Mechaniczna lalka

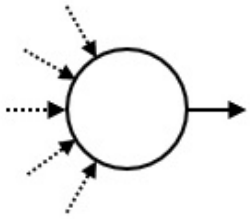
Mechaniczna lalka, to taka lalka, która powtarza konkretną sekwencję ruchów. W Japonii od czasów starożytnych wyprodukowano wiele mechanicznych lalek.

Ruchy mechanicznej lalki są sterowane za pomocą **obwodu** składającego się z **urządzeń**. Urządzenia są połączone rurkami. Każde urządzenie ma jedno lub dwa **wyjścia** oraz pewną liczbę (być może zero) **wejść**. Każda rurka łączy wyjście z pewnego urządzenia z wejściem do tego samego lub innego urządzenia. Do każdego wejścia i do każdego wyjścia jest podłączona dokładnie jedna rurka.

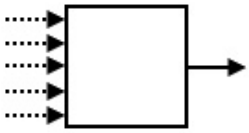
Aby zrozumieć, jak lalka działa, wyobraźmy sobie **piłeczkę** umieszczoną w jednym z urządzeń. Piłeczka zaczyna podróż przez obwód. W każdym kroku opuszcza urządzenie przez jedno z wyjść, przemieszcza się wzdłuż rurki i wpada do urządzenia znajdującego się na drugim końcu rurki.

Mamy trzy typy urządzeń: **start**, **lejek** i **przełącznik**. Start jest tylko jeden, lejków jest M, a przełączników S (S może być równe zero). Wartość S musisz sam dobrać. Każde urządzenie ma też unikatowy numer.

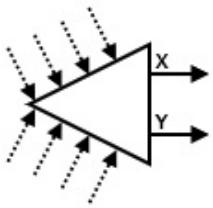
Start obwodu jest urządzeniem, które zawiera piłeczkę na początku. Ma on jedno wyjście. Numer tego urządzenia to 0.



Lejek powoduje, że lalka wykonuje jeden specyficzny ruch, kiedy tylko piłka wpadnie do niego. Każdy lejek ma jedno wyjście. Lejki są ponumerowane of 1 do M.



Każdy przełącznik ma dwa wyjścia nazwane 'X' oraz 'Y'. **Stan** przełącznika, to 'X' albo 'Y'. Piłeczka, która wpadła do przełącznika, opuszcza go przez wyjście zgodne z jego aktualnym stanem. Zaraz potem stan urządzenia zmienia się na przeciwny. Początkowo stany wszystkich przełączników to 'X'. Przełączniki mają ujemne numery od -1 do $-S$.



Masz daną liczbę lejków M. Dany jest również ciąg A długości N, którego elementami są numery lejków. Każdy lejek może się w tym ciągu pojawić wiele razy (być może zero). Twoim zadaniem jest konstrukcja obwodu spełniającego następujące warunki:

- Po pewnej liczbie kroków piłeczka wraca do startu.
- Kiedy piłeczka powraca po raz pierwszy do startu, wszystkie przełączniki są w stanie 'X'.
- Piłeczka wraca po raz pierwszy do startu po odwiedzeniu dokładnie N lejków. **Kolejne** numery odwiedzonych lejków to $A_0, A_1, \ldots, A_{N-1}$.
- Niech P będzie całkowitą liczbą zmian stanów wszystkich przełączników w czasie podróży piłeczki do pierwszego powrotu do startu. Wartość P nie może przekroczyć $20\,000\,000$.

Jednocześnie nie chcesz używać zbyt wielu przełączników.

Szczegóły implementacyjne

Powinieneś zaimplementować procedurę o nagłówku

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

gdzie:

- `M`: liczba lejków.
- `A`: tablica długości N zawierająca numery lejków, które ma odwiedzić piłeczka, w kolejności, w jakiej powinny być odwiedzone.
- Procedura ta będzie wywołana dokładnie raz
- Zwróć uwagę na to, że liczba N jest długością tablicy `A` i może być uzyskana w sposób podany w *Uwagach Implementacyjnych*.

Twój program powinien wywołać następującą procedurę ustalającą obwód

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

gdzie:

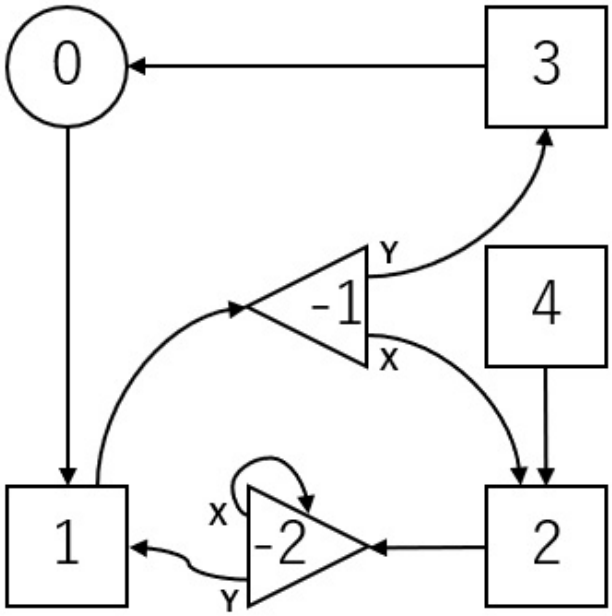
- `C` jest tablicą długości $M + 1$. Wyjście z urządzenia i $(0 \leq i \leq M)$ ma być podłączone do wejścia urządzenia `C[i]`.
- `X`, `Y` są tablicami tej samej długości. Długość tych tablic S to liczba przełączników.

