Programmazione I Prova di programmazione – 12 Gennaio 2023 – <u>2 ore</u>

Partendo dal frammento di codice fornito, realizzare un gioco per due giocatori, che chiamiamo semplicemente giocatore 0 e giocatore 1. Ciascun giocatore dispone di una fila di N combattenti, individuati ciascuno da un indice che va da 0 ad N-1. I giocatori giocano a turno, per N turni ciascuno. In ciascun turno cercano di eliminare uno dei combattenti dell'altro giocatore. Alla fine dei turni, vince il giocatore che ha eliminato più combattenti dell'altro. All'avvio del programma, le due file non possono contenere alcun combattente, ossia N=0. Il programma fornisce le seguenti funzionalità.

- 1. [+1] inizia_partita(N) Inizializza le due colonne a contenere N combattenti ciascuna, e, se N>0, allora inizializza il giocatore 0 come di turno. L'eventuale precedente stato del gioco è perso. Non vi è alcun limite esplicito al valore massimo di N. Si ottiene il punteggio aggiuntivo se si evitano deallocazioni e riallocazioni di memoria non necessarie.
- 2. **stampa_stato** Stampa lo stato del gioco combattenti, rappresentando ciascun combattente presente con un asterisco e ciascun combattente eliminato con un trattino. Dopo ciascuna fila di combattenti stampa il numero di turni rimasti per il giocatore proprietario di quella fila. Se si tratta del giocatore di turno, stampa tale numero tra parentesi angolate. Ad esempio, per una partita appena iniziata con N=6:

***** <6>

3. **elimina_combattente**(\mathbf{f} , \mathbf{i}) Calcola il valore $c=(\mathbf{i}^N)\% N$, e, se c < N e c'è un combattente in posizione c nella fila \mathbf{f} , elimina tale combattente. Per calcolare c, si può sfruttare la seguente proprietà: dato un qualsiasi numero naturale j, ($j * \mathbf{i}$) % $N = ((j\% N) * \mathbf{i})$ % N. Ad esempio, se invocata con f = 0 ed i = 2 sullo stato iniziale dell'esempio precedente, si ha c = 4, e si ottiene

****-* <6>

- 4. **[3] esegui_turno(i)** Se i turni sono già finiti per entrambi i giocatori, stampa PARTITA FINITA, altrimenti esegue la **elimina_combattente(f, i)** sulla fila opposta a quella del giocatore di turno. Se questo non è l'ultimo turno dell'ultimo giocatore, allora decresce il valore del turno e configura l'altro giocatore come giocatore di turno. Altrimenti, stampa VINCE x, ove x è uguale a 0 oppure 1, se il giocatore x ha vinto. Oppure stampa PAREGGIO se i due giocatori hanno pareggiato.
- 5. [3] salva_partita Salva lo stato della partita in un file di testo dal nome predefinito.
- 6. [2, +1] carica_partita Carica lo stato dal file. L'eventuale precedente stato è perso. Si ottiene il punteggio aggiuntivo se si evitano deallocazioni e riallocazioni di memoria non necessarie.
- 7. **[5] sposta_combattenti(f, s)** Sposta uno alla volta tutti i combattenti della fila **f**, a partire da quello in posizione più alta e fino a quello nella posizione più bassa. In particolare, sposta ciascun combattente di **s** posizioni in avanti, se tali **s** posizioni sono tutte libere; altrimenti lo sposta sulla più alta posizione libera disponibile tra tali **s** posizioni. Ad esempio, dal seguente stato a sinistra, si arriva allo stato a destra se si effettua uno spostamento con **f**==0 e con **s**==2 oppure **s**==3:

I parametri di ingresso delle funzionalità sono solo indicativi. Gestire opportunamente le situazioni di errore, tranne l'*overflow* e l'inserimento di dati in formato errato da *stdin*.

Per il collaudo: se fate stampare messaggi per invitare l'utente ad inserire valori, ricordate di aggiungere la stampa di caratteri accapo. Altrimenti nel puro output del programma vi saranno delle righe fuse, e di fatto tale output non sarà quello che credete (le righe fuse non le vedete quando usate il programma da terminale, perché inserite voi l'accapo da utenti).

REGOLE

- Si può utilizzare ogni genere di manuale e di materiale didattico
- Per superare la prova, bisogna svolgere almeno i punti 1, 2 e 3. Se si svolgono solo tali punti, il programma deve essere perfettamente funzionante. Il voto ottenuto in questo caso è 18.
- Ciascuna funzionalità DEVE essere implementata mediante almeno una funzione.
- Il voto massimo si ottiene se

- a)
- il programma è perfettamente funzionante in ogni sua parte tutti i principi di ingegneria del codice visti nel corso sono stati applicati b)
- sono state seguite eventuali altre indicazioni presenti nella traccia in merito al voto finale c)