Studio amico (`studioamico`)

Presso l'ITIS Paleocapa, da qualche anno, si svolge il progetto "Studio amico" in cui uno studente più grande (ovvero frequentante una classe superiore) mette a disposizione gratuitamente il proprio tempo per aiutare uno studente più piccolo con qualche lacuna.

Quest'anno il Dirigente Scolastico ha incaricato il professor Nevabe, che insegna in una classe seconda e in una classe quinta, di organizzare tale attività per la disciplina di sua competenza: informatica.

Il professor Nevabe ha quindi individuato un sottoinsieme di N studenti di classe quinta da appaiare ad altrettanti N studenti frequentanti la classe seconda. L'esperienza insegna che tale aiuto sarà efficace solo se lo studente più grande ha un voto in informatica *strettamente maggiore* dello studente più piccolo.

Preso dai tanti impegni che si avvicendano nella seconda parte dell'anno scolastico, il professor Nevabe non ha tempo di controllare se può effettivamente associare ad ogni studente del secondo anno uno studente del quinto anno adatto. Aiutalo tu!

Implementazione

Dovrai sottoporre esattamente un file con estensione `.c` o `.cpp`.

> ☞ Tra gli allegati a questo task troverai un template (`studioamico.c`, `studioamico.cpp`) con un esempio di implementazione.

Dovrai implementare la seguente funzione:

Funzione `associabili`

C/C++	`bool associabili(int N, int voti2[], voti5[]);`

- L'intero N rappresenta il numero di studenti di ciascuna classe.
- Gli array `voti2` e `voti5`, indicizzati da 0 a $N-1$, contengono rispettivamente i voti degli studenti della classe seconda e i voti degli studneti della classe quinta.
- La funzione dovrà restituire `true` se è possibile trovare un modo di abbinare ogni studente di seconda a uno studente di quinta con un voto strettamente maggiore, `false` altrimenti.

Il grader chiamerà la funzione `associabili` e ne stamperà il valore restituito sul file di output.

Grader di prova

Allegata a questo problema è presente una versione semplificata del grader usato durante la correzione, che potete usare per testare le vostre soluzioni in locale. Il grader di esempio legge i dati da `stdin`, chiama le funzioni che dovete implementare e scrive su `stdout`, secondo il seguente formato.

Il file di input è composto da 3 righe, contenenti:

- Riga 1: l'unico intero N.
- Riga 2: gli N voti degli studenti di seconda.
- Riga 3: gli N voti degli studenti di quinta.

Il file di output è composto da un'unica riga, contenente:

- Riga 1: il valore restituito dalla funzione `associabili`.

Assunzioni

- $1 \leq N \leq 10\,000\,000$.
- I voti degli studenti sono espressi come interi tra 1 e 10: $1 \leq$ `voti2[`i`]`, `voti5[`i`]` ≤ 10 per ogni $i = 0, \ldots, N-1$.
- L'associazione è rigorosamente uno-a-uno: ad uno studente di seconda viene associato uno e un solo studente di quinta (e viceversa).

Assegnazione del punteggio

Il tuo programma verrà testato su diversi test case raggruppati in subtask. Per ottenere il punteggio relativo ad un subtask, è necessario risolvere correttamente tutti i test relativi ad esso.

- **Subtask 1 [0 punti]**: Casi d'esempio.
- **Subtask 2 [30 punti]**: $N \leq 10$.
- **Subtask 3 [30 punti]**: $N \leq 100\,000$.
- **Subtask 4 [40 punti]**: Nessuna limitazione specifica.

Esempi di input/output

stdin	stdout
3 4 6 5 7 6 9	1
3 7 4 5 5 4 10	0

Spiegazione

Nel **primo caso di esempio** uno dei possibili abbinamenti realizzabili consiste nell'assegnare al primo studente di seconda (con voto 4) il secondo studente di quinta (con voto 6), al secondo studente di seconda (voto 6) il primo studente di quinta (voto 7) e infine al terzo studente di seconda (voto 5) il terzo studente di seconda (voto 9).

Nel **secondo caso di esempio** non esiste alcun modo di associare gli studenti in modo che a ogni studente di seconda venga assegnato uno studente di quinta con un voto strettamente maggiore del proprio.