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 25 Settembre 2017

COMPITO 1

Cognome e Nome
Numero di Matricola

Domanda 1 (punti 0,5)

Quali sono le principali funzioni del File System?

Domanda 2 (punti 0,5)

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

```c
#include <stdio.h>
int main(void) {
       int i, k;
       for (i = 0; i<3; i++)
          for (k = 3; k >= 0; k--)
              printf("%d ", i + k);
}
```

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

```c
#include <stdio.h>
#include <string.h>
#define str1 "mondo"
int main(void) {
       char str2[] = “ciao”;
       char * str3;

       str3 = str1 + str2;
       printf("%s ", str3);
}
```

FONDAMENTI DI INFORMATICA – Appello del 25 Settembre 2017

Domanda 4 (punti 1,5)

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

100, 150, 250, 400, 500, 502, 700, 710, 800

Quante iterazioni vengono effettuate da un algoritmo di ricerca binaria per verificare se il valore 101 esiste? Scrivere il valore Trovato in ogni interazione

Domanda 5 (punti 2)

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

6 2 4 5 8 1 7 9

Disegnare l'albero ottenuto e scrivere la sequenza di valori ottenuti quando si effettua una visita anticipata?

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 25 Settembre 2017

Implementare una applicazione console in linguaggio ANSI C che permetta di gestire l'organizzazione dei turni in sala operatoria in un ospedale. In una prima fase i dati vengono raccolti in una lista di richieste (**`lista_richieste`**) nella quale è indicato il nome del paziente, il tempo previsto e l'urgenza dell'intervento. Poiché l'ospedale ha quattro camere operatorie, in una seconda fase i dati vengono letti dalla lista delle richieste ed inserite nella coda di attesa (**`coda_attesa`**) di ogni sala operatoria garantendo che ogni sala operatoria non sia usata per più di 5 ore.

Specifica della struttura dati:

`t_richiesta`:

1. **`paziente`***: nome del paziente, rappresentato con una stringa senza caratteri bianchi, che può contenere al più 31 caratteri utili;*
2. **`urgenza`***: valore intero positivo nell'intervallo [0,3], a valore maggiore corrisponde urgenza maggiore;*
3. **`durata`***: indica la durata prevista in minuti dell'intervento.*

L'insieme delle sale operatorie è implementato tramite un vettore di code di attesa o, in subordine e con una valutazione inferiore, come lista di sale operatorie.

Elenco delle operazioni/funzioni che devono essere implementate:

1. Funzione di inserimento in lista di una nuova richiesta. La funzione deve rispettare le seguenti specifiche:
   nome della funzione: **`inserimento_richiesta`**;
   valore restituito: **`nessuno`**;
   parametri: **`dati della prenotazione, lista delle richieste`**
   Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica. La funzione **non prevede nessuna interazione diretta** con l'utente mediante tastiera, tutte le informazioni devono essere passate mediante gli argomenti della funzione. L'inserimento di un dialogo con l'utente (scanf, printf, ecc.) porta ad una valutazione nulla della funzione.
2. Salvataggio dei dati presenti nella lista delle richieste in un file di testo. La funzione deve rispettare le seguenti specifiche:
   nome della funzione: **`salva_richieste`**;
   valore restituito: **`numero di richieste salvate nel file`**;
   parametri: **`descrittore del file (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, La funzione **non prevede nessuna interazione diretta** con l'utente mediante tastiera, tutte le informazioni devono essere passate mediante gli argomenti della funzione. Il descrittore del file contiene il riferimento ad un file che deve essere già stato aperto nel programma chiamante;
3. Funzione di ricerca e di una prenotazione: dato il nome del paziente verificare se è presente in lista fornendo il numero il valore della durata prevista. La funzione deve rispettare le seguenti specifiche:
   nome della funzione: **`verifica_richiesta`**;
   valore restituito: **`0 se il paziente non è presente, 1 altrimenti`**;
   parametri: **`nome del paziente, lista delle richieste, durata dell'operazione`**
   Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica. La funzione **non prevede nessuna interazione diretta** con l'utente mediante tastiera, tutte le informazioni devono essere passate mediante gli argomenti della funzione
4. Funzione di riempimento delle code di attesa delle sale operatorie:
   nome della funzione: **`carica_code`**;
   L'algoritmo da implementare è il seguente: si scandisce la lista delle richieste selezionando inizialmente la priorità maggiore, e si procede ad inserire la richiesta nella prima coda dopo aver verificato che il tempo limite non sia stato superato, altrimenti si passa alla coda successiva fino all'esaurimento delle code, dopo l'inserimento in coda la richiesta deve essere cancellata dalla lista delle richieste. Si procede con lo stesso metodo a selezionare i pazienti con priorità inferiore (nota che avendo cancellato gli elementi non sono più presenti richieste con priorità maggiore) fino all'esaurimento delle richieste o del tempo a disposizione. Lo studente scelga i parametri e i valori di ritorno nel modo che ritiene più appropriato. Tutte le operazioni devono essere fatte in memoria centrale e non è **prevista nessuna interazione diretta** con l'utente mediante console.
5. Funzione che cancella dalla lista delle richieste un paziente dato il nome. Tutte le operazioni devono essere fatte in memoria centrale e non è **prevista nessuna interazione diretta** con l'utente mediante console;
6. Funzione che visualizza sullo schermo il contenuto dell'intera struttura dati implementata: nota che la stampa della struttura dati **`lista_richieste`** avrà una valutazione inferiore rispetto alla stampa della struttura che contiene le code di attesa.
7. Calcolo dell'occupazione media delle sale operatorie (deve essere calcolato il rapporto fra l'occupazione media delle quattro sale operatorie e la capienza)
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 per 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 25 Settembre 2017

Note:

- Le funzioni 1, 2, 3 hanno come ingresso e uscita esclusivamente gli argomenti ed il tipo di ritorno, pertanto qualsiasi interazione con l'utente atta ad acquisire i dati o stampare i risultati deve essere implementata nel programma chiamante;
- La funzione 2 prevede che venga passato come parametro il descrittore del file che si considera già correttamente aperto nel programma chiamante;
- La specifica degli argomenti in ingresso o uscita delle funzioni indica quale informazione deve essere fornita non il tipo di dato ed il meccanismo di passaggio che deve essere scelto dallo studente;
- La funzione 6 deve operare sulla struttura in memoria centrale e stampare il contenuto della struttura annidata, qualora si stampi solo il contenuto della lista dei pazienti la valutazione sarà inferiore;
- Il file nell'esempio ha la funzione di definire il formato dei dati che deve essere rispettato alla lettera, il numero di elementi presenti non è soggetto ad alcun vincolo.
- Tutte le operazioni, quando non specificato esplicitamente, devono essere effettuate in memoria centrale.

Esempio di file di di testo:

```
VERDI_CARLO          // nome paziente
0                    // urgenza
45                   // durata
ROSSI_GUIDO
2
210
BIANCHI_ANGELO
3
60
```

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 25 Settembre 2017

COMPITO 2

Cognome e Nome
Numero di Matricola

Domanda 1 (punti 0,5)

Quali sono le principali caratteristiche della rappresentazione dei numeri interi in complemento a 2?

Domanda 2 (punti 0,5)

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

```c
#include <stdio.h>
int main(void) {
	int i, k;
	for (i = 0; i<18; i += 3)
		if (i % 3 == 0) continue;
	printf("%d ", i);
}
```

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

```c
#include <stdio.h>
int main(void) {
	int x[10] = { 0 };
	int y, i;

	y = &x;
	for (i = 5; i < 15; ++i)
		printf("%d ", y[i]);
}
```

FONDAMENTI DI INFORMATICA – Appello del 25 Settembre 2017

Domanda 4 (punti 1,5)

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

100, 150, 250, 400, 500, 502, 700, 710, 800

Supponendo di utilizzare un algoritmo di ordinamento per selezione, quante iterazioni vengono effettuate? Scrivere lo stato del vettore ad ogni iterazione.

