

TestiGara - QuattordicesimaEdizione

Esercizi di Gara della XIV edizione

AVVISI:

- Se non specificato altrimenti negli esercizi, 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.

Esercizio 1: Numeri pari [Punti 3]. Si scriva un programma per macchina di Turing che, ricevuto in ingresso sul nastro un numero decimale lasci sul nastro “P” se il numero era pari, “D” se era dispari.

NASTRO INIZIALE NASTRO FINALE

| | |
|------|---|
| 16 | P |
| 137 | D |
| 0 | P |
| 5173 | D |
| 2 | P |

Esercizio 2: ZUDTQCSSON [Punti 4]. Il Sistema Sbilenzo di Numerazione (SSN) prevede che ogni cifra venga indicata con l'iniziale del suo nome in Italiano. Quindi, 0=Z, 1=U, 2=D, ecc. È semplice trasformare un numero decimale in un codice SSN, ma può essere complicato, dato un codice SSN, ricostruire il numero originale. Si scriva un programma per macchina di Turing che, dato un codice SSN (senza Z iniziali), lasci sul nastro il numero decimale corrispondente, se unico, oppure “X” se non è possibile ricostruire il numero.

NASTRO INIZIALE NASTRO FINALE

| | |
|-------|-------|
| DTZCO | 23058 |
| CCSU | X |
| CZQDN | 50429 |
| NOSSN | X |

Esercizio 3: Modulo tre [Punti 5]. Si scriva un programma per macchina di Turing che, ricevuto in ingresso sul nastro un numero decimale lasci sul nastro il valore del numero modulo tre (ovvero, il resto della divisione per tre).

NASTRO INIZIALE NASTRO FINALE

| | |
|----|---|
| 30 | 0 |
| 7 | 1 |

| | |
|------------|---|
| 29 | 2 |
| 5456423321 | 2 |

Esercizio 4: Numeri ascendenti [Punti 6]. Un numero si dice ascendente se ogni sua cifra ha un valore strettamente superiore a quello di ciascuna delle cifre che lo precedono. Per esempio, 368 è un numero ascendente (infatti, $3 < 6 < 8$). Si scriva un programma per macchina di Turing che, ricevuto in ingresso sul nastro un numero decimale, lasci sul nastro “A” se il numero era ascendente, “N” in caso contrario.

NASTRO INIZIALE NASTRO FINALE

| | |
|--------|---|
| 368 | A |
| 365 | N |
| 25789 | A |
| 204678 | N |
| 6 | A |

Esercizio 5: Numeri discendenti [Punti 7]. Similmente alla definizione data all'esercizio precedente, un numero si definisce discendente se ciascuna delle sue cifre è strettamente minore delle precedenti. Per esempio, 8652 è un numero discendente. Si scriva un programma per macchina di Turing che, ricevuto in ingresso sul nastro un numero decimale maggiore o uguale a 10, lasci sul nastro “A” se il numero era ascendente, “D” se era discendente, “V” altrimenti.

NASTRO INIZIALE NASTRO FINALE

| | |
|-----------|---|
| 550 | V |
| 2359 | A |
| 43 | D |
| 10 | D |
| 2543273 | V |
| 123456789 | A |

Esercizio 6: La Sequenza Generalizzata Essaria di Fibonacci [Punti 12]. La *Sequenza Generalizzata di Fibonacci* $SGF(n,m)$ è la sequenza di numeri generati sommando i due precedenti elementi della sequenza, assumendo che i primi due elementi siano n e m . Avremo quindi $SGF_1(n,m)=n$, $SGF_2(n,m)=m$, $SGF_3(n,m)=n+m$, $SGF_4(n,m)=m+(n+m)$, $SGF_5(n,m)=(n+m)+(m+(n+m))$, ecc. La normale sequenza di Fibonacci è uguale a $SGF(1,1)$, e genera i valori 1, 1, 2, 3, 5, 8, 13, 21, ecc.

La *notazione essaria* consiste nel denotare un numero non con la consueta notazione posizionale, ma semplicemente indicando il simbolo S tante volte quant'è il valore del numero. Per esempio, la stringa SSSSS denota il numero 5, mentre SS rappresenta il numero 2.

Si scriva un programma per Macchina di Turing che, ricevuta su nastro una sequenza non vuota di numeri positivi in notazione essaria, separati da un singolo simbolo N, lasci sul nastro il simbolo T se la sequenza descrive una sottosequenza di una qualche SGF , o F in caso contrario.

NASTRO INIZIALE NASTRO FINALE

| | |
|-------------------|---|
| SSNSSNSSSS | T |
| SSNSSNSSSSNSSSSSS | T |
| SNSSSNSNSSS | F |
| SNSNSSNSSSNSSSSS | T |
| SSSNS | T |
| SSSNSNSSSSSS | F |

S
SNSNSNS

T
F

Esercizio 7: Steganografia [Punti 14]. La steganografia è una tecnica per trasmettere un messaggio (anche non cifrato) nascondendolo all'osservatore non autorizzato “confondendolo” all'interno di altra informazione. Per esempio, si può inviare una lunga missiva, con la convenzione che solo la quinta lettera di ogni paragrafo è parte del messaggio “vero”, mentre tutto il resto è solo materiale di copertura, destinato a confondere l'avversario-spione. Per questo esercizio prenderemo in esame una forma semplice di steganografia, consistente nell'inviare una stringa quasi-simmetrica (intendendo che normalmente su un input di dimensione n , il simbolo alla posizione k -esima è uguale al simbolo in posizione $(n-k+1)$ -esima), con l'intesa che solo i punti in cui la stringa non è simmetrica costituiscono il messaggio “vero”.

