



**FONDAMENTI DI INFORMATICA – APPELLO DEL 24 SETTEMBRE 2018 - COMPITO 1**

Cognome e Nome	Matricola	Aula	Postazione
----------------	-----------	------	------------

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

Implementare una applicazione console in linguaggio ANSI C per la gestione di archivio di film. L'archivio dei film memorizza le proiezioni di film in un alcuni cinema. Tali informazioni sono memorizzate in un file di testo (un esempio di file è riportato alla fine del testo) e vengono caricate in una lista di programmazioni. Successivamente, a partire da tale lista, viene costruito l'elenco dei cinema organizzato come una lista in cui ogni nodo riporta l'elenco dei film proiettati presso lo specifico cinema.

Specifica della struttura dati

Programmazione

- Titolo del film (stringa di 80 caratteri utili senza spazi, gli spazi sono sostituiti da un carattere \_)
- Nome del cinema (stringa di 15 caratteri utili senza spazi)
- DataInizioProgrammazione (stringa nel formato aaaammgg)
- DataFineProgrammazione (stringa nel formato aaaammgg)

Cinema

- Nome del cinema (stringa di 15 caratteri utili senza spazi)
- Elenco dei film proiettati presso il cinema (lista di Programmazioni)
- Numero dei giorni totali nei quali è proiettato un film presso il cinema (un intero)

Elenco delle operazioni/funzioni che devono essere implementate

(Lo studente noti che il non rispetto delle specifiche nei quesiti 1 e 2, quindi la scelta di un nome della funzione diverso, di parametri differenti o di valori di ritorno differenti sono considerati errori gravi a prescindere del fatto che la funzione svolga quanto richiesto).

1. Funzione di caricamento da file di testo delle presenze nella lista delle programmazioni. La funzione da implementare deve avere la seguente struttura:  
nome della funzione: *CaricaProgrammazioni*  
valore restituito: numero degli elementi caricati  
parametri: *nome del file, lista delle programmazioni*
2. Funzione di ricerca del numero di cinema presso cui è stato proiettato un dato film. La funzione da implementare deve avere la seguente struttura:  
nome della funzione: *ContaCinema*  
valore restituito: *0 se la lista delle presenze è vuota, 1 altrimenti*  
parametri: *lista delle presenze, titolo del film, numero di cinema in cui è stato proiettato il film*  
(NB: *numero di cinema* è un parametro di uscita che la funzione deve produrre e non visualizzare).
3. CreazioneElencoCinema: Crea la lista dei cinema a partire dalla lista delle programmazioni.
4. VisualizzaTutto: Visualizza, per ogni giorno, l'elenco dei film proiettati presso ciascun cinema utilizzando la lista dei cinema
5. CinemaPiu: La funzione deve fornire (come parametri d'uscita) il cinema che ha avuto più giorni di proiezioni e il numero dei giorni in cui in questo cinema è stato proiettato un qualsiasi film.
6. 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 dei dati nel file di ingresso

Star_trek	Alfieri	20180917	20180920
Mamma_Mia	CineStar	20180917	20180925
Star_trek	Ariston	20180917	20181020
Una_Storia_Senza_Nome	Cinestar	20180918	20181020
Lucia	Alfieri	20180918	20181025



**SEZIONE 2: DOMANDE TEORICHE (VALORE MAX 5 PUNTI)**

**DOMANDA 1 (PUNTI 0,5)**

Descrivere le differenze fra un compilatore e un interprete

**DOMANDA 2 (PUNTI 0,5)**

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

```
int main(void) {  
    int i = -5, k;  
    while (i++ <= 0) {  
        k = -i;  
        for (; k > 100; k *= 2)  
            printf("(%d)(%d)", i, k);  
        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>  
#include <string.h>  
  
int main(void) {  
    char s1[] = "pluto";  
    char s2[] = "paperino";  
    if (s2 > s1) printf("%s %s\n", s1, s2)  
    else printf("errore");  
}
```



**DOMANDA 4 (PUNTI 1,5)**

Si supponga di avere un vettore di interi contenente i seguenti valori: 2, 4, 8, 25, 33, 57, 68, 80, 99

Quante iterazioni vengono effettuate dall' algoritmo di ricerca binaria per cercare il valore 3? 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, 40, 8, 14, 41, 2, 7, 89

Schematizzare l'algoritmo di visita in ordine anticipato (preordine) e scrivere la sequenza di valori ottenuti quando si effettua la visita?



