



FONDAMENTI DI INFORMATICA – Appello del 6 Settembre 2017

COMPITO 1

Cognome e Nome	Matricola n.
----------------	--------------

SEZIONE 1: PROVA AL CALCOLATORE (VALORE MAX 23 PUNTI)

Implementare una applicazione console in linguaggio ANSI C per la gestione delle comande in un fastfood. Le informazioni di ciascuna comanda sono memorizzate in un file (specificato nel seguito) e caricate in una **lista di comande** ordinata in base all'orario di acquisizione della comanda. Come ipotesi semplificativa si assuma che una comanda possa fare riferimento ad un unico menù del fastfood. In un momento successivo, l'applicazione dovrà organizzare le informazioni di tali comande in una **lista di menu** in cui ogni elemento della lista contiene, oltre all'identificativo del menu, anche l'elenco delle comande relative a quel menu.

Specifica della struttura dati:

Comanda:

1. *Codice* (codice univoco della comanda: stringa di 7 caratteri)
2. *Orario* (hh:mm nel formato 24 ore)
3. *Menu* (codice del menu: intero)
4. *Takeaway* (da portare via o consumare sul posto: intero)

Menu:

1. *Menu* (codice del menu: intero)
2. *Comande* (elenco delle comande relative a quel menu: lista di Comanda)

Elenco delle operazioni/funzioni che devono essere implementate:

1. Funzione di caricamento da file di testo degli articoli nella lista delle comande. La lista delle comande deve essere mantenuta **ordinata in base all'orario della comanda** (dalla meno recente alla più recente). La funzione da implementare deve avere la seguente struttura:
nome della funzione: *CaricaComanda*;
valore restituito: *numero di comande caricate*;
parametri: *nome del file, lista delle comande*.
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 menu di una determinata comanda. La funzione da implementare deve avere la seguente struttura:
nome della funzione: *ModificaComanda*;
valore restituito: *esito operazione (0: operazione fallita, 1: modifica effettuata con successo)*;
parametri: *lista delle comande, codice comanda, nuovo menu*.
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 comanda dalla lista. La funzione da implementare deve avere la seguente struttura:
nome della funzione: *EliminaComanda*;
valore restituito: *esito operazione (0: operazione fallita, 1: eliminazione effettuata con successo)*;
parametri: *lista delle comande, codice della comanda da eliminare*.
Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.
4. *SpostaComandeInListaMenu*: Sposta (cioè copia ed elimina le comande dalla lista di comande) le informazioni contenute nella lista delle comande nella lista dei menu, facendo in modo che ogni nodo della lista dei menu contenga l'elenco delle comande riferite a quel menu. 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. *PercentualeDiComandeTakeAway*: Funzione che, operando sulla lista dei menu e fornito come parametro di ingresso un menu, calcoli e restituisca la percentuale di comande di quel menù takeaway.
6. *VisualizzaComandeInAttesa*: Funzione che, operando sulla lista dei generi alimentari e fornito come parametro di ingresso l'orario attuale, visualizzi tutte le comande con un orario anteriore all'orario attuale di 30 minuti. La funzione deve inoltre restituire il numero di comande visualizzate.
7. Funzione che visualizza il contenuto dell'intera lista di menu.
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.



UNIVERSITÀ
degli STUDI
di CATANIA

DIPARTIMENTO di INGEGNERIA
ELETTRICA ELETTRONICA
e INFORMATICA

Corso di Laurea in
INGEGNERIA ELETTRONICA
INGEGNERIA INFORMATICA

FONDAMENTI DI INFORMATICA – Appello del 6 Settembre 2017

Struttura del file degli articoli

Le informazioni nel file per ciascuna comanda si trovano distribuite su una riga:

Codice Orario Menu Takeaway

Esempio di file:

```
0000001 18:00 3 1
0000002 18:13 2 0
0000003 18:25 1 0
0000004 18:40 1 1
```



FONDAMENTI DI INFORMATICA – Appello del 6 Settembre 2017

COMPITO 1

Cognome e Nome		Matricola n.	
----------------	--	--------------	--

SEZIONE 2: DOMANDE TEORICHE (VALORE MAX 5 PUNTI)

Domanda 1 (punti 0,5)

Qual è il ruolo dei registri in una CPU?

