



SECONDA PROVA IN ITINERE FONDAMENTI DI INFORMATICA

COMPITO 1

SEZIONE 1: PROVA AL CALCOLATORE (VALORE MAX 17 PUNTI)

Implementare una applicazione console in linguaggio ANSI C che permetta di gestire la prenotazione degli esami di un determinato insegnamento.

Le prenotazioni sono salvate in una **coda di prenotazioni** e possono essere relativi ad appelli differenti, i valori contenuti nella coda devono essere caricati da console e salvati su un file di testo preservando l'ordine di inserimento.

In un momento successivo l'applicazione carica i dati dal file in una **lista di code**, in cui ogni elemento della lista contiene la data dell'appello e la coda dei prenotati per quella data (lista di appelli).

Specifica della struttura dati:

Prenotazione:

1. *Numero matricola (matricola) implementato con una stringa, senza caratteri bianchi, che può contenere al più 15 caratteri utili;*
2. *Voto finale implementato con un intero (il valore -1 indica che non è stato assegnato alcun valore);*
3. *Data dell'appello (data) implementato con una stringa nel formato AAAA-MM-GG, senza caratteri bianchi, che può contenere al più 15 caratteri utili;*

Appello:

1. *Data dell'appello (data) implementato con una stringa nel formato AAAA-MM-GG, senza caratteri bianchi, che può contenere al più 15 caratteri utili;*
2. *Numero di studenti prenotati implementato con un intero;*
3. *Elenco degli studenti prenotati, implementato mediante una coda di prenotati.*

Elenco delle operazioni/funzioni che devono essere implementate:

1. Funzione di inserimento dei dati da console. La funzione da implementare deve avere la seguente struttura:

nome della funzione: *inserimento_prenotazione*;
valore restituito: *referimento all'elemento inserito*;
parametri: *coda delle prenotazioni, dati della prenotazione.*

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica;

2. Salvataggio dei dati presenti nella coda delle prenotazioni in un file di testo. La funzione da implementare deve avere la seguente struttura:

nome della funzione: *salva_prenotazioni*;
valore restituito: *numero di elementi salvati*;
parametri: *descrittore del file, coda delle prenotazioni.*

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica;

3. Funzione di caricamento dei dati da un file di testo nella struttura dati coda delle prenotazioni: La funzione da implementare deve avere la seguente struttura:

nome della funzione: *carica_coda_prenotazioni*;
valore restituito: *numero di elementi caricati*;
parametri: *descrittore del file, coda delle prenotazioni.*

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica;

4. Funzione di caricamento dei dati da un file di testo nella struttura dati lista di appelli. La funzione a partire dai dati presenti nel file crea una lista di appelli e la relativa coda di prenotazioni in base alla data dell'appello. Lo studente scelga la struttura della funzione, il tipo dei dati ed i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica
5. Funzione che calcola e fornisce come parametri il numero medio, minimo e massimo di prenotazioni per appello;
6. Funzione che dato il numero di matricola permette di assegnare il voto dell'esame allo studente. Nota bene che si suppone che lo studente possa avere inserito una sola prenotazione nell'intera struttura dati e che i dati necessari devono essere forniti alla funzione come parametri;
7. Funzione che stampa sulla console il contenuto dell'intera struttura dati;
8. Programma principale dotato di un menù testuale che permetta all'utente di usare tutte le funzioni implementate ed eventualmente di inserire i parametri necessari ed eseguire le operazioni richieste.

Esempio di file di salvataggio

```
046000111 -1 20170428
046000222 -1 20170428
046000333 -1 20170428
046000112 -1 20170426
046000223 -1 20170426
```



UNIVERSITÀ
degli STUDI
di CATANIA

DIPARTIMENTO di INGEGNERIA
ELETTRICA ELETTRONICA
e INFORMATICA

Corso di Laurea in
INGEGNERIA ELETTRONICA
INGEGNERIA INFORMATICA

SECONDA PROVA IN ITINERE FONDAMENTI DI INFORMATICA

SEZIONE 2: DOMANDE TEORICHE (VALORE MAX 3 PUNTI)

Domanda 1.

Si supponga di aver un vettore di interi contenente i seguenti valori

10, 3, 23, 5, 7, 8

Mostrare come cambia lo stato del vettore applicando un algoritmo di ordinamento di tipo bubble sort. Se il vettore fosse ordinato quante iterazioni verranno fatte?

Domanda 2.

In un albero binario di ricerca in quali nodi si trovano, rispettivamente il valore minimo ed il valore massimo. Scrivere il codice che trova il valore minimo ed il valore massimo contenuto in un albero binario di ricerca.