Si scriva quindi un programma per macchina di Turing che, ricevuta sul nastro una stringa quasi-simmetrica sull'alfabeto A-Z, contenente un messaggio steganografico dato dalle sole lettere non-simmetriche nella sequenza, lasci sul nastro il messaggio decodificato. In caso di stringa di lunghezza dispari, il simbolo centrale si considera facente parte del messaggio da recuperare.

| <i>NASTRO INIZIALE</i> | <i>NASTRO FINALE</i> |
|------------------------|----------------------|
| APBBEACCARBBOA | PERO |
| ZECCZMAAMEMCAZ | ECZEMA |
| A | A |
| DAMMADALADTAMAD | MALTA |
| ORTIRITATATTAIATROITRO | RITIRO |
| ANODITATACACIDONO | ATTACCO |

Esercizio 8: I conti del SSN [Punti 16]. Come si può comprendere facilmente, il SSN (introdotto nell'Esercizio 2) ha grossi problemi a far tornare i conti. Si scriva un programma per macchina di Turing che, ricevuta sul nastro un'espressione nella forma $n+m$, in cui n e m sono numeri in notazione SSN (senza limiti di valore), lasci sul nastro l'espressione $n+m=r$ (in cui r è il risultato della somma fra n e m espresso in SSN), se è possibile determinarlo univocamente, oppure il simbolo “?” in caso contrario.

| <i>NASTRO INIZIALE</i> | <i>NASTRO FINALE</i> |
|------------------------|----------------------|
| C+T | C+T=O |
| C+U | C+U=S |
| C+D | C+D=S |
| USS+Z | USS+Z=USS |
| USS+U | USS+U=? |
| SZT+DU | SZT+DU=SDQ |
| SZDUT+SQCQ | SZDUT+SQCQ=SSSSS |
| SUDUT+SQCS | SUDUT+SQCS=? |

Esercizio 9: I conti del SSN – parte 2 [Punti 18]. Si scriva un programma per macchina di Turing che, ricevuta sul nastro un'espressione nella forma $n+m=r$ in cui n , m e r sono numeri di esattamente tre cifre espressi in SSN, e con la garanzia che la somma sia corretta, lasci sul nastro la stessa espressione con i numeri in notazione decimale, se è possibile ricavarli, oppure “X” in caso contrario.

| <i>NASTRO INIZIALE</i> | <i>NASTRO FINALE</i> |
|------------------------|----------------------|
| UDT+TTD=QCC | 123+332=455 |
| QTS+UZD=CTO | 436+102=538 |
| UUU+CCC=SSS | 111+555=666 |

UUS+SSU=SSS 116+661=777
 UUS+UUS=DTT X (potrebbe essere 116+117=233 o 117+116=233)
 DZS+UUZ=TUS X (potrebbe essere 206+110=316 o 207+110=317)
 DZS+UUU=TUO 207+111=318

Esercizio 10: Numeri appaSSioNanti [Punti 25]. Un numero in notazione SSN è detto appaSSioNante se è divisibile per S, ovvero se il resto della divisione per S è 0 – per una qualche interpretazione di S. In altri termini, sono appaSSioNanti tutti i numeri SSN che hanno *almeno* una interpretazione decimale che è divisibile per 6 *oppure* per 7. Per esempio, USO è appaSSioNante (si può interpretare come 168, che è divisibile per 6 e per 7), come anche SO (interpretabile come 78, che è divisibile per 6) o UZZU (1001, divisibile per 7). Non sono invece appaSSioNanti NTC (935, non divisibile per 6 né per 7) o USZZ (interpretabile come 1600 o come 1700, nessuno dei quali è divisibile né per 6, né per 7).

Si scriva un programma per Macchina di Turing che, ricevuto in input un numero appaSSioNante in notazione SSN (di cui è quindi garantita la divisibilità per S), lasci sul nastro la sua interpretazione decimale di valore minore (che dovrà ancora essere divisibile per S).

| NASTRO INIZIALE | NASTRO FINALE | COMMENTO |
|--------------------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| UZZU | 1001 | Interpretazione univoca. |
| S | 6 | Si può interpretare come 6 o come 7, entrambi divisibili per S, 6 è l'interpretazione minore. |
| QSD | 462 | Si può interpretare come 462 (divisibile per 6 e per 7) o 472 (non divisibile per 6 o 7, quindi non appaSSioNante). |
| CSS | 567 | Si può interpretare come 566, 567, 576, 577. Di questi, 566 e 577 non sono appaSSioNanti; fra 567 e 576 che invece lo sono, il minore è 567. |
| USSSC | 16765 | Si può interpretare come 16665, 16675, 16765, 16775, 17665, 17675, 17765, 17775. Fra questi, solo 16765 e 17675 sono appaSSioNanti, e 16765 è il minore. |
| UOZQSS | 1804467 | Si può interpretare come 1804466, 1804467, 1804476, 1804477. Fra questi, solo 1804467 e 1804476 sono appaSSioNanti, e 1804467 è il minore. |