

I grattacieli

I **grattacieli** è un gioco enigmistico con una griglia NxN che rappresenta una città in cui ogni casella è occupata da un edificio, la cui altezza varia da 1 a N piani.

In ogni riga o colonna non ci sono edifici di pari altezza. I numeri esterni indicano quanti edifici si vedono da quel punto (i più alti nascondono i più bassi).

Chi risolve il gioco riempie tutte le caselle della griglia cercando di rispettare tutti i vincoli.

Si realizzi un **programma** che dati i numeri esterni e una griglia **completata** verifica se la soluzione è corretta, ossia:

- ◇ sono rispettati tutti i vincoli di visibilità dei grattacieli
- ◇ in ogni riga e ogni colonna, non ci sono grattacieli di ugual altezza

Specifiche

Il programma acquisisce da tastiera in ingresso il nome di un file binario (al più 13 caratteri, inclusi percorso ed estensione) che contiene:

- ◇ la dimensione della griglia (dimensione massima 10)
- ◇ i dati della griglia,
- ◇ i numeri esterni, in senso orario a partire da quello in alto a sinistra.

Il programma al termine dell'elaborazione visualizza 1 se la soluzione è valida, 0 altrimenti, seguito da UN carattere a capo ('\n').

Ipotesi di lavoro

- ◇ il file contiene tutti e soli i dati necessari (valori interi di tipo int),
- ◇ le altezze inserite e i numeri esterni sono senz'altro tutti compresi tra 1 e N

Vincoli

- ◇ non è possibile utilizzare variabili globali
- ◇ è possibile utilizzare i sottoprogrammi delle librerie standard (stdio, string, ...)
- ◇ ogni membro del gruppo deve scrivere **autonomamente** almeno un sottoprogramma

Aspetti di rilievo nella valutazione della qualità della soluzione proposta (oltre alla correttezza)

- ◇ utilizzo appropriato della memoria
- ◇ prestazioni

Compilazione ed esecuzione del codice

Sottomettere il **sorgente** (un unico file), che viene compilato con le opzioni (standard ANSI C 89):

```
-Wall -O2 -std=c89 -pedantic
```

Fate quindi attenzione a non utilizzare commenti // ma solo /**/.

Il risultato dell'esecuzione è confrontato automaticamente con il risultato atteso, carattere a carattere. Ogni differenza è considerata un errore, quindi è fondamentale NON aggiungere stampe a video di servizio.

Esempio di contenuto del file di partenza (il file veramente usato è binario)

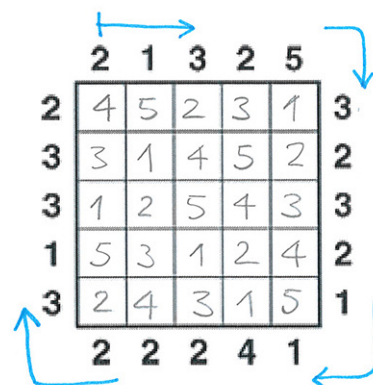
```
5 4 5 2 3 1 3 1 4 5 2 1 2 5 4 3 5 3 1 2 4 2 4 3 1 5 2 1 3 2 5 3 2 3 2 1 1 4 2 2 2 3 1 3 3 2
```

Sottomissione della soluzione

È stato creato un account per la vostra squadra (codice utente del referente) per sottomettere la soluzione al seguente indirizzo:

<http://poseidon.ws.dei.polimi.it/contests/>

L'autenticazione è tramite AUnica del Politecnico











Casi di test

Sono disponibili e verranno utilizzati 5 casi di test: per ogni caso di test è disponibile il file di ingresso e l'uscita attesa.

Prima di sottomettere la soluzione, verificate che **per ogni caso di test** il vostro programma visualizzi la soluzione corretta, e con il formato richiesto (un intero seguito da un a-capo).

Risultato dell'analisi

Il verificatore compila ed esegue il vostro programma in relazione ai 5 casi di test, e visualizza uno di questi risultati:

Risultato	Spiegazione
 CORRECT	Il programma è stato compilato ed eseguito con successo e ha fornito la risposta corretta per tutti i casi di test
 NOT CORRECT	Il programma è stato compilato ed eseguito con successo e ha fornito la risposta corretta per tutti i casi di test: la soluzione adottata però non soddisfa le specifiche date (ad esempio non si crea una nuova variabile, ma si visualizza il risultato dell'elaborazione, oppure si introduce una dimensione arbitraria di un array mentre la soluzione presuppone che non si debba memorizzare i dati ...). La soluzione non è considerata valida.
 TIME LIMIT	Il programma è stato compilato con successo ma una volta eseguito non ha completato l'elaborazione entro il tempo limite fissato. Prova a verificare che non ci siano cicli infiniti e prova ad ottimizzare l'algoritmo.
 NO OUTPUT	Il programma è stato compilato con successo ma una volta eseguito non ha prodotto alcun risultato. Questo può essere dovuto alla presenza di cicli infiniti (e ad un certo punto l'esecuzione viene interrotta).
 PRESENTATION ERROR	Il programma è stato compilato ed eseguito con successo ma il risultato prodotto non è identico a quello atteso in termini di presentazione. Spesso non sono stati rispettati i vincoli sul modo di visualizzare il risultato (ad esempio, presenza di eccessivi spazi o mancanza di un "a capo").
 WRONG ANSWER	Il programma è stato compilato ed eseguito con successo ma il risultato prodotto non è quello atteso per almeno uno dei casi di test.
 RUN ERROR	Il programma è stato compilato con successo ma durante l'esecuzione è andato in errore. Questo si verifica di solito per problemi di allocazione di memoria, di accessi a indirizzi sbagliati, di divisioni per zero.
 COMPILATION ERROR	Non è stato possibile compilare il programma. Controlla con attenzione i messaggi generati dal compilatore (facendo distinzione tra avvertimenti Warning ed errori Error). Il compilatore viene chiamato con le seguenti opzioni: <code>-Wall -O2 -std=c89 -pedantic -static</code>