**FONDAMENTI DI INFORMATICA – APPELLO DEL 24 SETTEMBRE 2018 - COMPITO 2**

Cognome e Nome	Matricola	Aula	Postazione
----------------	-----------	------	------------

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

Implementare una applicazione console in linguaggio ANSI C per la gestione di un registro di classe. Il registro di classe memorizza le presenze degli studenti giorno per giorno e le eventuali note di condotta. Tali informazioni sono memorizzate in un file di testo (un esempio di file è riportato alla fine del testo) e vengono caricate in una lista di presenze. Successivamente, a partire da tale lista, viene costruito il registro giornaliero organizzato come una lista in cui ogni nodo riporta l'elenco degli studenti presenti in una specifica data ed il numero degli stessi.

Specifica della struttura dati

Presenza

- Cognome dell'alunno (stringa di 15 caratteri utili senza spazi)
- Nome dell'alunno (stringa di 15 caratteri utili senza spazi)
- Data (stringa nel formato aaaammgg)
- Nota (stringa di 63 caratteri utili senza spazi)

Registro

- Data (stringa nel formato aaaammgg)
- Numero di presenti (valore intero)
- Elenco dei presenti (lista di Presenza)

Elenco delle operazioni/funzioni che devono essere implementate

(Lo studente noti che il non rispetto delle specifiche nei quesiti 1 e 2, quindi la scelta di un nome della funzione diverso, di parametri differenti o di valori di ritorno differenti sono considerati errori gravi a prescindere del fatto che la funzione svolga quanto richiesto).

7. Funzione di caricamento da file di testo delle presenze nella lista delle presenze. La funzione da implementare deve avere la seguente struttura:

nome della funzione: *CaricaPresenze*  
valore restituito: *esito del caricamento (0 errore durante il caricamento o file non trovato, 1 altrimenti)*  
parametri: *nome del file, lista delle presenze*

8. Funzione di ricerca del numero di note di condotta attribuite ad uno. Una nota di condotta è rappresentata da una stringa diversa dalla stringa "N/A". La funzione da implementare deve avere la seguente struttura:

nome della funzione: *ContaNote*  
valore restituito: *0 se la lista delle presenze è vuota, 1 altrimenti*  
parametri: *lista delle presenze, cognome e nome dello studente, numero di note*  
(NB: *numero di note* è un parametro di uscita che la funzione deve produrre e non visualizzare).

9. CreazioneRegistro: Crea la lista del registro giornaliero a partire dalla lista delle presenze. Ogni elemento della lista del registro contiene una data, il numero complessivo di studenti e l'elenco degli studenti presenti in quella specifica data.
10. VisualizzaTutto: Visualizza, per ogni giorno, l'elenco dei presenti e le eventuali note.
11. GiornoConPiuNote: Dato un periodo di osservazione definito da una data di inizio e da una data di fine (parametri di ingresso della funzione), restituisce (come parametri di uscita) la data in cui sono state effettuate il maggior numero di note.
12. 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 dei dati nel file di ingresso

Giulio	Rossi	20180917	N/A
Giovanni	Russo	20180917	N/A
Carmelo	Esposito	20180917	disturba_la_classe
Francesca	Bianchi	20180918	N/A
Lucia	Romano	20180918	N/A



**SEZIONE 2: DOMANDE TEORICHE (VALORE MAX 5 PUNTI)**

**DOMANDA 1 (PUNTI 0,5)**

Descrivere i principali vantaggi dell'utilizzo dei linguaggi ad alto livello rispetto a quelli di basso livello

**DOMANDA 2 (PUNTI 0,5)**

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

```
int main(void) {  
    int i = -5, k;  
    while (i++ <= 0) {  
        k = -i;  
        for (; k > 0; --k)  
            printf("(%d)(%d)", i, k);  
        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>  
#include <string.h>  
  
int main(void) {  
    char s1[] = "pluto";  
    char s2[];  
    s2 = s1;  
    printf("%s %s\n", s1, s2);  
}
```



**DOMANDA 4 (PUNTI 1,5)**

Si supponga di avere un vettore di interi contenente i seguenti valori: 2, 4, 8, 25, 33, 57, 68, 80, 93, 99

Quante iterazioni vengono effettuate dall' algoritmo di ricerca binaria per trovare il valore 93? 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, 4, 28, 14, 4, 52, 67, 89

Schematizzare l'algoritmo di visita in ordine simmetrico e scrivere la sequenza di valori ottenuti quando si effettua la visita?