

Preselezione della Venticinquesima Gara di Informatica per studenti delle Scuole Superiori

Esercizi

AVVISI:

- Se non specificato altrimenti, le sequenze iniziali su nastro si intendono *non vuote*, ovvero contenenti almeno un simbolo.
- Per numero decimale si intende un numero positivo o nullo rappresentato con le cifre 0,1,2,...,9, senza zeri iniziali non significativi. Per esempio, 0 e 19 sono numeri validi, mentre 0032 deve essere scritto come 32.
- Nel fornire le soluzioni, ricordarsi di pulire il nastro finale da ogni simbolo che non costituisca la risposta!
- Ogni volta che si salva la soluzione di un esercizio con il simulatore della macchina di Turing, il “timestamp” dell’esercizio viene aggiornato con il tempo trascorso fino a quel momento.
- Il tempo disponibile è di 2 ore e 30 minuti, qualsiasi esercizio non salvato nel sistema di gara allo scadere del tempo non sarà considerato.

Esercizio 1 Conta prese [Punti 2].

Consideriamo un mazzo di carte piacentine, dove il valore va da uno (Asso) fino a dieci (Re). Ogni carta è rappresentata da un numero decimale ad una cifra (per il momento ignoriamo i semi), in particolare l’asso corrisponde al numero 1, il due al 2, e così via fino al 9, mentre il dieci è rappresentato dal numero 0.

Alla fine di una partita, bisogna contare quante carte un giocatore ha preso. Si scriva un programma per macchina di Turing che, ricevuto sul nastro una sequenza di simboli indicante le carte prese, lasci sul nastro il numero di carte che compone la sequenza, espresso come numero decimale.

Le carte prese potrebbero essere 0 (nastro vuoto), ma si ha la garanzia che non siano più di 40, ovvero la dimensione del mazzo.

Esempio

Nastro Iniziale	Nastro Finale
	0
53412201	8
484198750487	12
0	1

Esercizio 2 Alto basso [Punti 3].

Nel gioco “Alto basso” abbiamo una carta scoperta sul tavolo, e un giocatore che scommette alto (A) oppure basso (B). Si scopre a questo punto una nuova carta: il giocatore vince (V) se ha scommesso “alto” e la seconda carta è più alta della precedente, oppure se ha scommesso “basso” e la seconda è più bassa, mentre perde (X) in caso contrario. Se le due carte sono uguali tuttavia il round è in ogni caso nullo (N).

Sul nastro è data la prima carta, la scommessa del giocatore, e la seconda carta (ad es, 3A5, che corrisponde a una vittoria del giocatore, in quanto ha scommesso “alto”, e in effetti dopo il 3 viene scoperto il 5, una carta più alta).

Si scriva un programma per macchina di Turing che, ricevuto in input il nastro descritto, lasci sul nastro solamente il risultato della scommessa (V, X, o N).

Esempio

Nastro Iniziale	Nastro Finale
3A5	V
4A3	X
3B3	N
1B2	X
1A0	V

Esercizio 3 Asso piglia tutto [Punti 4].

Nel gioco “asso piglia tutto” il giocatore ha 3 carte in mano, e può usare una carta soltanto per prenderne una dello stesso valore (ad es, si può usare un 5 per prendere un 5, invece NON si può usare un 5 per prendere un 2 e un 3). L’asso è l’unica carta che si comporta diversamente, in quando giocando l’asso si prendono tutte le carte a tavola. Vogliamo eseguire la mossa del giocatore. Il giocatore:

- se ha un asso, lo usa per prendere tutte le carte in tavola
- se non ha un asso, prende con la carta più alta con cui è possibile, dove la carta più alta è il dieci (0), a seguire il 9, e così via.
- se non può prendere, lascia in tavola la carta più bassa che ha.

Vengono date in input la mano del giocatore e carte in tavola separate dal simbolo # (ad es, se il giocatore ha in mano 123, e in tavola abbiamo 456, il nastro in input sarà 123#456). Ci sono sempre 3 carte (diverse fra loro) in mano al giocatore, e le carte in tavola (anche esse diverse fra loro) sono almeno una.

Si scriva un programma per macchina di Turing che, ricevuto in input il nastro con la mano e la tavola, lasci sul nastro la mano e la tavola dopo la presa del giocatore. Le carte prese sono scartate, e non va cambiato l’ordine delle carte (inoltre, se si lascia una carta in tavola va messa in fondo).

Esempio

Nastro Iniziale	Nastro Finale	Note
123#345	23#	l’asso piglia tutto
263#293476	23#29347	il 6 è la presa piu alta)
509#509	59#59	il dieci (0) è la presa piu alta
329#456	39#4562	nessuna presa, si lascia il 2

Esercizio 4 Shuffle perfetto [Punti 5].

Saper fare un ottimo shuffle è un’abilità fondamentale per un croupier. Lo shuffle perfetto divide il mazzo in due parti uguali, e lo ricompone alternando una carta da ogni parte, partendo dal fondo. Ad esempio, se il nostro mazzo corrisponde a 123456, verrà separato come “123” e “456”, e verrà ricomposto prendendo nell’ordine 1, 4, 2, 5, 3, e in fine 6.

Si scriva un programma per macchina di Turing che, ricevuto in input il mazzo, lasci sul nastro il mazzo ricomposto dopo aver effettuato uno shuffle. Il mazzo è sempre composto da un numero pari di carte.