SECONDA PROVA IN ITINERE FONDAMENTI DI INFORMATICA

COMPITO 2

SEZIONE 1: PROVA AL CALCOLATORE (VALORE MAX 17 PUNTI)

Implementare una applicazione console in linguaggio ANSI C che permetta di gestire un parcheggio. All'ingresso di una autovettura, insieme alla data di ingresso, i dati vengono caricati in una lista ordinata in base alla targa dell'autovettura. Periodicamente i dati vengono trasferiti in una lista di giorni (ordinata rispetto al giorno), ognuno degli elementi, quindi, contiene la data e la lista ordinata delle vetture entrate in quella data.

Specifica della struttura dati:

Autovettura:

1. Numero di targa (targa) implementato con una stringa, senza caratteri bianchi, che può contenere al più 9 caratteri utili;
2. Data di ingresso implementata come un stringa con il seguente formato AAAA-MM-GG (ad esempio il 26 aprile 2017 è memorizzato come 2017-04-26);

Data:

1. Data di ingresso implementata come un stringa con il seguente formato AAAA-MM-GG (ad esempio il 26 aprile 2017 è memorizzato come 2017-04-26)
2. Elenco delle vetture presenti.

Elenco delle operazioni/funzioni che devono essere implementate:

1. Funzione di inserimento dei dati da console, i dati devono essere inseriti solo nella lista ordinata. La funzione da implementare deve avere la seguente struttura:

nome della funzione: *inserimento_autovettura*;
valore restituito: *numero attuale di vetture presenti*;
parametri: *lista delle autovetture, dati dell'autovettura compresa l'ora*.

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica;

2. Caricamento dei dati da un file di testo nella lista di autovetture. La funzione da implementare deve avere la seguente struttura:

nome della funzione: *carica_autovetture*
valore restituito: *nessuno*;
parametri: *nome del file, lista delle autovetture*.

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica;

3. Funzione di conteggio delle autovetture entrate in una data specifica (la funzione lavora sulla lista di autovetture). La funzione da implementare deve avere la seguente struttura:

nome della funzione: *conta_autovetture*;
valore restituito: *numero di elementi presenti nella data specificata*;
parametri: *lista delle autovetture, data*.

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica;

4. Funzione di spostamento dei dati dalla lista di autovetture alla lista di liste (lista di date in cui ogni elemento contiene le autovetture). La funzione a partire dai dati presenti nella lista di autovetture crea una lista di date e la relativa lista ordinata di vetture in base alla data di ingresso. Lo studente scelga la struttura della funzione, il tipo dei dati ed i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica
5. Funzione che calcola il numero totale di autovetture ed il numero totale di date presenti nella lista di liste ordinate;
6. Funzione che dato il numero di targa gestisce l'uscita della vettura. La funzione deve restituire **una copia** della struttura che contiene i dati della vettura e cancellarla dalla lista cui appartiene, il programma principale deve stampare la vettura;
7. Funzione che stampa sulla console il contenuto dell'intera struttura dati;
8. Programma principale dotato di un menù testuale che permetta all'utente di usare tutte le funzioni implementate ed eventualmente di inserire i parametri necessari ed eseguire le operazioni richieste.

Esempio di file di testo

AZ430AX
2017-04-26
AC456BN
2017-04-26
BN66790
2017-04-28
DC466CN
2017-04-26



UNIVERSITÀ
degli STUDI
di CATANIA

DIPARTIMENTO di INGEGNERIA
ELETTRICA ELETTRONICA
e INFORMATICA

Corso di Laurea in
INGEGNERIA ELETTRONICA
INGEGNERIA INFORMATICA

SECONDA PROVA IN ITINERE FONDAMENTI DI INFORMATICA

SEZIONE 2: DOMANDE TEORICHE (VALORE MAX 3 PUNTI)

Domanda 1

Dato un vettore ordinato di interi indicare qual'è il migliore algoritmo di ricerca applicabile. Se il vettore è composto da 16 valori indicare il numero di passi necessari a verificare che un elemento non è presente nel vettore.

Domanda 2

Scrivere il tipo di dato necessario a descrivere un albero binario di ricerca in cui ogni nodo contiene un valore intero e scrivere la funzione di visita simmetrica. Qual'è la caratteristica di una visita simmetrica in un albero binario di ricerca?



SECONDA PROVA IN ITINERE FONDAMENTI DI INFORMATICA

COMPITO 3

SEZIONE 1: PROVA AL CALCOLATORE (VALORE MAX 17 PUNTI)

Implementare una applicazione console in linguaggio ANSI C che permetta di gestire il piano di carico di un traghetto per il trasporto di automobili. Ogni nave organizza le auto nei ponti parcheggi disponibili in base all'ordine di arrivo delle autovetture, ogni ponte può contenere al massimo N vetture (porre N pari a 5 per semplificare il test del programma).

Le autovetture, al momento dell'arrivo, vengono inserite in una coda e successivamente viene preparato il piano di carico trasferendo le vetture nei ponti disponibili fino al loro riempimento, l'algoritmo prevede che le prime N(=5) vetture siano caricate nel ponte 0, le vetture dalla 6 alla 10 nel ponte 1 fino all'esaurimento delle vetture (si supponga che il numero di ponti parcheggio sia sempre sufficiente). La struttura dati che rappresenta il piano di carico è, quindi, una lista di ponti (ognuno con una coda di autovetture).

Specifica della struttura dati:

Autovettura:

1. Numero di targa (targa) implementato con una stringa, senza caratteri bianchi, che può contenere al più 9 caratteri utili;
2. Numero di passeggeri prenotati per la vettura, implementato come intero

Ponte:

1. Numero del ponte implementato come intero a partire dal valore 0;
2. Elenco delle vetture presenti, implementato mediante una coda di vetture.

Elenco delle operazioni/funzioni che devono essere implementate:

1. Funzione di inserimento dei dati da console nella coda. La funzione da implementare deve avere la seguente struttura:
nome della funzione: *inserimento_vettura*;
valore restituito: *riferimento all'elemento inserito*;
parametri: *coda delle vetture, dati della vettura*.

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica;

2. Conteggio del numero totale e del numero medio di passeggeri presenti. La funzione da implementare deve avere la seguente struttura:

nome della funzione: *conta_passeggeri*;
valore restituito: *nessuno*;
parametri: *coda delle vetture, numero totale dei passeggeri, numero medio dei passeggeri*.

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica;

3. Funzione di caricamento dei dati da un file di testo nella struttura dati coda delle vetture. La funzione da implementare deve avere la seguente struttura:

nome della funzione: *carica_coda_vetture*;
valore restituito: *nessuno*;
parametri: *nome del file, coda delle vetture*.

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica;

4. Funzione di caricamento dei dati dalla coda di attesa nei ponti di destinazione. La funzione, a partire dai dati presenti nella coda trasferisce la vettura nella coda del ponte corrente, al raggiungimento del numero massimo N di vetture per quel ponte, ne crea uno nuovo, assegna il numero corrispondente ed inserisce l'autovettura. Lo studente scelga la struttura della funzione, il tipo dei dati ed i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica
5. Funzione che dato il numero di targa delle vetture la elimina dalla lista di code;
6. Funzione che calcola e restituisce il numero medio di passeggeri presenti in ognuno dei ponti;
7. Funzione che stampa sulla console il contenuto dell'intera struttura dati;
8. Programma principale dotato di un menù testuale che permetta all'utente di usare tutte le funzioni implementate ed eventualmente di inserire i parametri necessari ed eseguire le operazioni richieste.

Esempio di file di salvataggio

```
AZ430AX
5
AC456BN
3
BN66790
2
DC466CN
8
```



UNIVERSITÀ
degli STUDI
di CATANIA

DIPARTIMENTO di INGEGNERIA
ELETTRICA ELETTRONICA
e INFORMATICA

Corso di Laurea in
INGEGNERIA ELETTRONICA
INGEGNERIA INFORMATICA

SECONDA PROVA IN ITINERE FONDAMENTI DI INFORMATICA

SEZIONE 2: DOMANDE TEORICHE (VALORE MAX 3 PUNTI)

Domanda 1

Dato un vettore di interi scrivere un algoritmo di ricerca sequenziale ed uno binario. Qual'è la complessità media dei due algoritmi? Un algoritmo di ricerca binaria può essere utilizzato su qualsiasi vettore?

Domanda 2

Dato un albero binario di ricerca definire il concetto di profondità. Scrivere un algoritmo che dato un valore (supponendo che questo sia presente) ne stampa la profondità



SECONDA PROVA IN ITINERE FONDAMENTI DI INFORMATICA

COMPITO 4

SEZIONE 1: PROVA AL CALCOLATORE (VALORE MAX 17 PUNTI)

Implementare una applicazione console in linguaggio ANSI C che permetta di gestire una emeroteca (*raccolta ordinata di giornali e riviste per consultazione e lettura, di solito annessa a una grande biblioteca*).

I dati relativi alle riviste vengono caricati in una lista ordinata rispetto ad un codice univoco (ISBN) ed organizzati successivamente in una lista ordinata di scaffali (ognuno con una lista ordinata delle riviste che ospita).