Domanda 5 (punti 2)

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

10 20 30 40 50 60 70 80 90 100

Disegnare l'albero ottenuto e scrivere la sequenza di valori ottenuti quando si effettua una visita posticipata?

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 25 Settembre 2017

Implementare una applicazione console in linguaggio ANSI C che permetta di gestire l'istradamento dei treni di una ferrovia locale. I treni in attesa in stazione sono inseriti in una lista a contengono un campo intero che rappresenta i turni di attesa previsti. In una prima fase i dati vengono raccolti nella lista di treni in ingresso (**`lista_ingresso`**) nella quale è indicato il codice del treno, il nome della stazione di destinazione ed una variabile che conta i cicli di attesa (**`turni`**). Nella seconda fase i treni in cui il valore di turno è minore o uguale a zero vengono inseriti nella coda associata alla stazione di destinazione.

Specifica della struttura dati:

`t_treno`:

4. **`codice`***: codice alfanumerico che identifica il singolo treno, rappresentato con una stringa senza caratteri bianchi, che può contenere al più 11 caratteri utili;*
5. **`stazione`***: nome della stazione di destinazione, rappresentato con una stringa senza caratteri bianchi, che può contenere al più 31 caratteri utili;*
6. **`turni`***: valore intero che indica il numero di cicli che il treno deve aspettare per essere instradato;*
7. **`stato`***: valore intero che rappresenta lo stato del treno, 1 pronto, 0 non pronto*

`t_stazione`

- **`stazione`***: nome della stazione di destinazione, rappresentato con una stringa senza caratteri bianchi, che può contenere al più 31 caratteri utili;*
- **`treni_pronti`***: coda dei treni pronti alla partenza;*

Elenco delle operazioni/funzioni che devono essere implementate:

9. Funzione di inserimento in lista di un nuovo treno. La funzione deve rispettare le seguenti specifiche:
   nome della funzione: **`inserimento_treno`**;
   valore restituito: **`numero di treni presenti nella lista`**;
   parametri: **`dati del treno, lista delle richieste`**
   Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica. La funzione **non prevede nessuna interazione diretta** con l'utente mediante tastiera, tutte le informazioni devono essere passata mediante gli argomenti della funzione. L'inserimento di un dialogo con l'utente (ad esempio scanf, printf, ecc.) porta ad una valutazione nulla della funzione.
10. Caricamento dei dati da un file di testo alla lista dei treni. La funzione deve rispettare le seguenti specifiche:
    nome della funzione: **`carica_richieste`**;
    valore restituito: **`numero di richieste caricate`**;
    parametri: **`descrittore del file (FILE *), lista dei treni`**.
    Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica, La funzione **non prevede nessuna interazione diretta** con l'utente mediante tastiera, tutte le informazioni devono essere passata mediante gli argomenti della funzione. Il descrittore del file contiene il riferimento ad un file che deve essere già stato aperto nel programma chiamante;
11. Funzione di aggiornamento dei cicli di attesa. La funzione deve diminuire di uno il numero di turno di tutti i treni pronti (stato = 1) presenti nella lista. La funzione deve rispettare le seguenti specifiche:
    nome della funzione: **`aggiornamento_turni`**;
    valore restituito: **`nessuno`**;
    parametri: **`lista delle richieste, numero treni da instradare (turni <= 0)`**
    Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica. La funzione **non prevede nessuna interazione diretta** con l'utente mediante tastiera, tutte le informazioni devono essere passata mediante gli argomenti della funzione
12. Funzione di riempimento delle code di attesa:
    nome della funzione: **`carica_code`**;
    L'algoritmo da implementare è il seguente: si scandisce la lista delle richieste selezionando tutti i treni che hanno valore di turni <=0 che vengono estratti (e quindi cancellati dalla lista dei treni) ed inseriti nella coda appropriata in base al nome della stazione. Lo studente scelga i parametri e i valori di ritorno nel modo che ritiene più appropriato. Tutte le operazioni devono essere fatte in memoria centrale e non è **prevista nessuna interazione diretta** con l'utente mediante console.
13. Funzione che cancella dalla lista dei treni un elemento dato il suo codice. Tutte le operazioni devono essere fatte in memoria centrale e non è **prevista nessuna interazione diretta** con l'utente mediante console;
14. Funzione che visualizza sullo schermo il contenuto dell'intera struttura dati implementata: nota che la stampa della struttura dati **`lista dei treni`** avrà una valutazione inferiore rispetto alla stampa della struttura che contiene le code di attesa.
15. Identificazione della coda che contiene il maggior numero di treni in attesa ed estrazione dell'elemento affiorante (primo elemento inserito nella coda) che deve essere fornito al programma chiamante che procederà alla stampa.
16. Programma principale dotato di un menù testuale che permetta all'utente di usare tutte le funzioni implementate ed eventualmente di inserire i parametri necessari per eseguire le operazioni richieste o stampare i risultati.

