

NOME:

COGNOME:

MATRICOLA:

Si vuole realizzare una piccola parte di un sistema per la gestione di carte fedeltà. Ogni cliente ha un numero di carta registrato come intero, un nome e cognome separati da spazio e registrati in un'unica stringa di al più 40 caratteri e il numero totale di punti accumulati. Le carte fedeltà dei clienti vengono registrate in un binary search tree (BST) al fine di velocizzare la ricerca per codice (che è quindi la chiave).

Ogni giorno viene mantenuta traccia degli acquisti effettuati dai clienti, per ogni acquisto si registra il codice della carta e il totale punti accumulati con l'acquisto.

Nella cartella 18-06-22 sono presenti i file *carta.h*, *bst.h* e *bst.cc*.

Testo problema	Fatto	Val.	Max
<b>Punto 1:</b> Estendere il modulo “carta”( <i>carta.h- carta.cc</i> ): definire il tipo <code>tipo_inf</code> , contenente nome e cognome e totale punti, e implementare le relative primitive. Creare un progetto (e il corrispondente makefile) per l'acquisizione delle carte fedeltà da tastiera. Il progetto include il modulo “carta”, il modulo “bst” (dove sono già implementate le primitive per un bst con chiave <code>int</code> corrispondente al codice della carta) e il file <i>compito.cc</i> . Il file <i>compito.cc</i> deve contenere il codice per l'acquisizione di una sequenza di <i>n</i> carte, con <i>n</i> inserito da tastiera e dove i dati di ogni singola carta vengono inseriti riga per riga.			18
<b>Punto 2:</b> Estendere il modulo “bst” con la primitiva di stampa della sequenza di carte in ordine crescente di valore. Richiamare la primitiva dal <code>main</code> .			2
<b>Punto 2.b:</b> Estendere il file <i>compito.cc</i> aggiungendo la funzione <code>totalePunti</code> che dati in ingresso un bst e due numeri di carta, <i>inf</i> e <i>sup</i> , restituisce la somma dei punti accumulati dalle carte maggiori o uguali a <i>inf</i> e minori o uguali a <i>sup</i> . Richiamare la funzione dal <code>main</code> dopo aver richiesto all'utente i due estremi.			4
<b>Punto 3:</b> Estendere il file <i>compito.cc</i> aggiungendo la funzione/procedura <code>aggiorna</code> che dati un bst e i dati relativi ad un acquisto, aggiorna la carta nel bst aggiungendo i punti accumulati con l'acquisto, se esiste, mentre segnala errore se non esiste. Richiamare <code>aggiorna</code> dal <code>main</code> per un numero a piacere di acquisti (la sequenza termina quando l'utente inserisce un carattere specifico) e stampare il bst risultante.			3
<b>Punto 4:</b> Estendere il file <i>compito.cc</i> affinché al termine della fase di aggiornamento di cui al punto 3 venga stampato anche l'elenco delle carte che non hanno effettuato acquisti in giornata.			5
<b>Voto</b>			32

**NOTE**

La valutazione del codice prodotto avviene al termine della prova e punto per punto (dipendenze tra i punti: 1-2-3-4, 1-2.b). Per ogni punto completato, è fondamentale che il codice compili e rispetti le specifiche descritte nel punto stesso.

È VIETATO l'uso di `break` e 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.

NOME:

COGNOME:

MATRICOLA:

**Dati di prova:**

Inserendo i seguenti clienti

1236

Anna Viola

22

4412

Paolo Rossi

36

2628

Marco Neri

176

3036

Mario Bianchi

144

Il **Punto 2** deve stampare i dati di cui sopra in ordine crescente di carta.

Dati i seguenti acquisti

4412

34

1238

37

3036

22

Il **Punto 3** deve segnalare errore per il secondo acquisto e stampare le seguenti righe:

1236 Anna Viola 22

2628 Marco Neri 176

3036 Mario Bianchi 166

4412 Paolo Rossi 70

Dati in ingresso i codici di carta 2000 e 4000 e i dati di cui sopra il **Punto 2.b** deve stampare: 412

Il **Punto 4** deve stampare:

1236 Anna Viola 22

2628 Marco Neri 176