



Mechanical Doll

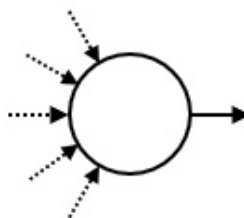
Mehanička lutka je igračka popularna u Japanu od pradavnih vremena. Te lutke su iznutra donekle slične fliperima koji su jednako dugo popularni u Hrvatskoj i šire.

Slično kao i kod flipera, kroz mehaničku lutku putuje **kuglica**, ona prolazi kroz **sklop** koji se sastoji od **elemenata**. Elementi sklopa su povezani tubama kroz koje kuglica prolazi. Svaki element ima nula, jedan ili više **ulaza** i jedan ili dva **izlaza**. Svaka tuba povezuje izlaz nekog elementa s ulazom nekog, možda istog, elementa.

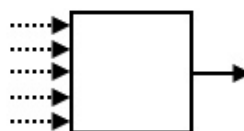
Dakle kuglica kroz lutku putuje tako da izađe kroz izlaz elementa, prođe kroz jednu tubu i dođe na ulaz nekog (možda istog) elementa. Za razliku od flipera na kuglicu ovdje ne utječe gravitacija, te ona stoga uvijek nastavlja svoj put unedogled.

Postoje tri vrste elemenata: **izvor**, **zvono**, i **klackalica**. U ovom zadatku će vaš program morati izgraditi neki od sklop takvih elemenata. Sklop mora imati točno jedan izvor, točno M zvona, i S klackalica (S može biti nula). Broj M će vam biti zadan, a sami možete odabrati broj klackalica S . Svaki element sklopa ima jedinstven serijski broj.

Izvor je element iz kojeg kuglica kreće na put kada se lutka upali. Izvor ima točno jedan izlaz. Serijski broj izvora je 0.

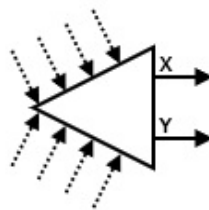


Zvono je element koji proizvodi zvuk kada kuglica uđe u njega. Svako zvono ima točno jedan izlaz. Serijski brojevi zvona su od 1 do M .



Klackalica ima točno dva izlaza označena s 'X' i 'Y'. Svaka klackalica može biti u jednom od dva stanja -- ili je okrenuta prema 'X' ili je okrenuta prema 'Y' (također kažemo da je klackalica *u stanju 'X'* odnosno *u stanju 'Y'*). Kada kuglica uđe u klackalicu, ona izlazi iz klackalice onim izlazom prema kojem je klackalica trenutno

okrenuta. Svaki put odmah nakon što kuglica izađe iz klackalice, njeno stanje se mijenja. Kada se lutka upali svaka je klackalica u stanju 'X'. Serijski brojevi klackalica su od -1 do $-S$.



Zadan vam je broj željeni zvona M . Također vam je zadan niz A duljine N , a svaki element tog niza je serijski broj jednog od zvona. Pojedino zvono se može pojavljivati nula, jedan ili više puta u nizu brojeva A (svako zvono se pojavljuje točno jednom u sklopu).

Vaš je zadatak da izgradite skop koji ima sljedeća svojstva:

- Kada se lutka upali, kuglica se iz početne pozicije nakon nekog broja koraka opet vraća u izvor.
- U tom trenutku (kada se kuglica prvi put vrati nazad u izvor), svaka klackalica mora biti u stanju 'X'.
- Putujući kroz sklop od početne pozicije do prvog povratka u izvor, kuglica udari zvono točno N puta. Pritom, serijski brojevi *redom* udarenih zvona moraju biti A_0, A_1, \dots, A_{N-1} .
- Neka P ukupan broj promjena stanja klackalica prije prvog povratka kuglice u izvor. Broj P ne smije biti veći od 20 000 000.

Potrebno je pronaći slop koji zadovoljava sva svojstva bez da koristi puno klackalica.

Implementacijski detalji

Morate implementirati sljedeću funkciju:

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

- M : broj zvona.
- A : niz brojeva duljine N koji sadrži serijske brojeve zvona koje je potrebno redom udariti.
- Ova funkcija se poziva točno jednom za svaki test podatak.
- Primijetite da vrijednost N odgovara duljini polja i i može se odrediti shodno dokumentu o implementacijskim uputama.

Vaš program mora pozvati sljedeću funkciju kada je pronašao traženi sklop:

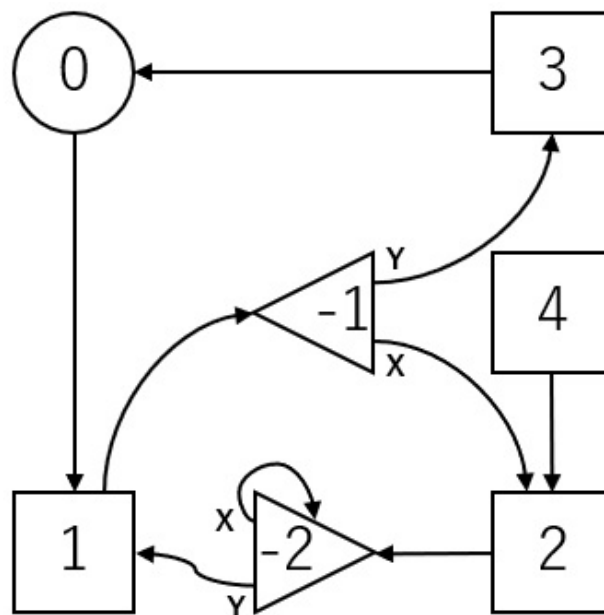
```
answer(int[] C, int[] X, int[] Y)
```

- C: niz duljine $M + 1$. Izlaz elementa i ($0 \leq i \leq M$) je povezan na ulaz elementa $C[i]$.
- X, Y: nizovi jednake duljine. Duljina S ovih nizova je broj klackalica u vašem sklopu. Za klackalicu $-j$ ($1 \leq j \leq S$), njezin izlaz 'X' je povezan na ulaz elementa $X[j - 1]$, a izlaz 'Y' na ulaz elementa $Y[j - 1]$.
- Svaki član nizova C, X, i Y mora biti cijeli broj između $-S$ i M , uključivo.
- S smije biti najviše 400 000.
- Ovu funkciju morate pozvati točno jednom.
- Sklop definiran nizovima C, X, i Y mora zadovoljavati svojstva opisana u tekstu zadatka.

