

NOME:

COGNOME:

MATRICOLA:

Si vuole creare un piccolo programma per il gioco delle carte tipo Scala 40 con carte da ramino (4 semi: Cuori Fiori Picche Quadri 13 carte per seme numerate da 1 a 13). Ogni carta è rappresentata dal seme (un carattere – iniziale del seme) e dal numero. Le carte in mano ad ogni giocatore *vengono mantenute in una lista doppia ordinata per seme e quindi per numero*. Nella cartella 05-02-19 sono presenti i file tipo.h che definisce il tipo per la carta, liste.cc e liste.h che implementano le liste.

Testo problema	Fatto	Val.	Max
<b>Punto 1:</b> Creare un progetto (e il corrispondente makefile) per la preparazione del gioco. Il progetto include i moduli per l'implementazione delle liste ordinate di carte dove è richiesta l'implementazione di <i>tipo.cc</i> (compare deve consentire di mantenere le carte in mano ad ogni giocatore come specificato sopra). Inoltre, il progetto include il file <i>compito.cc</i> che deve contenere: <ul style="list-style-type: none"> <li>• <code>void pesca(lista&amp;)</code> che aggiunge alla lista ordinata delle carte in mano ad un giocatore una nuova carta. Il seme e il valore della carta vengono inseriti da tastiera.</li> <li>• La procedura <code>stampa(lista)</code>.</li> <li>• Un <code>main</code> che prepara il gioco per 2 giocatori chiedendo il numero di carte da assegnare ad ogni giocatore, richiamando <code>pesca</code> più volte per generare due liste di carte della lunghezza specificata e quindi <code>stampa</code> per stampare le due liste di carte risultati.</li> </ul>			20
<b>Punto 2:</b> Estendere <i>compito.cc</i> con la funzione di ricerca di una scala: <ul style="list-style-type: none"> <li>• <code>carta* scala(lista carte, int&amp; lunghezza)</code> cerca nella lista <code>carte</code> la prima scala che trova<sup>1</sup>, se esiste, NULL altrimenti. La funzione restituisce il puntatore alla prima carta della scala e la lunghezza. La scala deve avere <i>lunghezza massima</i> ovvero la carta successiva alla sequenza e quella precedente non devono far parte della scala.</li> <li>• Estendere il <code>main</code> affinché richiami la funzione <code>scala</code> per ognuno dei due giocatori e stampi la scala risultante.</li> </ul>			4
<b>Punto 2.a:</b> Estendere <i>compito.cc</i> con l'implementazione di una mano di gioco: <ul style="list-style-type: none"> <li>• aggiungere la funzione <code>int cala(lista&amp; carte)</code> che cerca la prima scala nella lista richiamando la funzione <code>scala</code>, aggiorna la lista delle carte del giocatore eliminando la scala trovata e restituisce il punteggio della scala che è stata calata (somma dei numeri delle carte).</li> <li>• Estendere il <code>main</code> affinché implementi due turni di gioco tenendo conto del punteggio accumulato da ogni giocatore. Ad ogni turno, ogni giocatore pesca una carta (procedura <code>pesca</code>) e cala le carte (funzione <code>cala</code>). Il programma al termine di ogni turno stampa le carte restanti in mano ad ogni giocatore e il punteggio totale. Se il giocatore non ha più carte stampa "Fine gioco" e il nome del vincitore (ovvero chi ha più punti).</li> </ul>			3
<b>Punto 3:</b> Estendere il file <i>compito.cc</i> affinché il giocatore possa calare la scala migliore che ha in mano. A tale scopo: <ul style="list-style-type: none"> <li>• aggiungere la funzione <code>carta* best_scala(lista carte, int&amp; lunghezza)</code> che usando la funzione <code>scala</code> trova la scala migliore nel mazzo ovvero la scala a punteggio più alto. La funzione può far uso di strutture dati a piacere.</li> <li>• Aggiornare il <code>main</code> e le altre funzioni di cui al punto 2 o 2.a affinché usi <code>best_scala</code> al posto di <code>scala</code>.</li> </ul>			5
<b>Voto</b>			32

<sup>1</sup> **Scala:** sequenza di almeno 3 carte dello stesso seme e consecutive – ad esempio 1Q – 2Q – 3Q – 4Q

NOME:

COGNOME:

MATRICOLA:

**NOTE**

La valutazione del codice prodotto avviene al termine della prova e punto per punto (dipendenze tra i punti: 1-2-2.a, 1-2-3). Per ogni punto completato, è fondamentale che il codice compili e rispetti le specifiche descritte nel punto stesso.

È VIETATO l'uso di variabili globali.

Si ricorda che è possibile consultare solo il materiale cartaceo/digitale del corso e i libri di testo consigliati.

Al termine della prova, creare un folder nella directory /tmp/esame/risultato etichettato con il numero di matricola. All'interno devono essere presenti tutti i file .h e .cc del progetto e il makefile. Per essere valutati, il makefile deve produrre un eseguibile funzionante.

**Dati di prova:**

6 carte per giocatore, il giocatore n° 1 pesca 12Q 2P 10Q 3P 11Q 4P, il giocatore n° 2 pesca 1F 6F 3F 5F 1C 4F

**Punto 1** stampa:

giocatore 1: 2P 3P 4P 10Q 11Q 12Q.

giocatore 2: 1C 1F 3F 4F 5F 6F

**Punto 2:** stampa:

giocatore 1: 2P 3P 4P

giocatore 2: 3F 4F 5F 6F

**Punto 2.a** prima mano:

giocatore 1: pesca 9Q, cala 2P 3P 4P, carte in mano: 9Q 10Q 11Q 12Q punteggio: 9

giocatore 2: pesca 2C, cala 3F 4F 5F 6F, carte in mano: 1C 2C 1F punteggio: 18

seconda mano:

giocatore 1: pesca 13Q, cala 9Q 10Q 11Q 12Q 13Q

Fine gioco!! Vince giocatore 1 con punteggio 64

Per effetto del **Punto 3**

giocatore 1 scala migliore è 10Q 11Q 12Q

immutato per giocatore 2