Specifica della struttura dati:

Rivista:

1. *Titolo (titolo) implementato con una stringa, senza caratteri bianchi, che può contenere al più 63 caratteri utili;*
2. *Codice ISBN (codice) implementato con una stringa, senza caratteri bianchi, che può contenere al più 15 caratteri utili;*
3. *Codice Scaffale (scaffale) implementato con una stringa, senza caratteri bianchi, che può contenere al più 15 caratteri utili;*
4. *Numero di pagine, implementato come intero*

Scaffale:

1. *Codice Scaffale (scaffale) implementato con una stringa, senza caratteri bianchi, che può contenere al più 15 caratteri utili;*
2. *Numero di riviste presenti;*
3. *Elenco delle riviste, implementato mediante una lista ordinata.*

Elenco delle operazioni/funzioni che devono essere implementate:

1. Funzione di inserimento dei dati da console. La funzione da implementare deve avere la seguente struttura:
nome della funzione: *inserimento_rivista*;
valore restituito: *riferimento all'elemento inserito*;
parametri: *lista delle riviste, dati della rivista*.

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica;

2. Salvataggio dei dati presenti nella lista delle riviste in un file di testo. La funzione da implementare deve avere la seguente struttura:
nome della funzione: *salva_rivista*;
valore restituito: *numero di elementi salvati*;
parametri: *descrittore del file, lista delle prenotazioni*.

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica;

3. Cancellazione di una rivista dato il codice ISBN. La funzione da implementare deve avere la seguente struttura:
nome della funzione: *cancellazione_rivista*;
valore restituito: *1 se l'elemento è presente, 0 in tutti gli altri casi*;
parametri: *lista delle riviste, codice ISBN*.

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica;

4. Funzione di copia dei dati dalla lista di riviste nella lista di scaffali (ognuno dei quali deve contenere la lista delle riviste in esso contenute). La funzione crea una lista di scaffali e le liste di riviste associate. Lo studente scelga la struttura della funzione, il tipo dei dati ed i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica
5. Funzione che calcola e fornisce come parametri il numero medio, minimo e massimo di riviste;
6. Funzione che dato il codice ISBN permette di modificare il titolo della rivista nella struttura dati. Nota bene che l'operazione deve essere implementata nella struttura dati lista di scaffali (lista di liste di riviste);
7. Funzione che stampa sulla console il contenuto dell'intera struttura dati;
8. Programma principale dotato di un menù testuale che permetta all'utente di usare tutte le funzioni implementate ed eventualmente di inserire i parametri necessari ed eseguire le operazioni richieste.

Esempio di file di salvataggio:

```
PrimoTitolo 0-4114-34B SCAFFALE01 123
SecondoTitolo 0-2314-34B SCAFFALE02 86
TerzoTitolo 0-8184-35B SCAFFALE01 234
QuartoTitolo 0-4214-88B SCAFFALE01 54
QuintoTitolo 0-4884-34B SCAFFALE02 66
```



UNIVERSITÀ
degli STUDI
di CATANIA

DIPARTIMENTO di INGEGNERIA
ELETTRICA ELETTRONICA
e INFORMATICA

Corso di Laurea in
INGEGNERIA ELETTRONICA
INGEGNERIA INFORMATICA

SECONDA PROVA IN ITINERE FONDAMENTI DI INFORMATICA

SEZIONE 2: DOMANDE TEORICHE (VALORE MAX 3 PUNTI)

Domanda 1

Si supponga di avere un vettore di interi contenente i seguenti valori

10, 3, 23, 5, 7, 8

Mostrare come cambia lo stato del vettore applicando un algoritmo di ordinamento di Insertion sort. Qual'è la complessità dell'algoritmo nel caso medio?

Domanda 2

Dato un albero binario di ricerca indicare quale visita (pre-ordine, post-ordine, simmetrica) stampa per primo il valore della radice. Supponendo di avere un albero binario di ricerca scrivere l'algoritmo che permette di stampare tutti i valori maggiori di un dato N.



SECONDA PROVA IN ITINERE FONDAMENTI DI INFORMATICA

COMPITO 5

SEZIONE 1: PROVA AL CALCOLATORE (VALORE MAX 17 PUNTI)

Implementare una applicazione console in linguaggio ANSI C che permetta di gestire la programmazione giornaliera dei canali TV. Le informazioni di ciascun programma sono memorizzate in un file (specificato nel seguito) e caricate in una **lista dei programmi** ordinata in base all'ora di inizio del programma. In un momento successivo, l'applicazione dovrà organizzare le informazioni di tali programmi in una **lista dei canali TV** in cui ogni elemento della lista contiene, oltre ad informazioni sul canale, anche l'elenco dei programmi di quel canale televisivo.

Specifica della struttura dati:

Programma:

1. *Titolo del programma* (stringa, di al più 255 caratteri)
2. *Nome del canale* (stringa, senza spazi, che può contenere al più 15 caratteri utili)
3. *Genere televisivo* (stringa senza spazi, che può contenere al più 15 caratteri utili)
4. *Ora inizio* (implementata con una stringa nel formato hh:mm)
5. *Durata* (intero, espressa in minuti)

CanaleTV:

1. *Nome del canale* (stringa, senza spazi, che può contenere al più 15 caratteri utili)
2. *Elenco dei programmi di quel canale*, (implementato mediante una lista di programmi)

Elenco delle operazioni/funzioni che devono essere implementate:

1. Funzione di caricamento da file di testo dei programmi televisivi nella lista dei programmi. La lista dei programmi deve essere mantenuta **ordinata in modo crescente in base all'orario di inizio del programma**. La funzione da implementare deve avere la seguente struttura:

nome della funzione: *CaricaProgrammi*;
valore restituito: *numero di programmi caricati*;
parametri: *nome del file, lista dei programmi*.

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.

2. Funzione di modifica dell'orario di inizio di un programma. La funzione da implementare deve avere la seguente struttura:
nome della funzione: *ModificaOrarioInizio*;
valore restituito: *esito operazione (0: operazione fallita, 1: modifica effettuata con successo)*;
parametri: *lista dei programmi, titolo programma, nuovo orario di inizio*.

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.

3. Funzione di eliminazione di un programma dalla lista. La funzione da implementare deve avere la seguente struttura:
nome della funzione: *EliminaProgramma*;
valore restituito: *esito operazione (0: operazione fallita, 1: eliminazione effettuata con successo)*;
parametri: *lista dei programmi, titolo programma, nome del canale televisivo*.
Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.
4. *CopiaProgrammiInListaCanali*: Funzione di copia delle informazioni contenute nella lista dei programmi nella lista dei canali TV, facendo in modo che ogni canale TV contenga soltanto i programmi di quel canale. Lo studente scelga la struttura della funzione, il tipo dei dati ed i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.
5. *DurataMedia*: Funzione che, fornito come parametro di ingresso un nome di canale, calcola e restituisce la durata media dei programmi televisivi di quel canale
6. *VisualizzaProgrammiSuccessivi*: Funzione che, fornito in ingresso un orario (nel formato hh:mm) visualizzi i dati di tutti i programmi televisivi il cui orario di inizio è successivo o uguale all'orario fornito in ingresso.
7. Funzione che visualizza il contenuto dell'intera lista di canali TV.
8. Programma principale dotato di un menù testuale che permetta all'utente di usare tutte le funzioni implementate ed eventualmente di inserire i parametri necessari ed eseguire le operazioni richieste.

Struttura del file dei programmi

Le informazioni nel file per ciascun programma si trovano distribuite su tre righe:

Prima riga: canale TV
Seconda riga: titolo del programma
Terza riga: genere televisivo, orario di inizio, durata

Esempio di file:

Rai1
Festival di Sanremo
Musicale 20:30 240



UNIVERSITÀ
degli STUDI
di CATANIA

DIPARTIMENTO di INGEGNERIA
ELETTRICA ELETTRONICA
e INFORMATICA

Corso di Laurea in
INGEGNERIA ELETTRONICA
INGEGNERIA INFORMATICA

SECONDA PROVA IN ITINERE FONDAMENTI DI INFORMATICA

Rai1
TG1
Notizie 13:30 30
Canale5
Striscia la notizia
Varieta' 20.35 40
Rai3
L'attimo fuggente
Film 21.15 150

SEZIONE 2: DOMANDE TEORICHE (VALORE MAX 3 PUNTI)

Domanda 1.

Si supponga di aver un vettore di interi contenente i seguenti valori

3, 5, 7, 8, 10, 16, 24, 32, 38, 45, 67, 89, 92

Quante iterazioni al massimo esegue un algoritmo di ricerca binaria su tale vettore?

Quante iterazioni esegue invece, lo stesso algoritmo, quando viene cercato il valore 32?

Domanda 2.

Definire la struttura dati di un albero binario il cui campo informazione sia un intero. Si implementi una funzione che consenta di calcolare il valore medio delle informazioni contenute nei nodi dell'albero.



SECONDA PROVA IN ITINERE FONDAMENTI DI INFORMATICA

COMPITO 6

SEZIONE 1: PROVA AL CALCOLATORE (VALORE MAX 17 PUNTI)

Implementare una applicazione console in linguaggio ANSI C che permetta ad un centro di dietologia di gestire l'andamento del peso dei suoi pazienti al passare del tempo.