Domanda 2 (punti 0,5)

Dato il seguente frammento di codice, cosa viene visualizzato sullo schermo?

<pre>#include <stdio.h> int main(void) { int i, j; for (i=0; i<3; i++) { j=0; while (j<i) printf("%d %d", i, j++); } }</pre>	
--	--

Domanda 3 (punti 0,5)

Verificare se il programma è corretto e in caso affermativo indicare cosa viene visualizzato sullo schermo. Se invece il programma contiene degli errori, descriverli, proporre una soluzione ed indicare cosa verrebbe visualizzato sullo schermo.

<pre>#include <stdio.h> #include <string.h> int main(void) { char str1[] = "pippo"; char str2[] = "pluto"; if (str1 != str2) printf("primo"); else printf("secondo"); }</pre>	
---	--



FONDAMENTI DI INFORMATICA – Appello del 6 Settembre 2017

Domanda 4 (punti 1,5)

Si supponga di avere un array di interi contenente i seguenti valori:

15, 30, 40, 45, 75, 77, 84, 90, 100

Quante iterazioni vengono effettuate da un algoritmo di ricerca binaria per cercare il valore 90? Indicare il valore trovato ad ogni iterazione.

Domanda 5 (punti 2)

Si supponga di costruire un albero binario di ricerca secondo la seguente sequenza di inserimento:

20, 18, 5, 9, 23, 40, 15, 35

Qual è la sequenza di valori ottenuti quando si effettua una visita simmetrica?



FONDAMENTI DI INFORMATICA – Appello del 6 Settembre 2017

COMPITO 2

Cognome e Nome		Matricola n.	
----------------	--	--------------	--

SEZIONE 1: PROVA AL CALCOLATORE (VALORE MAX 23 PUNTI)

Implementare una applicazione console in linguaggio ANSI C che permetta di gestire le fatture di negozio. Le informazioni di ciascuna fattura emessa devono essere acquisite da standard input e caricate in una **lista dei prodotti acquistati**. In un momento successivo, l'applicazione dovrà organizzare le informazioni di tali prodotti in una **lista di fatture** in cui ogni elemento della lista contiene, oltre ai dati generali della fattura, anche l'elenco dei dettagli dei prodotti inclusi in quella fattura. La lista di fatture dovrà essere mantenuta **ordinata** in base al codice fattura.

Specifica della struttura dati:

Prodotto:

1. *Codice fattura* (codice alfanumerico senza spazi, che può contenere al più 15 caratteri utili)
2. *Codice cliente* (codice alfanumerico senza spazi, che può contenere al più 15 caratteri utili)
3. *Codice prodotto* (codice alfanumerico senza spazi, che può contenere al più 31 caratteri utili)
4. *Quantità acquistata* (intero)
5. *Prezzo unitario prodotto*
6. *Prezzo totale prodotto*

Fattura:

1. *Codice fattura* (codice alfanumerico senza spazi, che può contenere al più 15 caratteri utili)
2. *Codice cliente* (codice alfanumerico senza spazi, che può contenere al più 15 caratteri utili)
3. *Importo totale fattura*
4. *Elenco dei prodotti di quella fattura*, (implementato mediante una lista di Prodotti)

Elenco delle operazioni/funzioni che devono essere implementate:

1. Funzione di inserimento dei dati dei prodotti acquistati nella lista dei Prodotti. La funzione da implementare deve avere la seguente struttura:

nome della funzione: *InserimentoProdotto*;

valore restituito: *esito inserimento*;

parametri: *lista dei prodotti, prodotto*.

Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica. Inoltre, la funzione dovrà, di volta in volta, calcolare e valorizzare opportunamente il campo relativo al prezzo totale del prodotto.

2. Funzione di modifica del prezzo unitario di un determinato prodotto. La funzione da implementare deve avere la seguente struttura:

nome della funzione: *ModificaPrezzoProdotto*;

valore restituito: *esito operazione (0: operazione fallita, 1: modifica effettuata con successo)*;

parametri: *lista dei prodotti, codice prodotto, nuovo prezzo unitario prodotto*.