Esempio

Nastro Iniziale	Nastro Finale
11112222	12121212
123456	142536
368142197524	316987154224

Esercizio 5 Dare le carte [Punti 8].

Il croupier mischia le carte e le distribuisce tra 4 giocatori. Per mischiarle fa 3 shuffle perfetti, poi le distribuisce una ad una partendo dal fondo del mazzo.

Ad esempio, se il mazzo in input è 111122223333, avremo dopo il primo shuffle 121213132323, dopo il secondo 112312231233, e in fine 121321321323; a questo punto si procede a dare le carte partendo dalla carta più a sinistra (1) e dandole a turno al primo, secondo, terzo e quarto giocatore, ottenendo le 4 mani 121#213#132#323 (separate dal carattere #). Il numero di carte dato è sempre un multiplo di 4 (e almeno 4), quindi ogni giocatore riceve sempre almeno una carta, e lo stesso numero degli altri giocatori.

Si scriva un programma per macchina di Turing che, ricevuto in input il mazzo, lasci sul nastro le 4 mani ricevute dai giocatori separate da #.

Esempio

Nastro Iniziale	Nastro Finale
1234	1#3#2#4
11112222	12#12#12#12
111122223333	121#213#132#323
368142197524	316#987#154#224

Esercizio 6 Briscola [Punti 7].

Le cose si fanno più complicate: oltre ai numeri delle carte ora abbiamo bisogno dei semi! Ogni carta, oltre al suo valore, può essere di Bastoni (B), Coppe (C), Denari (D) o Spade (S), ad esempio 1C rappresenta l'asso di coppe, 7D il 7 di denari, e 0S il 10 di spade. Nella briscola, la carta più alta nell'ordine di presa è l'asso (1), poi abbiamo il 3, e poi a scendere 0, 9, 8, 7, 6, 5, 4, 2. Nel gioco viene scelto a inizio partita un seme di "briscola"; per semplicità sarà sempre "bastoni".

Ad ogni turno, 4 giocatori mettono a terra una carta ciascuno a partire dal primo, e vince la presa la carta più alta del seme dominante (ovvero il seme giocato dal primo giocatore), a meno che non sia stata giocata una carta del seme di briscola, in qual caso vince la presa la briscola più alta (ovvero la carta più alta di bastoni).

Ad esempio, se il primo giocatore gioca il 10 di denari (0D), il secondo giocatore può superare e provare a vincere la presa giocando una carta più alta di denari (ad es 3D) oppure una qualsiasi carta di briscola (ad es 5B).

Si scriva un programma per macchina di Turing che, ricevuto in input il nastro con le 4 carte giocate in ordine (ovvero la carta a sinistra è quella del primo giocatore), lasci sul nastro solamente la carta che vince la presa.

Esempio

Nastro Iniziale	Nastro Finale
0D3D2D1D	1D
0D3D2D1C	3D
0D3D2B1C	2B
4S1D0S9C	0S
4S0B9C1B	1B
2C2D2S2B	2B

Esercizio 7 Scopa [Punti 15].

Nel gioco della scopa, il giocatore ha 3 carte in mano, che può usare per prendere le carte in tavola. Per semplicità, con una carta si prende sempre solo una carta (ad es, si può usare un 5 per prendere un 5, invece NON si può usare un 5 per prendere un 2 e un 3). A fine partita viene assegnato un punto a chi possiede il “sette bello” (7D), un punto a chi possiede il maggior numero di carte di denari, e un punto a chi possiede il maggior numero di carte (ignoriamo qualsiasi altra regola).

Dato un nastro che contiene la mano del giocatore e le carte in tavola, separate da #, vogliamo scrivere un programma che effettui la presa migliore per il giocatore. Ovvero:

- se possibile prendere il 7 di denari (7D), lo si prende.
- se non possibile, si cerca di prendere una carta denari (in caso di parità, giocando la carta più alta).
- se non possibile, si cerca di prendere la carta più alta.
- se non possibile, si lascia a tavola la carta più bassa.

Si tenga conto che nella scopa l'ordine delle carte è semplicemente 0,9,8,7,6,5,4,3,2,1, dove il 10 (0) è la più alta, e l'asso (1) la più bassa.

Per evitare ambiguità, in mano non ci saranno mai due carte di semi diversi con lo stesso valore (ad es, non avremo mai in mano sia 5C che 5S), e lo stesso vale per le carte in tavola. Si osservi anche che con le regole date è impossibile prendere due carte di denari in una sola presa. Si può prendere una carta di denari ad esempio usando 2S per prendere 2D, ma anche usando 5D per prendere 5C (dato che viene presa anche la carta giocata): in questo caso si preferisce la seconda mossa dato che usa la carta più alta.

Si scriva un programma per macchina di Turing che, ricevuto in input il nastro con la mano e la tavola, lasci sul nastro la mano e la tavola dopo la presa del giocatore. Le carte prese sono scartate, e non va cambiato l'ordine delle carte (inoltre, se si lascia una carta in tavola va messa in fondo).

Esempio

Nastro Iniziale	Nastro Finale	Note
2S5D7C#5C2D7D	2S5D#5C2D	è possibile prendere il 7 di Denari
2S5D7C#5C2D7S	2S7C#2D7S	è possibile prendere denari col 5
2S5S7C#5C2D7S	5S7C#5C7S	è possibile prendere denari col 2
9S2D4D#0S8C7B	9S4D#0S8C7B2D	si lascia a tavola la carta più bassa, 2D