

PUNTEGGIO

0 / 64

+ Richiedi input

TEMPO RIMANENTE

22:58:17

PROBLEMI

[PESCI](#)

0/6

[SOCIAL](#)

0/13

[MOSTRA](#)

0/21

[INTERRUTTORI](#)

0/24

COMUNICAZIONI

[DOMANDE E ANNUNCI](#)

MATERIALE EXTRA

[DOCUMENTAZIONE](#)[HELP](#)

Asocial network

Per aiutarti con questo task, abbiamo preparato delle **tracce di soluzione**, che includono solo le parti di lettura dell'input e scrittura dell'output (da tastiera e su schermo). Puoi decidere se leggere/scrivere su file decommentando le opportune righe di codice.

- Scarica la traccia in C: [social.c](#)
- Scarica la traccia in C++: [social.cpp](#)
- Scarica la traccia in Python: [social.py](#)
- Scarica la traccia in Pascal: [social.pas](#)
- Scarica la traccia in Rust: [social.rs](#)

Descrizione del problema

Stufa dei moderni social network in cui milioni di persone seguono altrettanti milioni di persone, Monica ha deciso di sviluppare un *asocial network*: una piattaforma in cui ogni persona può solamente seguirne esattamente un'altra, a cui dedicare totale ammirazione e passione in tutti i suoi aggiornamenti di stato.

Mojito, il suo cagnolino, si è subito iscritto e ora si chiede chi sia a mettere tutti quei "mi piace" sui suoi ossi. Purtroppo, per motivi di privacy non è possibile sapere chi sia la persona $F[i]$ che l'utente i segue (per i da 0 a $N - 1$, dove N è il numero di iscritti). Tramite le API pubbliche dell'*asocial network*, Mojito ha potuto solamente accedere al numero di follower $A[i]$ che ogni utente possiede, ma ora non sa che farci.

Aiuta Mojito, ricostruendo una possibile assegnazione $F[i]$ tale per cui il numero di follower di ogni utente sia proprio $A[i]$.

Dati di input

La prima riga del file di input contiene un intero T , il numero di casi di test. Seguono T casi di test, numerati da 1 a T . Ogni caso di test è preceduto da una riga vuota.

Ciascun caso di test è composto da 2 righe: la prima contiene l'intero N , mentre la seconda contiene gli N interi $A[i]$.

Dati di output

Il file di output deve contenere la risposta ai casi di test che sei riuscito a risolvere. Per ogni caso di test che hai risolto, il file di output deve contenere una riga con la dicitura

```
Case #t: F[0] F[1] ... F[N-1]
```

dove t è il numero del caso di test (a partire da 1) e $F[0] F[1] \dots F[N - 1]$ è un array di N interi, separati da spazio, dove $F[i] = j$ indica che l'utente i segue l'utente j .

Nel caso ci fossero più soluzioni valide, Mojito si accontenterà di una qualsiasi di esse.

Assunzioni

- $T = 13$, nei file di input che scaricherai saranno presenti esattamente 13 casi di test.
- $2 \leq N \leq 1000$.
- $0 \leq A[i] < N$.

- Un utente non può seguire se stesso, quindi il tuo output deve rispettare $F[i] \neq i$.
- Si garantisce che sia sempre possibile trovare una soluzione.

Esempi di input/output

Input:

```
2
4
0 1 3 0
8
1 1 0 4 1 0 1 0
```

Output:

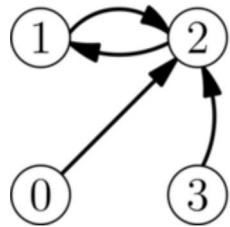
```
Case #1: 2 2 1 2
Case #2: 1 3 3 4 6 3 0 3
```

Spiegazione

Nel **primo caso d'esempio** sono presenti 4 iscritti, rispettivamente:

- L'utente **0** non è seguito da nessun altro utente,
- L'utente **1** è seguito solamente ad un altro utente,
- L'utente **2** è seguito da tutti gli altri tre utenti,
- L'utente **3** non è seguito da nessun altro utente.

L'unica soluzione possibile è la seguente:



Gli utenti **0**, **1** e **3** sono tutti follower dell'utente **2** che a sua volta è follower dell'utente **1**.

Nel **secondo caso d'esempio** una delle possibili soluzioni è la seguente:

