

Venticinquesima 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 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, 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 [753 a.C.] La fondazione [Punti 2].

L’avimanzia, o oroscopia, consisteva nell’osservare il volo degli uccelli per trarne presagi. Remo e Romolo, da buoni fratelli, stanno osservando il volo degli uccelli, onde decidere chi fra loro dovrà fondare una nuova città. Ogni uccello visto è indicato da una lettera A (per avis), e chi avrà visto più uccelli sarà il fondatore.

Si scriva un programma per macchina di Turing che, ricevuto sul nastro il risultato dell’osservazione degli uccelli come una sequenza di A, quindi un simbolo X, poi una sequenza di A, termini lasciando sul nastro la scritta “REMVS” o “ROMVLVS” a seconda di quale dei due fratelli ha osservato più uccelli. Nella sequenza di osservazione, la sequenza di A prima della X denota gli uccelli osservati da Remo, mentre la sequenza di A dopo la X denota gli uccelli osservati da Romolo. In caso di parità, per ridurre al minimo il ricorso alla violenza, vince Romolo.

Esempio

Nastro Iniziale	Nastro Finale
AAAXAAAA	ROMVLVS
AAAAAXAAAA	REMVS
AAAAAAAAAXAAAAAAA	ROMVLVS



Esercizio 2 [49 a.C.] Alea iacta est [Punti 1].

Cesare deve infine decidere se passare il Rubicone alla testa delle sue legioni, e scatenare una guerra civile, o se tornare a Roma da solo in segno di sottomissione al Senato. Affetto da momentanea indecisione, decide di lanciare un paio di dadi e lasciare scegliere alla sorte. Ciascun dado ha sei facce, numerate da 1 a 6. Si scriva un programma per macchina di Turing che, ricevuto sul nastro l'esito del lancio dei due dadi, rappresentato da due cifre consecutive, lasci sul nastro il simbolo P (per *pax*) se la somma dei due punteggi è pari, e B (per *bellum*) se è dispari.

Esempio

Nastro Iniziale	Nastro Finale
36	B
62	P
11	P

Esercizio 3 [312 d.C.] La battaglia di Ponte Milvio [Punti 4].

Sul ponte Milvio si affrontano gli eserciti di Flavio Valerio Aurelio Costantino e Marco Aurelio Valerio Massenzio. Il ponte è stretto, e solo una unità per parte può combattere in un dato momento. Le truppe dei contendenti sono quindi schierate in colonna; la prima unità di una parte si scontra con la prima dell'altra parte, e l'unità più forte riesce a distruggere l'altra, pur riportando delle perdite pari al valore dell'unità avversaria. Quando un'unità viene distrutta, quelle successive avanzano e la prima prende il suo posto, continuando il combattimento. La battaglia continua finché uno dei due imperatori, inizialmente schierati ciascuno dietro il proprio esercito, non si trova in prima linea e viene ucciso; in questo caso la battaglia termina.

Rappresenteremo il ponte con il simbolo =, Costantino con C (a sinistra), Massenzio con M (a destra), e ogni unità con un simbolo fra 1 e 9, rappresentante la sua forza. Lo scontro fra due unità ha come risultato, come detto, la distruzione dell'unità più debole e la riduzione della forza di quella più forte (di una quantità pari alla forza della più debole); se due unità hanno esattamente la stessa forza, vengono distrutte entrambe. I due imperatori si considerano di forza 0. Ad esempio, la battaglia C92=31M si evolve inizialmente in C9=11M, poi C8=1M, quindi C7=M per terminare in C7=.

Si scriva un programma per macchina di Turing che, ricevuta sul nastro la configurazione iniziale degli eserciti ai due lati del ponte come descritto sopra, termini lasciando sul nastro la situazione alla fine della battaglia.

Esempi:

Esempio

Nastro Iniziale	Nastro Finale
C92=31M	C7=
C4498=166211M	C44=
C228=455M	=2M
C4881=94332M	=

Esercizio 4 Pronuncia restituta [Punti 5].

La pronuncia del Latino classico è vicina a quella dell'Italiano moderno, ma non esattamente la stessa. Semplificando, e ignorando le questioni legate alla quantità delle vocali, possiamo limitarci a considerare le seguenti trasformazioni:

- *C* si pronuncia *K*
- *GI* si pronuncia *GHI*
- *GE* si pronuncia *GHE*
- *V* si pronuncia *U*
- *AE* si pronuncia *E*

Si scriva un programma per macchina di Turing che, ricevuto sul nastro un testo in latino, lo trasformi come indicato sopra per esprimerne la pronuncia. Si assuma che le parole del testo siano fatte da lettere in *A, . . . , Z*, e siano separate da un singolo carattere spazio.

Esempio

Nastro Iniziale	Nastro Finale
AVE CAESAR	AUE KESAR
GALLIA OMNIA	GALLIA OMNIA
CARPE DIEM	KARPE DIEM
COGITO ERGO SUM	KOGHITO ERGO SUM

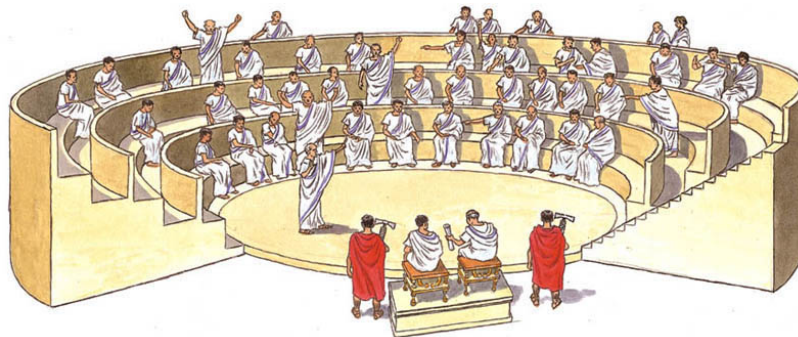
Esercizio 5 [219 a.C.] Sagunto brucia! [Punti 6].

Si vota in Senato se inviare aiuti a Sagunto, in Iberia, assediata dai cartaginesi di Annibale Barca. Si pone in votazione la proposta di inviare una legione. Ciascun senatore può votare a favore (che indicheremo con *S* per *sic*), contro (*N* per *non*), o astenersi (*A* per *abstineo*).

Si scriva un programma per macchina di Turing che, ricevuto sul nastro l'esito della votazione codificato da una lettera seguita dal numero di voti corrispondenti, lascia sul nastro il simbolo *L* (per *legio*) se il numero di voti a favore è maggiore della somma di contrari e astenuti, o *U* (per *uri*) in caso contrario. Si assuma che i risultati siano sempre dati nell'ordine favorevoli, contrari, astenuti e che il numero totale di senatori sia inferiore a 1000.

Esempio

Nastro Iniziale	Nastro Finale
S103N80A21	L
S103N80A25	U
S5N461A0	U
S50N0A50	U



Esercizio 6 Eleggiamo i consoli [Punti 7].

Ogni anno, i romani si riuniscono nei *comitia centuriata* per eleggere i due consoli dell'anno. I magistrati che presiedono all'elezione segnano, per ogni candidato, quante delle 193 centurie hanno votato per ciascun candidato. I due candidati che riportano più voti risultano eletti consoli. Ci sono sempre almeno due candidati. Si scriva un programma per macchina di Turing che, ricevuto sul nastro l'elenco dei candidati, ciascuno dei quali seguito da un punto e dal numero di voti raccolti (e un altro punto se segue un altro nome), lasci sul nastro i nomi dei due consoli eletti, separati da un punto, nell'ordine in cui comparivano nell'elenco iniziale. Per semplicità, assumiamo che non si presentino casi di parità, e che i candidati siano scritti sul nastro usando il solo *cognomen* (omettendo dunque il *praenomen* e la *gens*). I nomi dei candidati sono stringhe sull'alfabeto A-Z.

Esempio

Nastro Iniziale	Nastro Finale
SABINO.92.QUINTO.33.INCONSULTO.1.TUSCO.67	SABINO.TUSCO
CORNUTO.21.NASONE.22.RUTILIO.102.FUSO.40.CRASSO.8	RUTILIO.FUSO
MERENDA.85.VERRUCCOSO.0.DENTATO.108	MERENDA.DENTATO

Esercizio 7 Numeri romani [Punti 10].

Il sistema di numerazione romano è non-posizionale, a differenza di quelli a cui siamo abituati, e di tipo additivo e sottrattivo. Il valore non è quindi determinato dalla posizione di ciascuna cifra all'interno del numero, ma da operazioni di somma e sottrazione fra i diversi valori espressi dalle cifre.

I simboli usati sono i seguenti: I=1, V=5, X=10, L=50, C=100, D=500, M=1000.

Per i numeri più grandi venivano usati vari segni moltiplicatori (per 1.000, per 100.000, per 1.000.000 ecc.) che qui trascureremo. Il valore espresso da un numero è ottenuto sommando le singole cifre finché la cifra subito successiva è di valore inferiore o uguale a quella corrente, mentre in caso contrario si sottrae il valore della cifra da quello della cifra successiva. Le cifre I, X, C e M possono essere ripetute e dunque sommate fino a tre volte (poi si passa invece a usare la sottrazione dalla cifra successiva). Per esempio: III = 3, VI = 6, VIII=8, ma IX = 9, e XLV = 45. Si noti che da ogni cifra se ne può sottrarre al più un'altra (quindi il numero 8 si scrive VIII, e non si può scrivere IIX), e che si può sottrarre da una cifra solo quella immediatamente inferiore (quindi il numero 45 si scrive XLV, e non si può scrivere VL).

Si scriva un programma per macchina di Turing che, ricevuto sul nastro un numero romano fra 1 e 3000 (compresi), termini lasciando sul nastro il valore corrispondente espresso come un numero decimale intero (in notazione posizionale e cifre arabe).

Esempio

Nastro Iniziale	Nastro Finale
XXVI	26
IM	999
DCCXXVI	726
CMIII	903
MMC	2100
CXIX	119

Esercizio 8 Al banco del cambio [Punti 12].

Un sesterzio vale due assi e mezzo. Quattro sesterzi fanno un denaro; un aureo vale 25 denari. Al banco del cambio al mercato, chiedono 5 sesterzi per una dracma. Si scriva un programma per macchina di Turing che, ricevuto sul nastro un ammontare monetario espresso come una sequenza di simbolo di moneta e ammontare relativo, lasci sul nastro lo stesso ammontare espresso in dracme, approssimato per difetto. Useremo queste sigle per le monete: U=aureo, D=denaro, H=sesterzio, A=asse. Le monete il cui ammontare è 0 possono essere omesse, ma per semplicità assumiamo che quelle presenti saranno sul nastro nell'ordine indicato, che l'ammontare di assi sia sempre un multiplo di 2.5, e che la quantità di ogni moneta sia inferiore a 1000.

Esempio

Nastro Iniziale	Nastro Finale
U1D3H10A5	24
H5	1
D7A20	7
D20H120	40



Esercizio 9 [72 a.C.] Tasse dalle province [Punti 15].

Gaio Licinio Verre sta controllando gli incassi delle tasse della provincia di Sicilia. Le tasse pagate da ogni villaggio sono espresse da un numero in notazione romana, in sesterzi, e non superano mai i 100 sesterzi. L'elenco fornitogli da uno dei suoi questori include meno di 30 villaggi.

Si scriva un programma per macchina di Turing che, ricevuto sul nastro di ingresso l'elenco delle tasse pagate dai diversi villaggi, espressi in notazione romana e separati dal simbolo "+", lasci sul nastro l'importo totale riscosso, anch'esso in notazione romana.

Esempio

Nastro Iniziale	Nastro Finale
XLV+V	L
LXII+C+XCI+XXII	CCLXXV
XXI+XV+XLVI+IIII+XCVI+LXXVIII+C+XC	CDXLIX
I+I+I+I+I+I+I+I+I	X

Esercizio 10 Ludus duodecim scriptorum [Punti 25].

Il gioco detto *Ludus duodecim scriptorum* o anche semplicemente *XII scripta*, era un gioco popolare a Roma già dall'antichità, e un lontano antenato del moderno Backgammon. Lo si giocava su una *tabula* (in legno, marmo, o anche semplicemente un pezzo di pelle o stoffa) recante 6 parole di 6 lettere ciascuna, disposte su tre righe di due parole ciascuna. Le parole formavano spesso dei motti di spirito, o incitavano il giocatore a bere e mangiare le specialità della taverna, e così via. Di fatto, quali lettere fossero piazzate nello schema non era influente, e solo la griglia di 3x12 caselle era rilevante. Nel diagramma qui sotto, abbiamo indicato con numeri romani e arabi le caselle per semplificare la descrizione del gioco. Le caselle con numeri romani (in grigio) sono dette caselle d'entrata, mentre quelle con numeri arabi sono dette caselle di corsa.

	1	2	3	4	5	6		7	8	9	10	11	12	
B	VI	V	IV	III	II	I	↖	I	II	III	IV	V	VI	N
	24	23	22	21	20	19		18	17	16	15	14	13	

Il gioco prevede due giocatori, che chiameremo Bianco (B) a sinistra della *tabula*, e Nero (N) a destra, ciascuno dei quali dispone di 6 pedine identiche del colore corrispondente. Inizialmente tutte le pedine sono fuori dalla *tabula* (nella zona a fondo nero di ciascun giocatore). La partita si svolge a turni, e inizia il Bianco. Le pedine si muovono in modo diverso a seconda della zona della *tabula*. Quelle nelle caselle di entrata avanzano verso il centro, mentre quelle nelle caselle di corsa seguono l'ordine numerico (e dunque, compiono un percorso circolare). Quando una pedina, avanzando, raggiunge il centro delle caselle d'entrata (indicato nel diagramma con ↖), essa viene immediatamente portata sul numero 1 (senza che questo spostamento conti come una mossa). Ad ogni turno, il giocatore lancia due dadi a 6 facce. Il giocatore può decidere se usare i due valori dei dadi singolarmente (nel qual caso avrà due azioni) oppure sommarli e usare il valore totale (per una sola azione). Le azioni possibili sono:

1. mettere in gioco una pedina fuori *tabula* posizionandola nella casella di entrata del proprio lato corrispondente al valore (questo è possibile solo se il valore è al più 6);
2. avanzare una pedina già presente sulla *tabula* di un numero di caselle pari al valore.

Le mosse devono portare le pedine su caselle esistenti, in caso contrario non sono legali. Se un giocatore non ha nessuna azione possibile in base alla situazione sulla *tabula* e al tiro di dadi, passa il turno.

Ogni volta che una pedina ha raggiunto la propria casella di destinazione, possono verificarsi i seguenti casi:

1. La casella era vuota: viene occupata dalla pedina appena arrivata.
2. La casella era occupata da una o più pedine dello stesso colore: la pedina appena arrivata si aggiunge alla "colonna" così formata; per le mosse successive l'intera colonna si muove come una singola pedina.
3. La casella era occupata da pedine avversarie (singole o in colonna) in numero pari o inferiore al numero di pedine appena arrivate: tutte le pedine avversarie vengono tolte dalla *tabula* e rimesse nella zona di partenza del giocatore avversario (da cui potranno poi rientrare in gioco, come si è visto sopra).
4. La casella era occupata da pedine avversarie (singole o in colonna) in numero superiore al numero di pedine appena arrivate: tutte le pedine appena arrivate vengono tolte dalla *tabula* e rimesse nella zona di partenza del giocatore che ha mosso (da cui potranno poi rientrare in gioco, come si è visto sopra).
5. La casella di destinazione finale è la casella 24: le pedine del giocatore vengono tolte dalla *tabula* e considerate "salve" (queste pedine non possono rientrare in gioco).

La partita termina quando un giocatore ha salvato tutte le sue 6 pedine, vincendo così la partita.

Codifichiamo lo stato del gioco come segue: il numero di pedine Bianche su una casella sarà codificato con un simbolo fra 1 e 6; il numero di pedine Nere con un simbolo fra A e F (con l'ovvia convenzione che A=1, B=2 ecc.), una cella vuota sarà indicata dal simbolo #. Una configurazione del gioco sarà codificata da:

1. una cella contenente il simbolo B
2. una cella contenente un simbolo in 0, 1, ..., 6 che esprime le pedine del bianco fuori *tabula*

3. sei celle contenenti la situazione delle caselle di ingresso del bianco (da VI a I)
4. una cella contenente il simbolo “.”
5. 24 celle contenenti la situazione delle caselle di corsa in ordine numerico crescente (quindi, da 1 a 24)
6. una cella contenente il simbolo “.”
7. sei celle contenenti le situazione delle caselle di ingresso del nero (da I a VI)
8. una cella contenente un simbolo in, A, . . . , F che esprime le pedine del nero fuori tabula (o il simbolo Z se il nero ha zero pedine fuori tabula)
9. una cella contenente il simbolo N

Ad esempio, la configurazione iniziale del gioco (dove tutte le pedine del bianco e del nero sono fuori tabula) sarà codificata con la stringa `B6#####.#####.#####FN`

Si scriva un programma per macchina di Turing, che ricevuto su nastro di input due celle contenenti il tiro di dado (ogni dado espresso con un simbolo fra 1 e 6) immediatamente seguite da due configurazioni come descritte sopra, lasci sul nastro `SI` se la seconda configurazione rappresenta una mossa legale del Bianco partendo dalla prima (dati i valori dei dadi), e `NO` in caso contrario. Si assuma che entrambe le configurazioni siano di per sé legali (per esempio: non ci saranno più di 6 pedine di ogni colore).

Esempi: Sotto gli esempi, rappresentiamo graficamente la tabula e le pedine, per chiarezza, nello stesso ordine in cui appaiono nella tabella. La numerazione delle celle negli esempi grafici serve solo come riferimento; sul nastro avremo invece dei simboli # per le celle vuote, come indicato.

Esempio

Nastro Iniziale	Nastro Finale
<code>52B4##1##1.##A#####.#####ENB4#####.11A#####.#####EN</code>	SI
<code>52B4##1##1.##A#####.#####ENB4#####.##A1#####.#####EN</code>	SI
<code>52B4##1##1.##A#####.#####ENB31####1.##A1#####.#####EN</code>	NO
<code>33B0#####1.##1###A##3#####B###A####.#####BNB0#####1.##1###A#####3###A####.#####DN</code>	SI
<code>33B0#####1.##1###A##3#####B###A####.#####BNB0#####1.#####1##3#####B###A####.#####CN</code>	NO

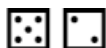
Caso 1 - risposta SI



	1	2	A	4	5	6		7	8	9	10	11	12	
B	VI	V	1	III	II	1	.	I	II	III	IV	V	VI	N
4	24	23	22	21	20	19		18	17	16	15	14	13	E

	1	1	A	4	5	6		7	8	9	10	11	12	
B	VI	V	IV	III	II	I	.	I	II	III	IV	V	VI	N
4	24	23	22	21	20	19		18	17	16	15	14	13	E

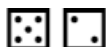
Caso 2 - risposta SI



	1	2	A	4	5	6		7	8	9	10	11	12	
B	VI	V	1	III	II	1	↖	I	II	III	IV	V	VI	N
4	24	23	22	21	20	19		18	17	16	15	14	13	E

	1	2	A	1	5	6		7	8	9	10	11	12	
B	VI	V	IV	III	II	1	↖	I	II	III	IV	V	VI	N
4	24	23	22	21	20	19		18	17	16	15	14	13	E

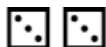
Caso 3 - risposta NO



	1	2	A	4	5	6		7	8	9	10	11	12	
B	VI	V	1	III	II	1	↖	I	II	III	IV	V	VI	N
4	24	23	22	21	20	19		18	17	16	15	14	13	E

	1	2	A	1	5	6		7	8	9	10	11	12	
B	1	V	IV	III	II	1	↖	I	II	III	IV	V	VI	N
3	24	23	22	21	20	19		18	17	16	15	14	13	E

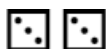
Caso 4 - risposta SI



	1	2	1	4	5	6		A	8	9	3	11	12	
B	VI	V	IV	III	II	1	↖	I	II	III	IV	V	VI	N
0	24	23	22	21	A	19		18	17	B	15	14	13	B

	1	2	1	4	5	6		A	8	9	10	11	12	
B	VI	V	IV	III	II	1	↖	I	II	III	IV	V	VI	N
0	24	23	22	21	A	19		18	17	3	15	14	13	D

Caso 5 - risposta NO



	1	2	1	4	5	6		A	8	9	3	11	12	
B	VI	V	IV	III	II	1	↖	I	II	III	IV	V	VI	N
0	24	23	22	21	A	19		18	17	B	15	14	13	B

	1	2	3	4	5	6		1	8	9	3	11	12	
B	VI	V	IV	III	II	1	↖	I	II	III	IV	V	VI	N
0	24	23	22	21	A	19		18	17	B	15	14	13	C