

Programmazione I – Informatica Generale

Prova di Programmazione – 1 Luglio 2009 – Zore 30 min

Consideriamo un gioco in cui, all'inizio di una ogni partita, sono posizionate N^2-1 tessere su una *tavola quadrata*, suddivisa in N^2 celle identiche. Ciascuna tessera è posizionata su una cella distinta ed è etichettata con un carattere. Ad esempio, rappresentando col simbolo – la cella vuota, una

tavola da nove celle potrebbe contenere le tessere:

C	-	B
S	I	T
O	T	O

. Per vincere una partita occorre spostare

le tessere, rispettando le regole contenute nei prossimi punti, fino a quando tutte le celle, tranne l'ultima in basso in destra, sono occupate, e la sequenza di caratteri presenti sulle tessere, ottenuta leggendo le righe da sinistra verso destra e l'una dopo l'altra, è uguale alla *parola obiettivo* definita all'inizio della partita. Ad esempio, se la parola obiettivo fosse BISCOTTO, le tessere andrebbero

spostate fino ad avere la configurazione:

B	I	S
C	O	T
T	O	-

. A partire dal frammento di codice fornito

realizzare una versione con interfaccia testuale del gioco in cui vi sia una sola tavola di dimensioni definite a tempo di scrittura del programma e che fornisca le seguenti funzionalità:

1. Inizializzazione del gioco: inizialmente non vi sono tessere sulla tavola e non vi è nessuna partita in corso.
2. Nuova partita: si inseriscono la parola obiettivo e la configurazione iniziale delle tessere, sotto forma di stringa ed utilizzando il carattere – per rappresentare la cella vuota. Inserendo ad esempio *BISCOTTO* e *C-BSITOTO* si ottengono la configurazione iniziale e la parola obiettivo mostrate nell'esempio iniziale. L'eventuale partita in corso viene persa.
3. Stampa disposizione tessere per una data tavola.
4. Controllo vittoria per una data tavola. In caso di vittoria la partita in corso viene terminata e le tessere vengono rimosse dalla tavola.
5. Memorizzazione della partita in corso su file (dal nome definito a tempo di scrittura del programma).
6. Caricamento di una partita da file.
7. Spostamento di una tessera: si inseriscono le coordinate della tessera da spostare e la direzione verso cui va spostata (*SINISTRA*, *DESTRA*, *ALTO*, *BASSO*). La tessera può essere spostata solo se c'è una partita in corso e la cella di destinazione è vuota. Ad ogni spostamento si controlla l'eventuale vittoria, agendo come nel punto 4.
8. Nuova partita con disposizione casuale tessere: uguale al punto 2, tranne che, letta solo la parola obiettivo, le tessere vengono disposte automaticamente in modo casuale sulla tavola. Il punteggio massimo si ottiene se nella soluzione proposta la funzione *rand* è invocata un numero di volte non maggiore della lunghezza della parola più uno.

Gestire opportunamente le situazioni di errore, tranne gli *overflow* dovuti a valori assoluti troppo elevati e l'inserimento di dati in formato errato da *stdin*.

REGOLE

- Si può utilizzare ogni genere di materiale di manuale o altro materiale didattico
- Per superare la prova, il programma deve essere perfettamente funzionante almeno nelle parti 1, 2, 3, e 4. Il voto ottenuto in questo caso è 18.
- Ciascuna funzionalità DEVE essere implementata mediante almeno una funzione.
- Il voto massimo (almeno 30) si ottiene se
 - a) il programma è perfettamente funzionante in ogni sua parte
 - b) tutti i principi di ingegneria del codice visti nel corso sono stati applicati