Le informazioni di ciascuna misurazione sono memorizzate in un file (specificato nel seguito) e caricate in una **lista delle misurazioni** ordinata in base alla data. In un momento successivo, l'applicazione dovrà organizzare le informazioni di tali misurazioni in una **lista dei pazienti** in cui ogni elemento della lista contiene, oltre ad informazioni sul paziente, anche l'elenco delle misurazioni di peso di quel paziente.

Specifica della struttura dati:

Misurazioni:

1. *Codice del paziente* (stringa, senza spazi, che può contenere al più 15 caratteri utili)
2. *Peso iniziale* (float, rappresenta il peso all'inizio della dieta)
3. *Valore peso corrente* (float, rappresenta il peso nella misurazione corrente)
4. *Data misurazione* (espressa con tre interi, rispettivamente giorno, mese e anno)

Pazienti:

1. *Codice del paziente* (stringa, senza spazi, che può contenere al più 15 caratteri utili)
2. *Peso iniziale* (float, rappresenta il peso all'inizio della dieta)
3. *Elenco delle misurazioni di peso di quel paziente*, (implementato mediante una lista di misurazioni)

Elenco delle operazioni/funzioni che devono essere implementate:

1. Funzione di caricamento da file di testo delle misurazioni di peso dei pazienti nella lista delle misurazioni. La lista delle misurazioni deve essere mantenuta **ordinata in modo crescente in base alla data di misurazione**. La funzione da implementare deve avere la seguente struttura:
nome della funzione: *CaricaMisurazioni*;
valore restituito: *numero di misurazioni caricate*;
parametri: *nome del file, lista delle misurazioni*.
Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.
2. Funzione che calcoli e restituisca il codice paziente che ha registrato il peso massimo in una delle misurazioni. La funzione da implementare deve avere la seguente struttura:
nome della funzione: *CalcolaPesoMassimo*;
valore restituito: *codice paziente (avente peso massimo)*;
parametri: *lista delle misurazioni*.
Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.
3. Funzione di eliminazione di una misurazione dalla lista. La funzione da implementare deve avere la seguente struttura:
nome della funzione: *EliminaMisurazione*;
valore restituito: *esito operazione (0: operazione fallita, 1: eliminazione effettuata con successo)*;
parametri: *lista delle misurazioni, codice paziente, data misurazione*.
Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.
4. *CopiaMisurazioniInListaPazienti*: Funzione di copia delle informazioni contenute nella lista delle misurazioni nella lista dei pazienti, facendo in modo che ogni paziente contenga soltanto le proprie misurazioni. Lo studente scelga la struttura della funzione, il tipo dei dati ed i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.
5. *ContaPazienti*: funzione che calcoli e restituisca il numero di pazienti presenti nella lista dei pazienti che nelle misurazioni di peso abbia superato almeno una volta il proprio peso iniziale
6. *PesoMedio*: funzione che, fornito come parametro di ingresso un codice paziente, calcoli e restituisca il peso medio di quel paziente in tutte le misurazioni effettuate da quel paziente (e presenti nella lista pazienti). La funzione deve anche restituire se tale peso medio è maggiore o minore del peso iniziale del paziente (1: maggiore, 0: uguale, -1: minore)
7. Funzione che visualizza il contenuto dell'intera lista dei pazienti.
8. Programma principale dotato di un menù testuale che permetta all'utente di usare tutte le funzioni implementate ed eventualmente di inserire i parametri necessari ed eseguire le operazioni richieste.

Struttura del file dei programmi

Le informazioni nel file per ciascuna misurazione si trovano su un'unica riga nel seguente formato:

codicepaziente pesoiniziale pesoscorrente giorno mese anno

Esempio di file:

```
TMR412 85.5 84.7 10 4 2016
TMR412 85.5 83.8 15 5 2016
CTA231 67.2 67.5 15 4 2016
```



UNIVERSITÀ
degli STUDI
di CATANIA

DIPARTIMENTO di INGEGNERIA
ELETTRICA ELETTRONICA
e INFORMATICA

Corso di Laurea in
INGEGNERIA ELETTRONICA
INGEGNERIA INFORMATICA

SECONDA PROVA IN ITINERE FONDAMENTI DI INFORMATICA

TMR412	85.5	84.2	18	6	2016
CRS563	63.4	63.2	20	5	2016
CRS563	63.4	63.8	22	6	2016
CTA231	67.2	66.3	25	6	2016

SEZIONE 2: DOMANDE TEORICHE (VALORE MAX 3 PUNTI)

Domanda 1.

Si supponga di aver un vettore di interi contenente i seguenti valori

12, 5, 25, 15, 17, 8, 42, 14

Mostrare come cambia lo stato del vettore (ad ogni iterazione) applicando un algoritmo di ordinamento di tipo Selection Sort.

