

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*. Vengono forniti i file tipo.h che definisce il tipo per la carta, liste.cc e liste.h che implementano le liste. Vengono inoltre forniti i file *g1.txt* e *g2.txt* che contengono le carte in mano al giocatore 1 e giocatore 2 (una carta per riga contenente numero e iniziale del seme).

Testo problema	Fatto	Val.	Max
<b>Punto 0 da AA 2019-2020 (doxygen + git):</b> Documentare il codice in modo opportuno, affinché la chiamata a doxygen generi la documentazione del proprio progetto nella cartella doc. Inoltre, è necessario fare il clone del repository indicato, creare un branch denominato con il vostro numero di matricola, spostarsi all'interno del branch, 1 commit per ogni punto svolto, e la consegna finale tramite push su repository git. Il push verrà abilitato dal docente pochi minuti prima del termine della prova. L'indirizzo del repository remoto è <a href="https://gitlab.com/bedogni/programmazione2-2024.git">https://gitlab.com/bedogni/programmazione2-2024.git</a>			3
<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 dove è richiesta l'implementazione di <i>tipo.cc</i> (compare deve restituire un valore <0 se il numero della prima carta precede il numero della seconda o, a parità di numero, se il seme della prima carta precede il seme della seconda). Inoltre, il progetto include il file <i>compito.cc</i> che deve contenere: <ul style="list-style-type: none"> <li>● void pesca(lista&amp;, int v, char s) che aggiunge alla lista in mano ad un giocatore una nuova carta con valore v e seme s.</li> <li>● La procedura stampa(lista, lista) che stampa le due liste di carte a coppie nella stessa posizione e, per ogni coppia, il messaggio &lt;carta prima lista&gt; "è maggiore/minore/uguale a" &lt;carta seconda lista&gt;.</li> <li>● Un main che prepara il gioco per 2 giocatori chiedendo il numero di carte da assegnare ad ogni giocatore, leggendo le carte pescate dai file <i>g1.txt</i> e <i>g2.txt</i> e richiamando pesca più volte per generare due liste di carte della lunghezza specificata e quindi stampa.</li> </ul>			20
<b>Punto 2.a:</b> Estendere <i>compito.cc</i> con la funzione di ricerca di tris: <ul style="list-style-type: none"> <li>● int* tris(lista carte) conta nella lista carte il numero di tris distinti<sup>1</sup> per valore e li restituisce in un vettore dinamico di dimensione 13.</li> <li>● Estendere il main affinché richiami la funzione tris per ognuno dei due giocatori e stampi il numero di tris risultanti.</li> </ul>			4

<sup>1</sup> **Tris:** 3 carte dello stesso seme. Ad esempio per il numero 3 C 3 Q 3 C costituiscono un tris. ATTENZIONE: si assuma che le carte possono avere lo stesso seme.

NOME:

COGNOME:

MATRICOLA:

<b>Punto 2.b:</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 un tris da calare sul tavolo, stampa il tris trovato e aggiorna la lista delle carte del giocatore eliminando il tris trovato e restituisce il punteggio del tris.</li> <li>● Estendere il main affinché implementi un turno di gioco tenendo conto del punteggio accumulato da ogni giocatore: ogni giocatore pesca una carta (procedura pesca con input da tastiera) e cala un tris carte (funzione cala) alla volta fintanto che ci sono tris. Il programma al termine del turno stampa 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>			5
<b>Punto 0 prima AA 2019-2020:</b> Estendere il file <i>compito.cc</i> affinché le carte vengano mantenute in mano ordinate. A tale scopo: <ul style="list-style-type: none"> <li>● aggiungere la funzione <code>void pesca_ordinato(lista&amp;&amp;, int v, char s)</code> che a differenza della funzione pesca mantiene le carte ordinate.</li> <li>● Aggiornare il main e le altre funzioni di cui al punto 1 affinché usi <code>pesca_ordinato</code> al posto di <code>pesca</code>.</li> </ul>			3
<b>Voto</b>			32

## NOTE

La valutazione del codice prodotto avviene al termine della prova e punto per punto (dipendenze tra i punti: 1-2.a, 1-2b). 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.

## Dati di prova:

**Punto 1** dato in input 12 carte stampa:

2 P minore di 6 P  
 2 Q minore di 6 Q  
 3 Q minore di 7 Q  
 3 P minore di 7 P  
 4 Q maggiore di 2 Q  
 3 F maggiore di 1 F  
 4 P minore di 6 P  
 3 C maggiore di 1 C  
 2 C minore di 6 C  
 4 P maggiore di 2 P  
 4 Q maggiore di 1 Q  
 3 C maggiore di 2 C

**Punto 2.a:** stampa:

giocatore 1: 0 1 1 1 0 0 0 0 0 0 0 0  
 giocatore 2: 1 1 0 0 0 1 0 0 0 0 0 0

Prova di laboratorio di Programmazione II del 17/07/24

NOME:

COGNOME:

MATRICOLA:

**Punto 2.b** esempio di mano:

giocatore 1: pesca 2 Q, cala 2Q 2P 2Q, 3Q 3P 3F, 4Q 4P 4P, punteggio: 27

giocatore 2: pesca 7C, cala 6P 6Q 6P, 7Q 7P 7C, 2Q 2P 2C, 1F 1C 1Q punteggio: 48