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 25 Settembre 2017

Note:

- Le funzioni 1, 2, 3 hanno come ingresso e uscita esclusivamente gli argomenti ed il tipo di ritorno, pertanto qualsiasi interazione con l'utente atta ad acquisire i dati o stampare i risultati deve essere implementata nel programma chiamante;
- La funzione 2 prevede che venga passato come parametro il descrittore del file che si considera già correttamente aperto nel programma chiamante;
- La specifica degli argomenti in ingresso o uscita delle funzioni indica quale informazione deve essere fornita non il tipo di dato ed il meccanismo di passaggio che deve essere scelto dallo studente;
- La funzione 6 deve operare sulla struttura in memoria centrale e stampare il contenuto della struttura annidata, qualora si stampi solo il contenuto della lista dei treni la valutazione sarà inferiore;
- Il file nell'esempio ha la funzione di definire il formato dei dati che deve essere rispettato alla lettera, il numero di elementi presenti non è soggetto ad alcun vincolo.
- Tutte le operazioni, quando non specificato esplicitamente, devono essere effettuate in memoria centrale.

Esempio di file di testo:

```
TR001              // codice del treno
MESSINA            // stazione di destinazione
12                 // turni
1                  // stato
TR002
MESSINA
12
0
TRA01
PALERMO
8
1
TRA05
RAGUSA
7
1
TRA05BIS
RAGUSA
7
0
```

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 25 Settembre 2017

COMPITO 3

Cognome e Nome
Numero di Matricola

Domanda 1 (punti 0,5)

Quali sono le differenze tra linguaggi ad alto livello e a basso livello?

Domanda 2 (punti 0,5)

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

```c
#include <stdio.h>
int main(void) {
       int i, j;
       for (i = 0; i<3; i++)
          for (j = 0; j<=2; j++)
              printf("%d %d\n", i, j);
}
```

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

```c
#include <stdio.h>
#include <string.h>
int main(void) {
       char src[] = “ciao”;
       char dst[100];

       dst = src;
       printf("%s ", dst);
}
```

FONDAMENTI DI INFORMATICA – Appello del 25 Settembre 2017

Domanda 4 (punti 1,5)

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

100, 150, 250, 400, 500, 502, 700, 710, 800

Quante iterazioni vengono effettuate da un algoritmo di ricerca binaria per verificare se il valore 501 esiste? Scrivere il valore trovato in ogni interazione

Domanda 5 (punti 2)

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

9 7 1 8 5 4 2 6

Disegnare l'albero ottenuto e scrivere la sequenza di valori ottenuti quando si effettua una visita anticipata?

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 25 Settembre 2017

PROVA AL CALCOLATORE (VALORE MAX 23 PUNTI)

Implementare una applicazione console in linguaggio ANSI C per la gestione delle graduatorie di un concorso. Il concorso prevede diverse posizioni. Le informazioni di ciascun candidato sono memorizzate in un file (specificato nel seguito) e caricate in una **lista di candidatie**. In un momento successivo, l'applicazione dovrà organizzare le informazioni dei candidati in una **lista di posizioni** in cui ogni elemento della lista contiene, oltre all'identificativo della posizione, anche l'elenco dei candidati a quella posizione. Tale elenco è ordinato in modo decrescente in base al punteggio del candidato.

