

Linguaggio C

Si consideri il gioco dell'*impiccato*: un *giocatore* deve indovinare una *parola* p , lunga al massimo 30 caratteri, che all'inizio viene mostrata con un numero di caratteri “_” pari alla sua lunghezza effettiva. Ad ogni turno il giocatore specifica una lettera c , e se questa è contenuta nella parola p da indovinare, tutte le occorrenze di c vengono visualizzate in p al posto di “_”, altrimenti il tentativo viene contato come *errore*. Il giocatore *vince* se indovina tutte le lettere contenute nella parola p senza commettere più di **10 errori**. Ogni giocatore ha associato un *codice*, un *insieme di parole indovinate* (**la dimensione di tale insieme non è nota a priori**) e un *punteggio* (dato dalla somma delle lunghezze delle parole indovinate).

Si definisca un **progetto** in linguaggio C in cui siano implementate le seguenti funzionalità:

- 1) le strutture dati più adeguate per rappresentare un **insieme di giocatori** (**la dimensione di questo insieme non è nota a priori**), e i file **header** del progetto (**.h**);
- 2) la funzione `indovina` che accetti come parametro una parola p e realizzi il gioco dell'impiccato ipotizzando che ci sia un giocatore che da tastiera digiti le varie lettere da indovinare. Tale funzione restituirà `true` in caso di vittoria del giocatore, `false` altrimenti. Ad esempio, si consideri l'esecuzione della funzione `indovina` sulla parola `cocco`, come di seguito riportato:

```
Parola attuale: _ _ _ _ _
Digita una lettera: c

Giusto!
Parola attuale: c _ c c _
Digita una lettera: e

Sbagliato!
Hai altri 9 tentativi a disposizione.
Parola attuale: c _ c c _
Digita una lettera: o

Giusto!
Complimenti hai indovinato la parola: cocco
```

In particolare, la funzione `indovina` chiede al giocatore di digitare una lettera da tastiera: se questa è contenuta nella parola p , allora la funzione visualizza un messaggio (**Giusto!**) e tutte le occorrenze di quella lettera in p . Nel caso in cui la lettera inserita non sia contenuta nella parola p , allora la funzione visualizza un messaggio (**Sbagliato!**) e il numero di tentativi ancora a disposizione (ad esempio **Hai altri 9 tentativi a disposizione.**). Nel caso in cui il giocatore indovini l'intera parola, la funzione visualizza un messaggio (come mostrato nell'esempio) e la parola completamente scoperta, altrimenti interrompe la richiesta di digitare lettere e visualizza il messaggio: **Mi spiace hai commesso troppi errori.**

- 3) la funzione `inserisciParola` che dato l'insieme di giocatori G , il codice `cod` di un giocatore e una parola p , esegua il gioco dell'impiccato sulla parola p (come descritto al punto precedente) ipotizzando che il giocatore alla tastiera abbia codice pari a `cod`; se il giocatore indovina la parola allora `inserisciParola` inserisce p nell'insieme di parole indovinate del giocatore in G che ha il codice pari a `cod` e aggiorna il punteggio. Nel caso in cui non esista in G un giocatore con il codice `cod`, allora la funzione `inserisciParola` crea un nuovo giocatore con codice pari a `cod`, con un insieme di parole composto dalla parola indovinata e con il relativo punteggio, e inserisce tale giocatore nell'insieme G . Nel caso in cui il giocatore non vinca la funzione non esegue alcun inserimento. Si consideri ad esempio il seguente insieme di due giocatori

Codice: 1 Punteggio: 7 Parole indovinate: [casa] [asd]	Codice: 2 Punteggio: 17 Parole indovinate: [oro] [villa] [algoritmi]
--	--

si consideri di invocare la funzione nel seguente modo: `inserisciParola(G,1,"cocco")`. Quindi se il giocatore indovinerà la parola come mostrato al punto 2 allora l'insieme G sarà aggiornato come segue

Codice: 1 Punteggio: 12 Parole indovinate: [cocco] [casa] [asd]	Codice: 2 Punteggio: 17 Parole indovinate: [oro] [villa] [algoritmi]
---	--

E' possibile definire metodi di supporto e utilizzare ogni libreria conosciuta.