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

3. Funzione che salva i dati presenti nella lista di prodotti in un file di testo. La funzione da implementare deve avere la seguente struttura:

nome della funzione: *SalvaProdotti*;

valore restituito: *numero di prodotti salvati su file*;

parametri: *lista dei prodotti, nome del file*.

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

4. *CopiaProdottiInListaFatture*: Funzione di copia delle informazioni dalla lista dei prodotti nella lista delle fatture, facendo in modo che ogni fattura contenga soltanto i prodotti pertinenti a quella fattura. La lista di fatture dovrà essere mantenuta **ordinata** in base al codice fattura. Inoltre, la funzione dovrà, di volta in volta, calcolare e valorizzare opportunamente l'importo totale di ogni fattura. 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. *CalcoloImportoMinMax*: Funzione che, operando sulla lista delle fatture, calcoli e restituisca l'importo minimo e quello massimo delle fatture presenti in lista. Lo studente scelga la struttura della funzione, il tipo dei dati ed i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.



UNIVERSITÀ
degli STUDI
di CATANIA

DIPARTIMENTO di INGEGNERIA
ELETTRICA ELETTRONICA
e INFORMATICA

Corso di Laurea in
INGEGNERIA ELETTRONICA
INGEGNERIA INFORMATICA

FONDAMENTI DI INFORMATICA – Appello del 6 Settembre 2017

6. *QuantitaProdottoinFatture*: Funzione che, operando sulla lista delle fatture, e fornito come parametro di ingresso il codice di un prodotto, calcoli e restituisca la quantità media acquistata di quel prodotto in tutte le fatture in cui esso è presente.
7. Funzione che visualizza il contenuto dell'intera lista di fatture.
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.



FONDAMENTI DI INFORMATICA – Appello del 6 Settembre 2017

COMPITO 2

Cognome e Nome		Matricola n.	
----------------	--	--------------	--

SEZIONE 2: DOMANDE TEORICHE (VALORE MAX 5 PUNTI)

Domanda 1 (punti 0,5)

Descrivere le fasi di fetch, decode ed execute svolte dalla CPU durante il suo funzionamento.

Domanda 2 (punti 0,5)

Dato il seguente frammento di codice, cosa viene visualizzato sullo schermo?

<pre>#include <stdio.h> int main(void) { int v[5] = {3,2,6,4,1}; int prodotto=1; for (i=0; i<5; i++) if (i % 2 == 0){ prodotto *= v[i]; printf("%d %d\n", i, v[i]); } printf("%d", prodotto); }</pre>	
--	--

Domanda 3 (punti 0,5)

Verificare se il programma è corretto e in caso affermativo indicare cosa viene visualizzato sullo schermo. Se invece il programma contiene degli errori, descriverli, proporre una soluzione ed indicare cosa verrebbe visualizzato sullo schermo.

<pre>#include <stdio.h> int main(void) { int v[]={1,3,5,7,9}; for (i<5; i>=0; i--) printf("%d %d \n", i, v[i]); }</pre>	
---	--



UNIVERSITÀ
degli STUDI
di CATANIA

DIPARTIMENTO di INGEGNERIA
ELETTRICA ELETTRONICA
e INFORMATICA

Corso di Laurea in
INGEGNERIA ELETTRONICA
INGEGNERIA INFORMATICA

FONDAMENTI DI INFORMATICA – Appello del 6 Settembre 2017

Domanda 4 (punti 1,5)

Si supponga di avere un array di interi contenente i seguenti valori:

19, 33, 44, 48, 57, 61, 72, 77, 83, 86, 94

Quante iterazioni vengono effettuate da un algoritmo di ricerca binaria per cercare il valore 94? Indicare inoltre la parte del vettore considerato dall'algoritmo ad ogni iterazione.

Domanda 5 (punti 2)

Scrivere un algoritmo che, dato un albero binario di ricerca le cui etichette sono valori interi, permetta di calcolare la somma di tutti i nodi dell'albero aventi etichetta maggiore di un determinato valore N.