

Tutorato 03

Alla ricerca del ladro

Prigione Luca

30/10/XXXX

Note

- Potete usare solo la libreria standard di input/output quindi `stdio.h` in C oppure `iostream` in C++.
- Poichè il ladro vi ha rubato il computer non potrete usare l'interfaccia grafica per creare i file, ma dovrete utilizzare il terminale.

La bambina del vicolo

Lele si inoltrò nei vicoli stretti e fumosi dei quartieri bassi in cerca del mercante. Le strade erano un labirinto di voci e sguardi sospettosi. All'improvviso, una voce squillante lo fermò.

«Ehi, signore! Vuoi giocare con me?»

Davanti a lui c'era una bambina sui dieci anni con un sorriso furbo stampato sul viso. Lele scosse la testa, cercando di andarsene.

«Mi dispiace, piccola, non ho tempo per i giochi.»

«Ah no?» ribatté lei, socchiudendo gli occhi. «Peccato... pensavo ti interessasse sapere dov'è finito Fad»

Lele si fermò di colpo. «Tu sai dov'è?»

«Forse sì... ma prima devi battermi al mio gioco.»

Sospirando, Lele accettò la sfida e Livia sorrise.

Dovete scrivere un programma che tramite l'antica arte della ricorsione riesca a restituire il numero di elementi più grandi della somma dei due elementi che li precedono.

P.S.: Se non avete idea di come si faccia la ricorsione potete barare utilizzando l'iterazione

La rivincita

Livia, a corto di parole dopo la sconfitta, strinse i pugni e fece una smorfia. «Non vale! È solo fortuna!» sbottò.

Lele sorrise divertito. «O magari un po' di logica.»

«Tsk... allora rifacciamo! Al meglio delle tre!»

Prima che Lele potesse rispondere, la bambina aveva già tirato fuori due tavolette di legno, su cui erano incise due file di numeri ordinati.

«Adesso facciamo il Gioco delle Correnti, nessuno qui è mai riuscito a battermi!»

Lele sospirò, rendendosi conto che non aveva molta scelta.

«Va bene, piccola. Vediamo di cosa si tratta.» Per superare questa sfida dovete nuovamente utilizzare l'antica arte della ricorsione per risolvere il seguente problema: vi vengono forniti due array ordinati, il vostro obiettivo è quello di generare un terzo array che contenga il contenuto dei due array precedenti in modo ordinato

P.S.: Se non avete idea di come si faccia la ricorsione potete barare utilizzando l'iterazione

Sfida finale

Livia fissò il risultato, le guance rosse di rabbia. «Non è possibile! Hai barato!» gridò.

Lele rise. «Io? Ho solo seguito le regole.»

«Allora cambiamo gioco!» sbottò lei, tirando fuori di nuovo le tavolette. Lele notò subito che i numeri non erano più in ordine. «Aspetta... non dovevano essere ordinati?» «Nei vicoli si gioca così!» replicò Livia con un sorrisetto furbo.

Lele sospirò. «Va bene, siamo pari. C'è però ancora una sfida.» «Giusto!» esclamò lei. «Il mio preferito: il Gioco delle Parole Intrecciate!»

Prese due pergamene. «Devi sommare le parole lettera per lettera. Se una è più lunga, i caratteri in più si copiano così come sono.»

Create un programma per aiutare Lele a risolvere questo enigma, per fare ciò dovete creare una funzione che prende due caratteri passati per puntatore e restituisce il risultato tramite una variabile passata per riferimento.

La firma della funzione deve dunque essere la seguente:

*voidsumm(char * primo, char * secondo, char&risultato)*

La pista del mercante

Livia, ancora rossa per la sconfitta, borbottò: «Va bene... ti dirò dove si trova Fad.»

«Dove?» chiese Lele, curioso.

«Nel vicolo dei giochi, dietro la taverna del Lupo Grigio. Sta giocando al Gioco delle Tre Carte!»

Lele corse verso il vicolo. Fad rideva alla sua bancarella, muovendo rapidamente le carte. Senza esitazione, Lele si preparò alla sfida.

Per affrontare il Gioco delle Tre Carte e battere dunque il mercante dovrete implementare alcune funzioni:

- Swap: scambia due carte passate per puntatore senza variabili di supporto.

*voidSwap(char * carta_uno, char * carta_due)*

- Move: dato lo stato delle carte e un numero casuale, esegue uno swap e verifica l'esito.

voidMove(char&carta_destra, char&carta_centrale, char&carta_sinistra)

- Round: esegue più Move e chiede di indovinare la posizione della regina.

boolRound()

- Game: gestisce più round e calcola quante volte si ha vinto.

intGame(intnumero_round)

P.S.: Per rappresentare le carte potete utilizzare il carattere 'Q' per la carta vincente e '8' per le carte perdenti

P.P.S.: Il numero di partite che si vuole fare lo si può prendere in input

Esercizi aggiuntivi

Se avete già finito e volete qualche esercizio in più chiedete al Tutor, così potete provare a testare il vostro livello.