

Ventiseiesima Gara di Programmazione della Macchina di Turing per studenti delle Scuole Superiori

Esercizi

AVVISI:

- Se non specificato altrimenti, le sequenze iniziali su nastro si intendono *non vuote*, ovvero con 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, si sovrascrive il salvataggio precedente e il “timestamp” dell’esercizio viene aggiornato col tempo trascorso fino a quel momento.
- Il tempo disponibile è di 3 ore, qualsiasi esercizio non salvato nel sistema di gara allo scadere del tempo non sarà considerato.



Esercizio 1 London Bridge is falling down [Punti 1].

Siamo nel Regno Unito per assistere all’incoronazione di Re Carlo III. Stiamo andando verso l’albergo a riposarci... Il ponte di Londra crolla! Dovremo prendere un’altra strada.

Fortunatamente il ponte ha degli enormi pilastri, che riescono a ridurre i danni. I segmenti del ponte protetti da un pilastro sia a destra che a sinistra si salveranno, e anche i pilastri stessi, mentre crollerà tutto il resto.

Si scriva un programma per macchina di Turing che, ricevuto sul nastro il ponte, con i segmenti rappresentati dal simbolo =, e i pilastri dal simbolo P, termini lasciando sul nastro la porzione del ponte rimasta in piedi.

Esempio

Nastro Iniziale	Nastro Finale
==P===P==	P===P
P=====P	P=====P
==P===PP===P=	P===PP===P
=====P=====	P
=====	



Esercizio 2 Qual giorno? [Punti 3].

Ci svegliamo nel nostro albergo e cerchiamo di capire se oggi è il gran giorno. Che confusione!

Gli inglesi usano il formato *mese/giorno*, contrariamente al nostro *giorno/mese*, quindi 5/6 è il 6 Maggio in formato UK, ed è invece il 5 Giugno in formato IT. A volte, il formato è ovvio: ci sono solo 12 mesi, quindi 13/5 sarà sicuramente il 13 Maggio in formato IT, mentre 11/20 sarà sicuramente il 20 Novembre in formato UK, mentre con 5/6 non possiamo essere sicuri.

Si scriva un programma per macchina di Turing che, ricevuta sul nastro una data, termini lasciando sul nastro la scritta IT se è sicuramente in formato italiano, la scritta UK se è sicuramente in formato inglese, e ? altrimenti. Si assuma che la data in input sia valida in almeno uno dei due formati, e che siano omessi gli zeri non significativi.

Esempio

Nastro Iniziale	Nastro Finale
6/5	?
13/5	IT
11/30	UK
12/12	?

Esercizio 3 High tea [Punti 2].

E' il giorno! Prepariamoci per l'incoronazione di Re Carlo III del Regno Unito. Ovviamente, la prima cosa da preparare è il Tè.

Abbiamo preparato le tazze per gli ospiti, ognuna con la sua bustina di tè, ma dobbiamo stare attenti alla temperatura dell'acqua: il tè nero e il tè Earl Grey possono essere preparati con l'acqua fino a 90 gradi, mentre per il tè jasmine e il tè verde non dobbiamo superare gli 80! Per non rovinare il tè di nessuno, bolliremo l'acqua a 80 gradi se almeno una persona ha scelto il tè verde o il jasmine, e a 90 gradi altrimenti. Inoltre, se almeno una persona ha preso il tè nero, porteremo a tavola anche il latte.

Si scriva un programma per macchina di Turing che, ricevuta sul nastro la sequenza dei tè scelti (con N per tè nero, E per Earl Grey, V per tè verde, J per tè jasmine), termini lasciando sul nastro la temperatura scelta per bollire l'acqua, seguita da una L se va portato anche il latte.

Esempio

Nastro Iniziale	Nastro Finale
NNEVJ	80L
JJJVVV	80
EENNENN	90L
NNNENENEEEEENEEJNNEE	80L

Esercizio 4 Cucumber sandwiches [Punti 5].

Non si può avere il vero high tea senza *cucumber sandwiches*, piccoli panini farciti con cetrioli e altri companatici.

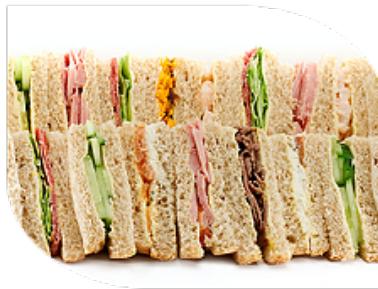
Abbiamo a disposizione 4 ingredienti: pane (P), burro (B), cetrioli (C), e salmone (S), e con questi sappiamo preparare due tipi di panino: il panino con burro e cetrioli (PBCP, ovvero burro e cetrioli, racchiusi tra due fette di pane) e il panino con salmone (PSP). Dati gli ingredienti, vogliamo preparare quanti più panini possibili: ad esempio avendo a disposizione gli ingredienti P PPPPBBC, siamo in grado di preparare un panino con burro e cetrioli, ma ci avanzano 3 fette di pane e una porzione di burro, che scarteremo in quanto non sono sufficienti a farne un altro.

Si scriva un programma per macchina di Turing che, ricevuta sul nastro la lista di ingredienti disponibili, termini lasciando sul nastro il maggior numero possibile di panini con cetrioli e burro, seguiti poi dal maggior numero possibile di panini col salmone che si possono preparare con gli ingredienti rimanenti.

Gli ingredienti sul nastro vengono sempre dati in ordine (prima P, poi B, poi C, poi S). Gli ingredienti non utilizzati vanno eliminati.

Esempio

Nastro Iniziale	Nastro Finale
PPPPBCC	PBCP
PPPPBCCSS	PBCPPSP
PPPPBCCSSS	PBCPPBCP
PPCC	
PPC	PSP



Esercizio 5 I gioielli della corona [Punti 4].

Per essere incoronato, Carlo deve indossare durante la cerimonia svariati artefatti, tra cui la *Corona di Sant'Edoardo* (CE), la *Corona Imperiale di Stato* (CS), lo *Scettro di Sant'Edoardo* (SE), e il *Globo del Sovrano* (GS) (dimentichiamoci per il momento degli altri, come l'antico *Cucchiaino della corona*).

E' nostro compito recuperare questi artefatti dai reali magazzini, se non si trovano abbiamo un bel problema!

Gli artefatti sono rappresentati sul nastro dalle stringhe sopra (rispettivamente CE, CS, SE, e GS), il nastro/magazzino può contenere spazio vuoto (rappresentato da =) o altri oggetti, sempre rappresentati da due lettere. Ogni oggetto ha sempre almeno uno spazio vuoto (=) sia a destra che sinistra.

Si scriva un programma per macchina di Turing che, ricevuto sul nastro il contenuto del magazzino, termini lasciando sul nastro GOOD se sono presenti *tutti e quattro* gli artefatti necessari, in qualsiasi ordine, e PANIC altrimenti.

Esempio

Nastro Iniziale	Nastro Finale
=GS=CS==SE===CE===	GOOD
=TY=CS=CU=GS==SE===CE===	GOOD
=CE=CU=AA==SE==TY==	PANIC
===	PANIC



Esercizio 6 Prendiamo posto [Punti 6].

La cerimonia sta per iniziare nell'abbazia di Westminster. Secondo il tableau dei posti, i partecipanti sono disposti per famiglia, con i genitori a sinistra ed eventuali bambini a destra.

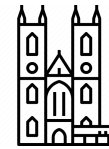
Rappresentiamo gli uomini con U, le donne con D e i bambini con B.

Dobbiamo contare le *famiglie numerose* per potergli prenotare alloggi adeguati. Definiamo una famiglia numerosa come due adulti (uomini o donne) seguiti da almeno due bambini. As es, DUBBBB, e UUBB sono famiglie numerose, ma non lo sono UBBB o UDB.

Si scriva un programma per macchina di Turing che, ricevuta sul nastro una sequenza di persone sedute, termini lasciando sul nastro solamente il numero di famiglie numerose. Si può assumere che il numero di famiglie numerose abbia al massimo 3 cifre.