Ako neki od gornjih uvjeta nije zadovoljen, vaš program će biti ocijenjen s rezultatom **Wrong Answer**. Inače, vaš program će biti ocijenjen s rezultatom **Accepted** te će broj bodova ovisiti o broju klackalica S (vidi poglavlje Podzadaci).

Primjer

Neka je $M = 4$, $N = 4$, i $A = [1, 2, 1, 3]$. Program *grader* zove `create_circuit(4, [1, 2, 1, 3])`.



Slika prikazuje sklop opisan pozivom `answer([1, -1, -2, 0, 2], [2, -2], [3, 1])`. Brojevi unutar elemenata su upravo njihovi serijski brojevi.

Ovaj sklop koristi dvije klackalice, dakle $S = 2$.

Na početku su klackalice -1 i -2 u stanju 'X'.

Kuglica putuje na sljedeći način:

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

- Kada kuglica prvi put uđe u klackalicu -1 , njezino stanje je 'X'. Dakle, kuglica putuje u zvono 2, a stanje klackalice -1 se mijenja u 'Y'.
- Kada kuglica drugi put uđe u klackalicu -1 , njezino stanje je 'Y'. Dakle, kuglica putuje u zvono 3, a stanje klackalice -1 se mijenja u 'X'.

Kuglica se vraća u izvor, nakon što je redom prošla kroz zvona 1, 2, 1, 3. Obje klackalice -1 and -2 su stanju 'X'. Vrijednost od P je 4. Dakle, ovaj skop zadovoljava sve zahtjeve.

Ulazna datoteka `sample-01-in.txt` u privitku odgovara ovom primjeru. Privitak sadrži i druge primjere ulaznih datoteka.

Ograničenja

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

Podzadaci

Bodovi i ograničenja za svaki test podataka su sljedeći:

1. (2 boda) Za svaki i ($1 \leq i \leq M$), broj i se pojavljuje najviše jednom u nizu A_0, A_1, \dots, A_{N-1} .
2. (4 boda) Za svaki i ($1 \leq i \leq M$), broj i se pojavljuje najviše dvaput u nizu A_0, A_1, \dots, A_{N-1} .
3. (10 bodova) Za svaki i ($1 \leq i \leq M$), broj i se pojavljuje najviše 4 puta u nizu A_0, A_1, \dots, A_{N-1} .
4. (10 bodova) $N = 16$
5. (18 bodova) $M = 1$
6. (56 bodova) Nema dodatnih ograničenja.

Za svaki test podatak, ako je vaš program ocijenjen s rezultatom **Accepted**, broj bodova ovisi o broju S :

- Ako je $S \leq N + \log_2 N$, dobivate sve bodove za test podatak.
- Za svaki test podatak u podzadacima 5 i 6, ako je $N + \log_2 N < S \leq 2N$, dobivate parcijalne bodove. Bodovi za test podatak su jednaki umnošku koeficijenta $0.5 + 0.4 \times \left(\frac{2N - S}{N - \log_2 N} \right)^2$ i ukupnog broja bodova za podzadatak.
- Inače, broj bodova je 0.

Broj bodova za svaki podzadatak je najmanji osvojeni broj bodova među svim test podacima u tom podzadatku.

Ogledni *grader*

Ogledni *grader* čita ulazne podatke sa standardnog ulaza u sljedećem formatu:

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

Ogledni *grader* ispisuje izlazne podatke na tri načina.

Kao prvo, ogledni *grader* ispisuje vaš sklop u datoteku `out.txt` u sljedećem formatu.

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

Drugo, ogledni *grader* simulira kretanje kuglice po sklopu te ispisuje serijske brojeve elemenata u koje kuglica redom ulazi u datoteku `log.txt`.

Treće, ogledni *grader* ispisuje podatke o evaluaciji vašeg sklopa na standardni izlaz.

- Ako je vaš program ocijenjen s rezultatom **Accepted**, ogledni *grader* ispisuje brojeve S i P u obliku `Accepted: S P`.
- Ako je vaš program ocijenjen s rezultatom **Wrong Answer**, ispisuje `Wrong Answer: MSG`. Značenje poruke `MSG` je sljedeće:
 - `answered not exactly once`: Funkcija `answer` nije pozvana točno jednom.
 - `wrong array length`: Duljina niza C nije $M + 1$, ili su duljine nizova X i Y različite.
 - `over 400000 switches`: S je veći od 400 000.
 - `wrong serial number`: Postoji element nizova C , X , ili Y koji je manji od $-S$ ili veći od M .
 - `over 20000000 inversions`: Kuglica se ne vraća u izvor nakon ukupno 20 000 000 promjena stanja klackalica.
 - `state 'Y'`: Postoji klackalica koja je u stanju 'Y' u trenutku kad se kuglica vrati u izvor.
 - `wrong motion`: Niz zvona koji se oglase ne odgovaraju zadanom nizu A .

Imajte na umu da ogledni *grader* ponekad neće napraviti datoteke `out.txt` i/ili `log.txt`, ako je vaš program ocijenjen s rezultatom `Wrong Answer`.