Dla każdego j ($1 \leq j \leq S$), przełącznik $-j$ ma wyjście 'X' podłączone do urządzenia `X[j - 1]`, a wyjście 'Y' do urządzenia `Y[j - 1]`.
- Każdy element `C`, `X` oraz `Y` musi być liczbą całkowitą między $-S$ a M włącznie.
- S nie może przekroczyć $400\,000$.
- Procedura ta musi zostać wywołana dokładnie raz.
- Obwód zdefiniowany przez tablice `C`, `X`, oraz `Y` musi spełniać warunki z treści zadania.

Jeśli którykolwiek z tych warunków nie będzie spełniony, program dostanie werdykt **`Wrong Answer`**. W przeciwnym razie program dostanie werdykt **`Accepted`**, a jego wynik będzie zależał od S (zob. Podzadania).

Przykład

Niech $M = 4$, $N = 4$, $A = [1, 2, 1, 3]$. Sprawdzaczka wywołuje `create_circuit(4, [1, 2, 1, 3])`.



Powyższy diagram przedstawia obwód, który odpowiada wywołaniu `answer([1, -1, -2, 0, 2], [2, -2], [3, 1])`. Liczby w węzłach są numerami urządzeń.

Ponieważ użyte zostały dwa przełączniki, $S = 2$.

Początkowo oba przełączniki -1 i -2 są w stanie 'X'.

Piłeczka porusza się tak:

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

- Kiedy piłeczka po raz pierwszy trafia do przełącznika -1, jest on w stanie 'X'. Zatem piłeczka kierowana jest do lejka 2. Stan przełącznika -1 zmienia się na 'Y'.
- Kiedy piłeczka trafia do przełącznika -1 po raz drugi, jest on w stanie 'Y', więc

piłeczka trafia do lejka 3. Stan przełącznika -1 wraca do 'X'.

Piłeczka wraca do startu po raz pierwszy, odwiedzając po drodze lejki $1, 2, 1, 3$. Stany obu przełączników, -1 i -2 są równe 'X'. Wartością P jest 4. Zatem obwód spełnia narzucone warunki.

Plik `sample-01-in.txt` w zzipowanym załączniku odpowiada temu przykładowi. W pakiecie znajdują się też inne dane wejściowe.

Ograniczenia

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

Podzadania

Punkty i ograniczenia będą następujące dla każdego testu:

1. (2 punkty) Dla każdego i $(1 \leq i \leq M)$, liczba i występuje co najwyżej raz w każdym ciągu $A_0, A_1, \ldots, A_{N-1}$.
2. (4 punkty) Dla każdego i $(1 \leq i \leq M)$ liczba i występuje co najwyżej dwukrotnie w ciągu $A_0, A_1, \ldots, A_{N-1}$.
3. (10 punktów) Dla każdego i $(1 \leq i \leq M)$ liczba i występuje co najwyżej 4 razy w ciągu $A_0, A_1, \ldots, A_{N-1}$.
4. (10 punktów) $N = 16$
5. (18 punktów) $M = 1$
6. (56 punktów) Brak dodatkowych ograniczeń

Dla każdego testu, który dostał werdykt **`Accepted`**, Twój wynik zależy od S:

- Jeśli $S \leq N + \log_2 N$, to dostaniesz komplet punktów za ten test,
- Dla każdego testu w podzadaniach 5 i 6, jeśli $N + \log_2 N < S \leq 2N$, to dostaniesz niepełną liczbę punktów. Wzór na nią to $0.5 + 0.4 \times \left(\frac{2N - S}{N - \log_2, N}\right)^2$, przemnożone przez liczbę punktów z danego podzadania.
- W przeciwnym razie ocena to 0.

Zwróć uwagę na to, że dla każdego podzadania wynikiem jest minimum z ocen testów wchodzących w skład tego podzadania.

Przykładowa sprawdzaczka

Przykładowa sprawdzaczka czyta dane ze standardowego wejścia w następującym formacie.

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

i wypisuje wyniki na trzy wyjścia.

Najpierw przykładowa sprawdzaczka wypisuje Twoją odpowiedź do pliku `out.txt` w następującym formacie

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

Następnie przykładowa sprawdzaczka symuluje poruszanie się piłeczki. Kolejne numery odwiedzanych urządzeń wypisuje do pliku `log.txt`.

W końcu przykładowa sprawdzaczka wypisuje ocenę Twojej odpowiedzi na standardowe wyjście.

- Jeśli Twój program ma werdykt **`Accepted`**, to przykładowa sprawdzaczka wypisuje S oraz P w następującym formacie: `Accepted: S P`.

- Jeśli Twój program ma werdykt **`Wrong Answer`**, to wypisuje `Wrong Answer: MSG`. Znaczenie komunikatu `MSG` jest następujące:

  - `answered not exactly once`: Procedura `answer` nie została wywołana dokładnie raz.
  - `wrong array length`: Długość tablicy `C` nie jest równa $M + 1$, lub długości `X` i `Y` różnią się.
  - `over 400000 switches`: S przekracza $400\,000$.
  - `wrong serial number`: Istnieje element `C`, `X` lub `Y`, którego wartość jest mniejsza od $-S$ lub większa od M.
  - `over 20000000 inversions`: Piłeczka nie wraca do startu przez pierwsze $20\,000\,000$ zmian stanu przełączników.
  - `state 'Y'`: Istnieje przełącznik, którego stan to 'Y' w momencie pierwszego powrotu piłeczki do startu.
  - `wrong motion`: lejki, które napotyka piłeczka nie odpowiadają ciągowi A.

Zwróć uwagę na to, że przykładowa sprawdzaczka może nie stworzyć plików `out.txt` i/lub `log.txt` w przypadku werdyktu `Wrong Answer`.