Esempio

Nastro Iniziale	Nastro Finale
UDBBBBBUDUD	1
UUBBUDBBBUDBBUBB	3
UDBDUDBUBBB	0
UUBBUUBBUUBBUDBBUUBBUUBBDDBBUUBBDUBBUUBBB	11



Esercizio 7 La Successione [Punti 9].

E' bene avere le idee chiare su chi, dopo Carlo, debba essere il prossimo sovrano. In successione ha priorità la prole (figli e figlie) in ordine di nascita, e solo in mancanza di prole di considerano fratelli e sorelle, sempre in ordine di nascita. Quindi il figlio primogenito del Re (*William*) viene prima del secondo figlio (*Harry*), ma il secondo figlio viene comunque prima dei fratelli e sorelle del Re. I figli vengono dopo i propri genitori, in ordine di età, ma ereditano la priorità: se, ad esempio, William viene prima di suo fratello minore Harry, tutti i discendenti di William (figli, nipoti..) vengono dopo William ma prima del loro zio Harry.

Indichiamo i pretendenti al trono col loro nome e la linea di discendenza dal re, usando P per *prole* (figli e figlie) e F per *fratelli e sorelle*, oltre all'ordine di età: WILLIAM.P1 in dica che William è il primo figlio del Re, HARRY.P2 sarà invece il secondo figlio del Re, LOUIS.P1.P3 il terzo figlio (3P) del primo figlio (1P) del Re (quindi Louis è figlio di William), e BEATRICE.F1.P1 sarà la prima figlia (1P) del primo fratello (1F) del Re.

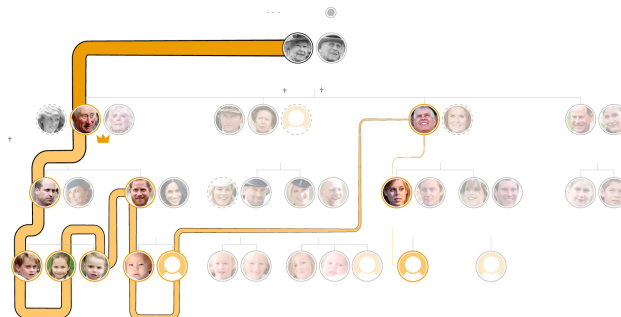
Con le regole sopra dette, avremo che LOUIS.P1.P3 viene prima di Harry.P2, In quanto Louis discende dal primo figlio del Re. Inoltre Archie.P2.P1 viene prima di Beatrice.F1.P1, In quanto Archie discende dal secondo figlio del Re (2P), mentre Beatrice discende dal primo fratello (1F). Si noti quindi che i gli elementi più a sinistra sono più significativi per la precedenza.

Si scriva un programma per macchina di Turing che, ricevuta sul nastro una sequenza di pretendenti al trono con discendenza, separati da #, termini lasciando sul nastro solamente il nome del primo tra loro in linea per la successione.

Per semplicità, si tenga conto che tutte le discendenze contengono al *massimo due generazioni* (ad es, LOUIS.P1.P3), e che nessuno nella famiglia reale ha più di 3 figli, o 3 fratelli e sorelle.

Esempio

Nastro Iniziale	Nastro Finale
HARRY.P2#LOUIS.P1.P3#ARCHIE.P2.P1	LOUIS
GEORGE.P1.P1#CHARLOTTE.P1.P2#LOUIS.P1.P3	GEORGE
ARCHIE.P2.P1#BEATRICE.F1.P1#LILIBET.P2.P2	ARCHIE
ANDREW.F1#WILLIAM.P1#HARRY.P2#GEORGE.P1.P1	WILLIAM



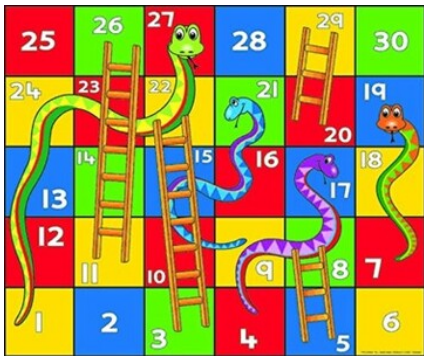
Esercizio 8 Linea di Successione [Punti 12].

