

S U D O K U

Specifica

Il Sudoku ([wikipedia](#)) è un puzzle numerico, che si gioca con una griglia 9x9 suddivisa in nove riquadri 3x3. Lo scopo del gioco è completare la griglia (partendo da una griglia iniziale parzialmente riempita) con i numeri da 1 a 9, in modo tale che in ogni riga (orizzontale), in ogni colonna (verticale) ed in ogni riquadro 3x3 siano presenti tutte le cifre da 1 a 9, senza ripetizioni.

Si vuole realizzare un programma per giocare a Sudoku che offra all'utente il seguente menù:

```
1 - avvia una nuova partita
2 - inserisci valore
3 - cancella valore
4 - verifica la soluzione attuale
5 - carica una soluzione e verificala
6 - riavvia partita attuale
0 - esci
```

All'avvio, il programma chiede all'utente di inserire lo schema della nuova partita (corrisponde implicitamente alla voce 1 del menù) e, se quello caricato è uno schema valido, visualizza la griglia e presenta il menù. Ogni volta che l'utente seleziona un'opzione del menù, il programma esegue la corrispondente azione, visualizza eventuali messaggi, visualizza la griglia con lo stato attuale dello schema, e visualizza nuovamente il menù.

Quando l'utente inserisce l'ultimo valore, il programma verifica la soluzione (senza che l'utente debba esplicitamente selezionare l'opzione 4). Nel caso in cui la partita sia terminata perché lo schema è completo e corretto, il programma scrive un messaggio

opportuno e chiede all'utente se vuole ripartire con una nuova partita o terminare.

Ulteriori indicazioni vincolanti

- all'avvio del programma viene fornito lo schema della partita in termini di una stringa di 81 caratteri. Si assuma che la dimensione della stringa sia sempre corretta. I valori ammessi sono i caratteri da '1' a '9' (corrispondenti ai valori da mettere nella griglia) e il carattere '_' per indicare una cella vuota (è anche possibile utilizzare il carattere '0').
- la visualizzazione della griglia deve mostrare i bordi dei vari quadranti ed evidenziare quali valori appartengono allo schema iniziale e quali sono stati aggiunti durante il gioco. Un modo per evidenziare i valori iniziali può essere aggiungervi un asterisco sulla destra, come mostrato nel seguente esempio:

```
+-----+-----+-----+
|      9*      |      |      4      |      |
|  8*      5  |      4*  |  1*      |      |
|      4*      |  6*  8*  1*  |  5*  3*  9*  |      |
+-----+-----+-----+
|      *8      |      4*  9*  |  7*      3*  |      |
|      7*      |      6*      |      4  8*  |      |
|  4*  1*      |  8*  3*      |  6*      |      |
+-----+-----+-----+
|  3*  5*      |  4*      |  2*  9*  1  |      |
|  9*      2  |  1*      3*  |      7  6*  |      |
|  1*      |      6      |      5*  4  |      |
+-----+-----+-----+
```

- l'**avvio di una nuova partita** richiede l'inserimento di un nuovo schema rispettando il formato precedentemente specificato. Il nuovo schema va a sovrascrivere lo schema precedente e l'eventuale soluzione (parziale o completa che sia).

- l'**inserimento** di un valore richiede che il programma acquisisca da tastiera le coordinate della cella ed il valore da inserire. Il programma deve verificare che le coordinate siano valide (siano all'interno della matrice), che il valore sia compreso tra 1 e 9 e che la cella sia vuota. In caso di errore il programma visualizza un apposito messaggio (senza richiedere i dati corretti), seguito dalla griglia e dal menù. L'utente inserisce i valori facendo riferimento ad una numerazione delle posizioni che parte da 1 (ignora il fatto che in C si parta da 0), per cui, per inserire il valore nella posizione indicata nell'esempio riportato di seguito, specificherà la seguente tripletta: 2 5 7 (riga 2, colonna 5, valore 7).

v

	+	-----	+	-----	+	-----	+
		9*				4	
>		8*	5		7	4*	1*
			4*		6*	8*	1*
					5*	3*	9*
	+	-----	+	-----	+	-----	+
			8		4	9*	7*
		7*				6*	4
		4*	1*		8*	3*	6*
	+	-----	+	-----	+	-----	+
		3*	5*		4*		2*
		9*	2		1*	3*	7
		1*			6		5*
	+	-----	+	-----	+	-----	+

- la **cancellazione** di un valore richiede che il programma acquisisca da tastiera le coordinate della cella da cancellare. Il programma deve verificare che le coordinate siano valide e che la cella non contenga un valore dello schema iniziale (i valori dello schema iniziale non possono mai essere cancellati). In caso di errore il programma visualizza un apposito messaggio (senza richiedere i dati corretti). Nel caso in cui la cella non

contenga alcun valore, il programma non effettua alcuna azione e viene riproposta la griglia ed il menù.

- la **verifica** della soluzione attuale effettua un controllo della griglia in quel momento (che si tratti di una soluzione completa o meno) e notifica l'utente dell'esito ("VALIDA" oppure "ERRATA"). Nel caso in cui la soluzione sia completa e corretta, il programma notifica l'utente ("CORRETTA").
- l'opzione di **caricamento** di una soluzione (completa o meno) consente di acquisire una nuova stringa che contiene i valori della soluzione da caricare. Il programma verifica innanzitutto che essa sia compatibile con lo schema iniziale (i valori presenti nello schema iniziale devono sempre essere al loro posto) e se così è, l'attuale configurazione della griglia sostituisce quella precedente avviandone la verifica (opzione 4 del menù). Nel caso in cui la stringa inserita non sia valida o non sia compatibile con lo schema, viene visualizzato un opportuno messaggio e riproposto il menù.
- la richiesta di **riavvio** della partita riporta la griglia alla configurazione iniziale dell'ultimo schema inserito, cancellando tutti i valori inseriti dall'utente.

Note per la realizzazione

- identificare e definire le opportune strutture dati per la gestione delle informazioni
- definire un insieme di sottoprogrammi che consentano di realizzare le funzionalità richieste, specificando parametri di ingresso/uscita
- per semplificare la collaborazione nello sviluppo della soluzione complessiva, si suggerisce l'utilizzo di una piattaforma di sviluppo collaborativa: ce ne sono diverse e tra queste [CodeCollab](#) non richiede di registrarsi. E' sufficiente che una persona inizi a lavorare e condivida il link con le altre persone

del gruppo perché si possa "lavorare" tutti nello stesso spazio o quantomeno avere un unico punto in cui collezionare tutti i contributi. E' sufficiente seguire l'opzione "Get Started" per avviare una collaborazione senza doversi registrare.

Prima di iniziare a stendere il codice, mostrare al responsabile/tutor l'organizzazione del progetto in relazione alle strutture dati e ai sottoprogrammi progettati

Esempi

schema iniziale:

```
__68__2____26_9_5_9_3____9__15____5____7_378_____
```

parziale:

```
__68__2__7_3__26_9_5_9_37_4_9_7_15____5____7_378_____
```

corretta:

```
91687425374315268985296371429473156816548937237862549143
```

errata:

```
91687425373415268985296371429473156816548937237862549143
```

Per i responsabili

- verificare che le strutture dati siano tali da consentire di salvare le informazioni necessarie e siano facilmente manipolabili
- verificare che i sottoprogrammi siano ragionevoli in termini di complessità e che ricevano/trasmittano/restituiscano i parametri/risultati necessari