches`: S е повече от $400\,000$.
  - `wrong serial number`: Има елемент в масивите `C`, `X` или `Y` който е по-малък от $(-S)$ или по-голям от M.
  - `over 20000000 inversions`: Топчето не е се е върнало в началното устройство преди $20\,000\,000$ промени на състоянията на превключвачите.
  - `state 'Y'`: Има превключвач в състояние 'Y' след завръщане на топчето в началното устройство.

- `wrong motion`: Придвижването на топчето през активаторите не следва зададената в A последователност.

Забележка: Примерният грейдър може да не създаде файлове `out.txt` и/или `log.txt`, когато вашият отговор е оценен като `Wrong Answer`.