Piuttosto che determinare solo il prossimo sovrano, è utile ordinare i pretendenti presenti dal primo all'ultimo in ordine di priorità nella successione: ad esempio, perché le portate a tavola vanno servite in questo ordine!

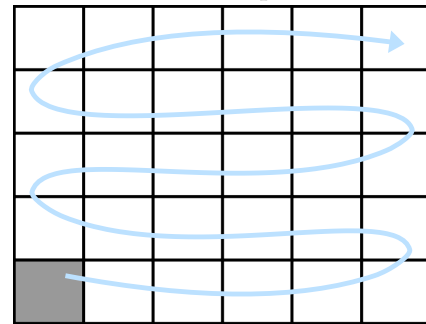
Si scriva un programma per macchina di Turing che, ricevuta sul nastro una sequenza di pretendenti al trono con discendenza, separati da #, termini lasciando sul nastro i nomi dei pretendenti in ordine di priorità nella successione (da sinistra a destra), separati da #.

Esempio

Nastro Iniziale	Nastro Finale
HARRY.P2#LOUIS.P1.P3#ARCHIE.P2.P1	LOUIS#HARRY#ARCHIE
GEORGE.P1.P1#CHARLOTTE.P1.P2#LOUIS.P1.P3	GEORGE#CHARLOTTE#LOUIS
ARCHIE.P2.P1#BEATRICE.F1.P1#LILIBET.P2.P2	ARCHIE#LILIBET#BEATRICE
ANDREW.F1#WILLIAM.P1#HARRY.P2#GEORGE.P1.P1	WILLIAM#GEORGE#HARRY#ANDREW



Direzione del percorso



Esercizio 9 Snakes and ladders [Punti 15].

La cerimonia è stata un successo, e ora vogliamo riposarci con un bel gioco da tavola. *Snakes and Ladders* è il classico gioco da tavola inglese, una variante del Gioco dell'Oca. Il giocatore ha una pedina (P) che parte dalla prima casella in basso a sinistra, e l'obiettivo è raggiungere l'ultima casella in alto a destra.

In ogni turno il giocatore lancia un dado e si muove di quel numero di caselle *in avanti*, seguendo i numeri come rappresentato in figura. Nella prima riga, in avanti è da sinistra a destra, ma arrivato in fondo ad una riga, la pedina si muove di una casella in alto e cambia direzione. Quindi la seconda riga viene percorsa da destra a sinistra, la terza nuovamente da sinistra a destra, e così via.

Ma il turno non finisce qui: il tabellone contiene serpenti (snakes) e scale (ladders). Se, una volta mossa la pedina di *tutti* i passi indicati dal dado, la pedina si ferma sul fondo di una scala, ovvero la sua casella più in basso, si arrampica fino alla sua cima! Se invece ha la sfortuna di fermarsi sulla testa di un serpente (ovvero la sua casella più in alto), scivola giù fino alla coda. Se la pedina si ferma su qualsiasi altra casella, come la metà di una scala o la coda di un serpente, o su una casella vuota, rimane semplicemente dov'è.

Si scriva un programma per macchina di Turing che, ricevuto sul nastro il tabellone di gioco con una singola pedina, preceduto dal punteggio ottenuto dal dado, termini lasciando sul nastro il tabellone con la pedina mossa adeguatamente. In caso la pedina termini sull'ultima casella (o oltre), si lasci sul nastro solamente la scritta WIN.