Specifica della struttura dati:

Candidato:

1. *Codice fiscale* *(codice fiscale del candidato: stringa di 16 caratteri)*
2. *Punteggio* *(punteggio: intero)*
3. *Codice posizione* *(codice della posizione a cui sta partecipando: stringa di 4 caratteri)*
4. *Disponibilità* *(disponibilità a spostamenti di lungo periodo: interi (si/no))*

Posizione:

1. *Codice posizione* *(codice della posizione: stringa di 4 caratteri)*
2. *Elenco candidati* *(elenco dei candidati per quella posizione ordinato per punteggio: lista di Candidato)*

Elenco delle operazioni/funzioni che devono essere implementate:

1. Funzione di caricamento da file di testo dei candidati nella lista dei candidati. La funzione da implementare deve avere la seguente struttura:
   nome: *CaricaCandidati;*
   valore restituito: *numero di candidati caricati*;
   parametri formali: *nome del file, lista dei candidati*.
2. Funzione di modifica del punteggio di un candidato per una specifica posizione. La funzione da implementare deve avere la seguente struttura:
   nome: *ModificaPunteggio*;
   valore restituito: *esito operazione (0: operazione fallita, 1: modifica effettuata con successo)*;
   parametri formali: *lista delle candidati, codice fiscale candidato, codice posizione, nuovo punteggio*.
3. Funzione di eliminazione di un candidato per una specifica. La funzione da implementare deve avere la seguente struttura:
   nome: *EliminaCandidato*;
   valore restituito: *esito operazione (0: operazione fallita, 1: eliminazione effettuata con successo)*;
   parametri formali: *lista dei candidati, codice fiscale del candidato, codice posizione*.
4. *CreaListaDellePosizioni*: A partire dalla lista dei Candidati crea la lista delle Posizioni. Ogni nodo della lista delle Posizioni contiene il codice della posizione e l'elenco dei candidati per quella posizione. L'elenco dei candidati è ordinati in modo decrescente in base al punteggio. Si noti che, lo stesso individuo (cioè lo stesso codice fiscale) può partecipare per più posizioni. 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. *PunteggioMedio:* Funzione che, operando sulla lista delle posizioni e fornito come parametro di ingresso il codice di una posizione, calcoli e restituisca la media dei punteggi per quella posizione.
6. *CandidatoDisponibileTrasferimento*: Funzione che, operando sulla lista delle posizioni e fornito come parametro di ingresso il codice di una posizione, restituisca le informazioni del candidato con il punteggio massimo e che è disponibile a trasferimenti.
7. Funzione che visualizza il contenuto dell'intera lista delle posizioni.
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 25 Settembre 2017

COMPITO 4

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 di fondi di investimento. Le informazioni di ciascun fondo sono memorizzate in un file (specificato nel seguito) e caricate in una **lista di fondi**. In un momento successivo (funzione 4), l'applicazione dovrà organizzare le informazioni di tali fondi in una **lista di tipologie** in cui ogni elemento della lista contiene, oltre all'identificativo della categoria, anche l'elenco dei fondi appartenenti a quella tipologia.

Specifica della struttura dati:

Fondo:

5. *Codice* *(codice univoco del fondo: stringa di 7 caratteri)*
6. *Descrizione* *(stringa, al più 31 caratteri utili)*
7. *Tipologia* *(stringa, al più 15 caratteri, es. "azionario", "bilanciato", "obbligazionario",...)*
8. *Valore quota* *(valore in euro della quota)*
9. *Rating* *(intero: da 1 a 5 stelle, indica la qualità del fondo))*

TipologiaFondi:

3. *Tipologia* *(stringa, al più 15 caratteri,*
4. *ElencoFondi* *(elenco dei fondi di quella tipologia, implementato mediante una lista di Fondo)*

Elenco delle operazioni/funzioni che devono essere implementate:

9. Funzione di caricamento da file di testo dei fondi nella lista dei fondi. La funzione da implementare deve avere la seguente struttura:
   nome della funzione: *CaricaFondi;*
   valore restituito: *numero di fondi caricati*;
   parametri: *nome del file, lista dei fondi*.
   Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.
10. Funzione di modifica del valore della quota di un determinato fondo. La funzione da implementare deve avere la seguente struttura:
    nome della funzione: *ModificaValore*;
    valore restituito: *esito operazione (0: operazione fallita, 1: modifica effettuata con successo)*;
    parametri: *lista dei fondi, codice fondo, nuovo valore quota*.
    Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.
11. Funzione di eliminazione di una fondo dalla lista. La funzione da implementare deve avere la seguente struttura:
    nome della funzione: *EliminaFondo*;
    valore restituito: *esito operazione (0: operazione fallita, 1: eliminazione effettuata con successo)*;
    parametri: *lista dei fondi, codice del fondo da eliminare*.
    Lo studente scelga sia il tipo dei dati che i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.
12. *SpostaFondiInListaCategorie*: Sposta (**cioè copia ed elimina i fondi dalla lista dei fondi**) le informazioni dalla lista dei fondi alla lista delle tipologie, facendo in modo che ogni nodo della lista delle tipologie contenga l'elenco dei fondi classificati con quella tipologia. Lo studente scelga la struttura della funzione, il tipo dei dati ed i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.
13. *ValoreMedioFondi:* Funzione che, operando sulla lista delle tipologie e fornito come parametro di ingresso una tipologia di fondi, calcoli e restituisca il valore medio delle quote dei fondi di quella tipologia. Lo studente scelga la struttura della funzione, il tipo dei dati ed i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.
14. *Rating*: Funzione che, operando sulla lista delle tipologie e fornito come parametro di ingresso una valore soglia di rating, calcoli e fornisca in uscita in un opportuno vettore il numero di fondi aventi rating maggiore o uguale al valore soglia passato come parametro per ciascuna tipologia di fondi. Lo studente scelga la struttura della funzione, il tipo dei dati ed i meccanismi di passaggio dei parametri più opportuni per rispettare la specifica.
15. Funzione che visualizza il contenuto dell'intera lista di tipologie.
16. 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 25 Settembre 2017

Struttura del file dei fondi

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

Codice
Descrizione
Tipologia ValoreQuota Rating

Esempio di file:

```
A07R534
Pioneer Global Fund
Azionario     132.45 4
B56TG42
Allianz Eastern Europe
Azionario     22.85  5
5HJ34R6
Anima Iniziativa Italia
Obbligazionario      36.98  3
HJ73K83
Parvest Real Estate Securities
Monetario     13.43  2
YJ56W34
Amundi Accumulazione Attiva
Obbligazionario      25.38  2
```

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 25 Settembre 2017

COMPITO 4

Cognome e Nome		**Matricola n.**	

SEZIONE 2: DOMANDE TEORICHE (VALORE MAX 5 PUNTI)

Domanda 1 (punti 0,5)

Quali sono le principali funzioni del File System in un sistema operativo?

Domanda 2 (punti 0,5)

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

```
#include <stdio.h>
int main(void) {
   int i, j;
   for (i=0; i<3; i++) {
      for (j=0; j<=5; j++)
        printf("%d ", i*j);
      printf("\n");
   }
}
```

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.

```
#include <stdio.h>
int main(void) {
      int x[6] = {1,2,3,4,5,6};
      int z, i;

      z = &x;
      for (i = 0; i <= 6; ++i)
        printf("%d ", z[i]);
}
```

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 25 Settembre 2017

Domanda 4 (punti 1,5)

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

16, 32, 41, 46, 74, 78, 84, 92, 100

Quante iterazioni vengono effettuate da un algoritmo di ricerca binaria per cercare il valore 32? 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 inerimento:

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

Disegnare l'albero ottenuto e scrivere la sequenza di valori ottenuti quando si effettua una visita in preordine.