Domanda 2.

Definire la struttura dati di un albero binario il cui campo informazione sia un intero. Si implementi una funzione di visita (lo studente indichi quale tipo di visita sta implementando) in maniera iterativa.



SECONDA PROVA IN ITINERE FONDAMENTI DI INFORMATICA

COMPITO 7

SEZIONE 1: PROVA AL CALCOLATORE (VALORE MAX 17 PUNTI)

Implementare una applicazione console in linguaggio ANSI C che permetta di gestire una compagnia di assicurazioni.

Le informazioni dei clienti che hanno stipulato una polizza assicurativa sono memorizzate in un file e caricate in una **lista dei contraenti**.

In un momento successivo, l'applicazione organizza le informazioni della lista dei contraenti in una **lista di agenzie** in cui ogni elemento della lista contiene il riferimento dell'agenzia e la lista dei contraenti che hanno stipulato la polizza assicurativa in quella agenzia.

Specifica della struttura dati:

Contraente:

1. *Codice contraente, implementato con una stringa senza caratteri bianchi, che può contenere al più 7 caratteri utili;*
2. *Codice agenzia, implementato con una stringa senza caratteri bianchi, che può contenere al più 7 caratteri utili;*
3. *Classe di merito, implementato con un intero;*
4. *Data della stipula, implementato con una stringa nel formato AAAA-MM-GG, senza caratteri bianchi, che può contenere al più 15 caratteri utili.*

Agenzia:

1. *Codice agenzia, implementato con una stringa senza caratteri bianchi, che può contenere al più 7 caratteri utili;*
2. *Numero di contraenti, implementato con un intero;*
3. *Elenco dei contraenti, implementato mediante una lista di clienti*

Elenco delle operazioni/funzioni che devono essere implementate:

1. Funzione di caricamento da file dei contraenti nella lista dei contraenti. La funzione da implementare deve avere la seguente struttura:

nome della funzione: *CaricaContraenti*;
valore restituito: *numero di contraenti*;
parametri: *nome del file, lista dei contraenti*.

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.

2. Funzione di modifica della classe di merito di un contraente. La funzione da implementare deve avere la seguente struttura:
nome della funzione: *ModificaClasseDiMerito*;
valore restituito: *esito operazione (0: operazione fallita, 1: modifica effettuata con successo)*;
parametri: *lista dei contraenti, codice contraente, nuova classe di merito*.

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica;

3. Funzione di eliminazione di un contraente. La funzione da implementare deve avere la seguente struttura:
nome della funzione: *EliminaContraente*;
valore restituito: *esito operazione (0: operazione fallita, 1: eliminazione effettuata con successo)*;
parametri: *lista dei contraenti, codice contraente*.

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.

4. Funzione di copia delle informazioni contenute nella lista dei contraenti nella lista di agenzie. La lista di agenzie è ordinata secondo il codice di agenzia. Lo studente scelga la struttura della funzione, il tipo dei dati ed i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.
5. Funzione che calcola e fornisce come parametro il codice dell'agenzia con il maggior numero di contraenti.
6. Funzione che calcola e fornisce come parametro la classe di merito media dei contraenti per ogni agenzia.
7. Funzione che visualizza il contenuto dell'intera lista di agenzie.
8. Programma principale dotato di un menù testuale che permetta all'utente di usare tutte le funzioni implementate ed eventualmente di inserire i parametri necessari ed eseguire le operazioni richieste.

Esempio di file dei contraenti

```
CON0313 CT00123 14 20090426
CON0128 PA00254 8 20010612
CON0098 CT00150 2 19991022
CON0412 ME00025 4 20021104
```



UNIVERSITÀ
degli STUDI
di CATANIA

DIPARTIMENTO di INGEGNERIA
ELETTRICA ELETTRONICA
e INFORMATICA

Corso di Laurea in
INGEGNERIA ELETTRONICA
INGEGNERIA INFORMATICA

SECONDA PROVA IN ITINERE FONDAMENTI DI INFORMATICA

SEZIONE 2: DOMANDE TEORICHE (VALORE MAX 3 PUNTI)

Domanda 1.

Si supponga di aver un vettore di interi contenente i seguenti valori

5, 10, 23, 25, 27, 48

Quante iterazioni esegue la ricerca binaria quando viene cercato il valore 8?

Domanda 2.

Definire la struttura dati di un albero binario il cui campo informazione sia un intero. Si implementi la funzione di visita anticipata. Si consideri la costruzione dell'albero binario di ricerca utilizzando la seguente sequenza di valori: 7 5 1 9 12 8 4 2. Riportare la sequenza di valori ottenuta a seguito della visita anticipata.