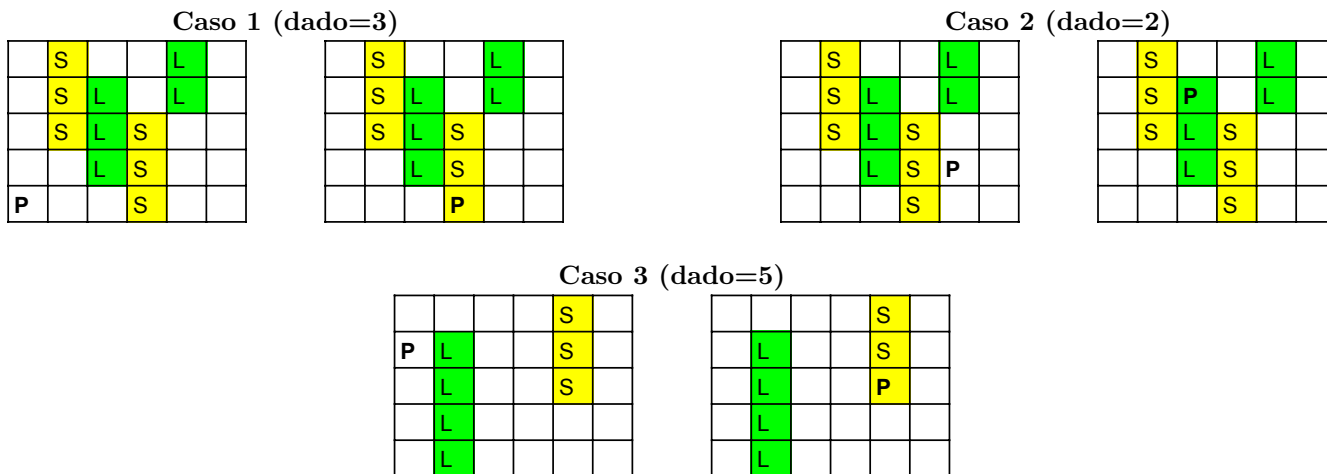
Formato del nastro: il tabellone è 6x5 come in figura, ovvero 5 righe da 6 caselle. Le righe sono messe sul nastro una ad una partendo da quella più in basso; per facilitare il riconoscimento delle direzioni, le righe sono separate o da "." (alla fine delle righe dispari) oppure da ":" (alla fine delle righe pari).

La pedina sarà rappresentata da P, i serpenti da S, le scale da L, e le caselle vuote da #. Per semplicità, tutti i serpenti e tutte le scale hanno lunghezza *almeno* 2, e non hanno curve (sono linee verticali). Assumiamo inoltre che la pedina parta sempre da una casella vuota.

Esempio

Nastro Iniziale	Nastro Finale	
3P##S##.##LS##:#SLS##.#SL#L#:#S##L#	###P##.##LS##:#SLS##.#SL#L#:#S##L#	Caso 1
2###S##.##LSP##:#SLS##.#SL#L#:#S##L#	###S##.##LS##:#SLS##.#SP#L#:#S##L#	Caso 2
5#L####.#L####:#L##S#.PL##S#:#S##S#	#L####.#L####:#L##P#.#L##S#:#S##S#	Caso 3
6#L####.#L####:#L##S#.L##S#:#P#S#	WIN	

Le immagini mostrano il tabellone nei primi tre casi (prima e dopo la mossa)



Esercizio 10 Si è incastrato il dado [Punti 18].

Stiamo giocando a *Snakes and Ladders*, ma per qualche ragione il dado continua a fare lo stesso numero... se continua così, quanti turni ci vorranno per arrivare al traguardo?

Si scriva un programma per macchina di Turing che, ricevuto sul nastro il tabellone di gioco con una singola pedina, preceduto dal punteggio ottenuto dal dado, termini lasciando sul nastro il numero di turni necessario a raggiungere (o superare) l'ultima casella, oppure la scritta LOOP se si rimane intrappolati in un ciclo infinito di mosse.

Esempio

Nastro Iniziale	Nastro Finale	
3P##S##.##LS##:#SLS##.#SL#L#:#S##L#	6	Caso 4
3###S##.##LSP#:#SLS##.#SL#L#:#S##L#	5	Caso 5
5#L####.#L####:#L##S#.PL##S#:#S##S#	4	Caso 6
1#L####.#L####:#L##S#.L##S#:#P#S#	LOOP	

Le immagini mostrano le mosse in tre dei casi in tabella (dopo la prima mossa la pedina sarà sul numero 1, e così via)

