

Traccia prova d'esame

Forti della loro storia accademica, le Università Riunite di Quel'Thalas hanno scelto di potenziare i corsi da offrire ai propri studenti, tenendo in considerazione l'elevato numero di quest'ultimi. Per evitare sovrapposizioni di corsi per gli studenti che seguono due o più corsi, si è deciso di assegnare ad ogni corso uno di tre timeslot differenti, ovvero due corsi distinti, seguiti da uno studente, non possono avere lo stesso timeslot. Poiché i corsi e gli studenti sono tanti, il Rettore Kael'Thalas ci ha chiesto una mano. Dato un insieme di corsi C , ad ogni corso si può assegnare uno tra tre timeslot differenti (10:00 – 12:00, 14:00 – 16:00, 17:00 – 19:00). Avendo in input una lista di scelte L , dove ogni scelta rappresenta due corsi scelti da uno studente, si deve scrivere un programma che dia in output SI se è possibile assegnare i timeslot ai corsi in modo che non ci siano sovrapposizioni, altrimenti NO .

Struttura dell'input

L'input è costituito da diverse righe. La prima riga è nel formato $c\ m$, dove c indica il numero di corsi disponibili e m identifica il numero di scelte presenti. Le successive c righe sono nel formato s , dove s identifica il nome di un corso. Le successive m righe sono nel formato $(a\ b)$ e indicano che uno studente ha scelto di seguire i corsi a e b .

Struttura dell'output

L'output è costituito dalla stringa SI , nel caso in cui esista un assegnamento dei timeslot ai corsi, in modo da non avere sovrapposizioni tra i corsi seguiti; se non esiste alcun assegnamento del genere, l'output è costituito dalla stringa NO .

Esempi input – output

5 5 ASD PO Fond Arch DS (ASD DS) (ASD PO) (ASD Fond) (PO Fond) (Fond Arch)	SI
4 6 CS101 CS201 CS333 CS203 (CS101 CS201) (CS333 CS101) (CS203 CS101) (CS203 CS333) (CS201 CS203) (CS201 CS333)	NO

Regole e istruzioni

- Si può scegliere se usare C++ o Java; in entrambi i casi, si presuppone che lo studente sappia compilare il codice sorgente e avviare l'eseguibile ottenuto tramite terminale/prompt dei comandi.
- Si può assumere che l'input sia sempre corretto.
- Il vostro programma deve **leggere da stdin** e **scrivere su stdout**. La lettura da file o l'hardcoding di un input nel vostro programma **non rappresenta una soluzione corretta**.
- Per effettuare le (vostre) varie prove, potete creare dei file testuali contenenti input di prova e
 - o Scrivere o copiare riga per riga il vostro input al programma,
 - o Reindirizzare il contenuto del file di input al vostro programma (consigliato).
- Il vostro programma verrà valutato su vari input utilizzando il secondo metodo.

Come posso reindirizzare su stdin?

- Supponendo di utilizzare C++ e di essere su Linux/OS X

```
cat input.txt | ./programma
```

dove input.txt è un file testuale contenente un input e programma è l'eseguibile ottenuto dalla compilazione del vostro codice sorgente (input.txt e programma devono essere nella stessa cartella)

- Supponendo di utilizzare C++ e di essere su Windows

```
type input.txt | programma.exe
```

dove input.txt è un file contenete un input e programma.exe è l'eseguibile ottenuto dalla compilazione del vostro codice sorgente (input.txt e programma.exe devono essere nella stessa cartella).