SECONDA PROVA IN ITINERE FONDAMENTI DI INFORMATICA

COMPITO 8

SEZIONE 1: PROVA AL CALCOLATORE (VALORE MAX 17 PUNTI)

Implementare una applicazione console in linguaggio ANSI C che permetta di gestire un sistema di inserzioni online.

Le inserzioni sono memorizzate in un file e caricate in una **lista delle inserzioni**.

In un momento successivo, l'applicazione organizza le inserzioni in una **lista di inserzioni per categoria** in cui ogni elemento della lista contiene la categoria a cui fa riferimento l'inserzione e la lista delle inserzioni che ricadono all'interno di quella.

Specifiche della struttura dati:

Inserzione:

1. *Codice inserzione, implementato con un intero*
2. *Categoria, implementato con una stringa senza caratteri bianchi, che può contenere al più 15 caratteri utili;*
3. *Nickname dell'inserzionista, implementato con una stringa senza caratteri bianchi, che può contenere al più 15 caratteri utili;*
4. *Testo, implementato con una stringa di al più 255 caratteri;*
5. *Costo, implementato con un intero.*

Categoria:

1. *Categoria, implementato con una stringa senza caratteri bianchi, che può contenere al più 15 caratteri utili;*
2. *Numero di inserzioni, implementato con un intero;*
3. *Elenco delle inserzioni, implementato mediante una lista di inserzioni*

Elenco delle operazioni/funzioni che devono essere implementate:

1. Funzione di caricamento da file delle inserzioni nella lista delle inserzioni. La funzione da implementare deve avere la seguente struttura:

nome della funzione: *CaricaInserzioni*;
valore restituito: *numero di inserzioni*;
parametri: *nome del file, lista delle inserzioni*.

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.

2. Funzione di modifica del costo in una inserzione. La funzione da implementare deve avere la seguente struttura:
nome della funzione: *ModificaCostoInserzione*;
valore restituito: *esito operazione (0: operazione fallita, 1: modifica effettuata con successo)*;
parametri: *lista delle inserzioni, codice inserzione, nuovo costo*.

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica;

3. Funzione di eliminazione di una inserzione. La funzione da implementare deve avere la seguente struttura:
i. nome della funzione: *EliminaInserzione*;
ii. valore restituito: *esito operazione (0: operazione fallita, 1: eliminazione effettuata con successo)*;
iii. parametri: *lista delle inserzioni, codice inserzione*.

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.

4. Funzione di copia delle informazioni contenute nella lista delle inserzioni nella lista delle categorie. La lista delle categorie è ordinata per categoria. Lo studente scelga la struttura della funzione, il tipo dei dati ed i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.
5. Funzione che calcola e fornisce come parametro la categoria con il maggior numero di inserzioni.
6. Funzione che accetta come parametro il nickname di un inserzionista e calcola e fornisce come parametro il numero delle sue inserzioni in ogni categoria.
7. Funzione che visualizza il contenuto dell'intera lista delle categorie.
8. Programma principale dotato di un menù testuale che permetta all'utente di usare tutte le funzioni implementate ed eventualmente di inserire i parametri necessari ed eseguire le operazioni richieste.

Struttura del file delle inserzioni

Le informazioni dell'inserzione si trovano distribuite su tre righe:

Prima riga: codice inserzione, categoria, nickname

Seconda riga: testo

Terza riga: costo

Esempio:

```
1      IMMOBILI      Giovanni
Appartamento 3 vani a Catania
150000
5      INFORMATICA Luca
Stampante laser 600dpi
50
8      CUCINA        Giovanni
Robot da cucina
100
```



UNIVERSITÀ
degli STUDI
di CATANIA

DIPARTIMENTO di INGEGNERIA
ELETTRICA ELETTRONICA
e INFORMATICA

Corso di Laurea in
INGEGNERIA ELETTRONICA
INGEGNERIA INFORMATICA

SECONDA PROVA IN ITINERE FONDAMENTI DI INFORMATICA

12 INFORMATICA Filippo
Monitor 20 pollici
90

SEZIONE 2: DOMANDE TEORICHE (VALORE MAX 3 PUNTI)

Domanda 1.

Si supponga di avere un vettore di interi contenente i seguenti valori
10, 30, 223, 500, 700, 800

Quante iterazioni esegue la ricerca binaria quando viene cercato il valore 30?

Domanda 2.

Definire la struttura dati di un albero binario il cui campo informazione sia un intero. Si implementi la funzione di visita anticipata. Si consideri la costruzione dell'albero binario di ricerca utilizzando la seguente sequenza di valori: 8 4 7 9 12 5 6 1. Riportare la sequenza di valori ottenuta a seguito della visita simmetrica.