

# ESERCIZI di esame sulle liste

## ESERCIZIO 1

Data una lista di pazienti in attesa di essere operati, i cui elementi includono:

- codice del paziente
- urgenza dell'operazione (in un range  $[1,10]$ )
- tempo richiesto dall'operazione in minuti
- età del paziente

La lista è ordinata in base all'urgenza e a parità di urgenza in base all'età. Si risolvano i seguenti problemi:

1. Definire gli opportuni tipi di dato per rappresentare la lista.
2. Calcolare l'età media dei pazienti in attesa di essere operati.
3. Inserzione ordinata di un paziente in attesa di operazione data la lista delle operazioni, il codice e l'età del paziente, l'urgenza e il tempo richiesto dall'operazione;
4. Progettare una funzione ricorsiva che, presa in ingresso la lista dei pazienti, il tempo in cui è libera la sala operatoria, restituisca il codice del paziente con priorità più alta, che può essere operato nel tempo specificato, e cancelli l'elemento del paziente selezionato dalla lista.

## ESERCIZIO 2

Si vogliono rappresentare le prenotazioni in una palestra per una lezione di spinning con un numero  $K$  costante di bike a disposizione. Ogni prenotazione contiene il codice del cliente che ha fatto la prenotazione, la tipologia del cliente (abituale, nuovo e saltuario) e il suo numero di cellulare. Le prenotazioni sono collezionate ordinate in base all'ordine di arrivo della richiesta (finchè c'è una bike disponibile), cioè per i primi  $k$  elementi. Dal  $K + 1$ -esimo elemento le prenotazioni sono in lista d'attesa e viene data precedenza ai clienti nuovi, seguiti dai clienti abituali e successivamente dai clienti saltuari, all'interno di una data tipologia le prenotazioni sono memorizzate nell'ordine in cui sono pervenute.

- (i) Si definiscano i tipi di dato necessari per implementare in  $\mathbf{C}$  la rappresentazione indicata. Si identifichi con `Prenotazioni` il tipo di dato principale.

Si definiscano le seguenti operazioni mediante opportune procedure o funzioni.

- (ii) Data una lista di prenotazioni, un codice e una tipologia di cliente e un numero di cellulare, scrivere una funzione che inserisce la prenotazione del cliente nella lista restituendo true se il cliente ha una bike disponibile e false se il cliente è stato inserito in lista d'attesa.
- (iii) Data una lista di prenotazioni e un codice cliente scrivere una funzione che cancella il cliente dalla lista e restituisce 0 se non ci sono clienti in lista d'attesa o il numero di cellulare del primo cliente in lista d'attesa.

## ESERCIZIO 3

Data una lista di aspiranti impiegati di IKEA, i cui elementi includono:

- codice dell'aspirante impiegato
- punteggio del curriculum in un range  $[0,20]$

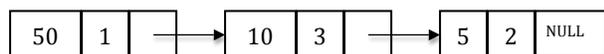
- il titolo di studio ('M' per Laurea magistrale, 'L' per laurea triennale, 'D' per diploma di scuola media superiore e 'I' per diploma di scuola media inferiore)
- età dell'aspirante impiegato

La lista è ordinata in base al punteggio del curriculum e a parità di punteggio in base all'età. Si risolvano i seguenti problemi:

1. Definire gli opportuni tipi di dato per rappresentare la lista.
2. Calcolare l'età media degli aspiranti impiegati
3. Inserzione ordinata di un aspirante impiegato data la lista degli aspiranti impiegati, il codice e l'età dell'aspirante impiegato, il punteggio e il titolo di studio.
4. Cancellazione di un aspirante impiegato dato il codice
5. Scrivere una funzione Assunzione che data la lista degli aspiranti impiegati e un intero  $k$  restituisce come risultato la lista degli assunti ovvero dei primi  $k$  elementi e modifica la lista degli aspiranti impiegati eliminando i primi  $k$  elementi.

#### ESERCIZIO 4

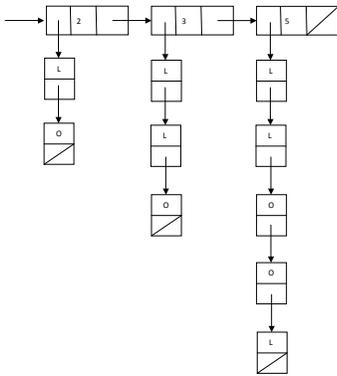
Si vuole rappresentare un portafoglio contenente banconote in euro dei vari tagli: 5 euro, 10 euro, 20 euro, 50 euro, 100 euro, 200 euro, 500 euro. Il portafoglio deve essere rappresentato con una lista in cui ogni elemento rappresenta le banconote di un determinato taglio, quindi ogni elemento della lista contiene informazioni per un taglio e il numero di banconote di quel taglio presenti nel portafoglio. Le banconote sono mantenute ordinate per taglio. Il portafoglio non deve contenere elementi che rappresentino tagli di banconote con 0 elementi. Ad esempio la lista riportata di seguito rappresenta un portafoglio contenente 1 banconota da 50, 3 banconote da 10 euro e 2 banconote da 5 euro, per un totale di 90 euro. Si noti che non sono presenti elementi che rappresentano i tagli da 20, 100, 200 e 500 euro perchè non ci sono banconote di tali tagli nel portafoglio.



- Definire i tipi opportuni per la rappresentazione indicata.
- Scrivere una funzione che data un puntatore ad un portafoglio calcoli quanti soldi sono contenuti nel portafoglio.
- Scrivere una funzione che dato un puntatore ad un portafoglio e un ammontare modifica il portafoglio eliminando le banconote che servono per pagare l'ammontare richiesto. Si assuma l'ammontare arrotondato a 5 euro.
- Scrivere una funzione *ricorsiva* che dato un puntatore ad un portafoglio e un taglio di banconota, inserisca la banconota nel portafoglio.

#### ESERCIZIO 5

Si vuole rappresentare un sistema per la prenotazione di posti a teatro. Vogliamo quindi avere una lista concatenata, dove ogni elemento rappresenta una fila, con il suo numero di posti liberi e la sua sottolista dei posti in dotazione. Ad ogni posto è associato un campo che ci dice se il posto è libero oppure no. Di seguito riportiamo un esempio.



- Definire i tipi opportuni per la rappresentazione indicata.
- Scrivere una funzione che data un puntatore ad una fila controlli che l'informazione sui posti liberi sia coerente con l'informazione contenuta nella sottolista.
- Scrivere una funzione *iterativa* che cerchi la prima fila, in cui si trovano  $n$  posti liberi contigui e che restituisca il puntatore al primo posto libero individuato, oppure `NULL`, in caso di insuccesso.
- Scrivere una procedura *ricorsiva* che, data la lista delle file, individui ed elimini tutte le file, i cui posti risultino tutti occupati.

## ESERCIZIO 6

Si vogliono rappresentare gli ordini di un fornitore di ciliegie. Ogni ordine contiene il codice del cliente che ha fatto l'ordine, il quantitativo espresso in kg di ciliegie prenotate e un campo che dice se il cliente è stato fin a quel momento in regola con i pagamenti degli ordini precedenti. Gli ordini sono collezionati in una lista nella quale gli ordini dei clienti in regola con i pagamenti precedono quelli dei clienti in ritardo sui pagamenti. A parità di situazione dei pagamenti poi gli ordini sono ordinati in base al codice cliente.

- Si definiscano i tipi di dato necessari per implementare in C la rappresentazione indicata. Si identifichi con `Ordin` il tipo di dato principale. Si definiscano le seguenti operazioni mediante opportune procedure o funzioni.
- Dato una lista di ordini, e un quantitativo (espresso in kg) di ciliegie disponibili, scrivere una funzione ricorsiva che evada, cancellandoli, tutti gli ordini che possono essere evasi per l'intero quantitativo richiesto rispettando l'ordine della lista. La funzione deve restituire il numero di kg di ciliegie che avanzano. Ad esempio se i quantitativi corrispondenti agli ordini della lista sono  $5 \rightarrow 7 \rightarrow 4 \rightarrow 6 \rightarrow 2$  e abbiamo 10 kg a disposizione la funzione deve restituire 1 e modificare la lista ottenendo  $7 \rightarrow 6 \rightarrow 2$ ;
- Data una lista di ordini, un codice cliente, un quantitativo ordinato e una situazione di pagamento, scrivere una procedura che inserisca il nuovo ordine nella lista;
- Data una lista di ordini, scrivere una funzione che controlli che la quantità di ciliegie ordinate dai clienti in regola con i pagamenti costituisca almeno i due terzi della quantità totale di ciliegie ordinate.

## ESERCIZIO 7

Si vogliono rappresentare i numeri civici di una strada. I numeri civici possono essere sia rossi che neri: puo' esistere il 10 nero ma anche il 10 rosso. Il comune mantiene una lista (ordinata per valore crescente) che contiene i numeri civici corrispondenti ad immobili che devono pagare la Service Tax, e che, per ogni numero civico, contiene anche un booleano che indica la presenza (o meno) di altri civici con lo stesso colore nel resto della lista. Un esempio di lista valida è

$(2, R, True) - > (3, N, False) - > (3, R, True) - > (4, R, True) - > (9, R, False) //$

L'informazione del valore booleano deve essere utilizzata per fermarsi, per esempio, dopo aver analizzato il secondo elemento, se cerco il 5  $N$ .

- (i) Definire i tipi opportuni per una rappresentazione con le caratteristiche indicate;
- (ii) Dato un colore (rosso o nero) definire una funzione che conti quanti numeri civici di quel colore sono presenti nella lista. Utilizzare tutta l'informazione disponibile per non scorrere (se non necessario) tutta la lista;
- (iii) Dato un colore (rosso o nero), definire una funzione che controlli che tutti i campi booleani relativi a quel colore siano corretti. La funzione deve scorrere la lista una sola volta;
- (iv) Definire una procedura *ricorsiva* che, dato un numero e un colore (rosso o nero), inserisca il nuovo civico nella lista, aggiornando le opportune informazioni.
- (v) Definire una funzione *ricorsiva* che, dato un numero e un colore (rosso o nero), cancelli (se esiste) il civico dalla lista, aggiornando anche le opportune informazioni.