

Esercizio 1.

Si scriva un programma che letta da input due matrici $N \times N$: A e B. Verifichi che A sia uno schema valido per il gioco del Sudoku e che B sia una sua soluzione. Si noti che per il sudoku N deve essere tale per cui è possibile costruire all'interno della matrice N settori di dimensione $K \times K$, dove $K = \sqrt{N}$.

Esercizio 2.

Si scriva un programma completo C++ che simuli il gioco con le carte napoletane "Asino" tra due giocatori. Le regole del gioco sono le seguenti:

- 1) Dal mazzo di carte (composto da 40 carte, 4 semi, 10 carte per ogni seme) viene scartato uno dei cavalli (la carta numero nove) e le rimanenti 39 carte vengono distribuite tra i due giocatori, un giocatore avrà quindi 20 carte in mano, mentre l'altro ne avrà 19.
- 2) Alla prima mano, i due giocatori scartano tutte le coppie di carte con lo stesso valore, ad esempio (due otto, due cinque, etc.), per cui dopo questa fase ogni giocatore si troverà con un numero di carte sicuramente inferiore di quello iniziale.
- 3) A questo punto inizia il gioco vero e proprio. A turno ogni giocatore prende **a caso** una carta tra quelle dell'altro e controlla se, dopo la presa, si è formata una nuova coppia (ad esempio, il giocatore ha in mano un due di bastoni e prende dall'altro giocatore un due di coppe). Se questo avviene, il giocatore scarta la coppia e si ritroverà con due carte in meno. Se invece, dopo la presa, non si forma una nuova coppia, il giocatore si ritroverà con un carta in più.
- 4) Il gioco termina quando uno dei due giocatori rimane senza carte in mano vincendo la partita. L'altro giocatore rimarrà invece con una sola carta in mano (un cavallo) e sarà perciò l'asino del gioco.

Suggerimenti: (i) Per la rappresentazione delle carte di ogni giocatore, si potrebbe utilizzare una matrice di valori booleani, in cui le righe sono i 4 semi mentre le colonne rappresentano i 10 possibili valori delle carte. Se nella matrice, nella posizione con un indice di riga i e colonna j c'è `TRUE` allora la carta numero j con seme i è presente tra le carte del giocatore, altrimenti non lo è

(ad esempio, supponendo di indicare con `C1` la matrice corrispondente alle carte del primo giocatore, se `C1[2][3] == TRUE` allora la carta 3 con seme 2 è presente tra le carte del primo giocatore).

(ii) Per controllare l'andamento del gioco si potrebbero tenere dei contatori per il numero di carte di ogni giocatore. (iii) Per la distribuzione delle carte tra i due giocatori si può utilizzare la funzione: `void distribuisci(bool C1[][10], int & cont1, bool C2[][10], int& cont2);`

che riempie la matrice delle carte di ogni giocatore ed imposta il contatore corrispondente.