

Traccia prova d'esame

Si è deciso, per il prossimo appello di Algoritmi e Strutture Dati, di creare un sistema automatico per posizionare gli studenti nelle varie postazioni, in modo da evitare leak di codici sorgenti e scopiazzature varie! L'aula dell'esame è rappresentata da una griglia di $n \times m$ posti e ogni posto può ospitare al più uno studente. Dati p studenti (con $p \leq n \times m$), e per ogni studente una lista di studenti "conoscenti", determinare, se esiste, una configurazione di posizionamento degli studenti nella griglia tale che tutti gli studenti siano posizionati sulla griglia e che ogni studente non abbia nel suo intorno (ovvero nelle celle in alto, basso, sinistra, destra e diagonale) un suo conoscente. Per semplicità, si assume che ogni studente sia identificato univocamente da un numero intero e che la cella in alto a sinistra nella griglia abbia coordinate $(0, 0)$.

Struttura dell'input

L'input consiste in un certo numero di righe. La prima riga è nel formato $n \ m \ p$, dove n e m rappresentano la dimensione della griglia e p è il numero di studenti presenti all'esame. Le successive p righe sono nel formato $s_0 \rightarrow s_1 \ s_2 \ \dots \ s_3$ dove s_0 è uno studente e $s_1 \ s_2 \ \dots \ s_3$ la lista dei suoi studenti conoscenti. Ogni studente in questa lista è separato da uno spazio. Si può assumere che l'input sia sempre corretto.

Struttura dell'output

L'output della soluzione consiste in due parti. La prima parte è una riga contenente la stringa SI se esiste una configurazione valida, altrimenti NO. Nel caso in cui esista, la seconda parte dell'output contiene p righe nel formato $s_i = (x, y)$ dove s_i è uno studente e x, y le coordinate in cui è stato posizionato.

Esempio input – output

2 3 4 42 -> 5 8 -> 5 9 5 -> 42 9 -> 42 8	SI 42 = (0,0) 8 = (1,0) 5 = (0,2) 9 = (1,2)
--	---

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).