

**ARTICOLAZIONE INFORMATICA CLASSE 4 SEZIONE A**

**DISCIPLINA Informatica**

**CLASSE DI CONCORSO A041**

**DOCENTE Prof. POSTACCHINI FABRIZIO ITP Prof. ROMAGNOLI LUIGI**

**PROGRAMMA SVOLTO**

I File, file di testo, file binari, struttura sequenziale, puntatore a file. Gestione di file in C: Apertura di un file, Accesso ad un file, Chiusura di un file. FILE \*fp, fopen, modalità di apertura. Esempi file di testo. File standard di I/O. Lettura e Scrittura di file: per File di testo, Lettura/scrittura con formato: fscanf, fprintf, Lettura/scrittura di caratteri: fgetc, fputc, Lettura/scrittura di stringhe di caratteri: fgets, fputs, per File binari, Lettura/scrittura di blocchi: fread, fwrite. Accesso a file di testo: lettura/scrittura con Formato, esempi di uso fscanf e fprintf. Passaggio di file a funzioni.

Esempio in C di scrittura di un file di testo, esempio in C di lettura e stampa di un file di testo  
Accesso a file binari: Lettura/scrittura di blocchi, Lettura di file binari, fread, esempio: file binario contenente interi, scrittura di file binari, fwrite, il controllo effettuato mediante feof.  
Gestione completa di un file sequenziale binario con un certo tracciato record con un'interfaccia a menu per inserire, inserire ordinatamente, cancellare, modificare, visualizzare.

Esercitazione effettuata:

Un allevatore vuole automatizzare la gestione dei capi di bestiame della sua azienda dotandosi di un sistema di elaborazione che permetta di gestire:

- arrivo di un nuovo capo di bestiame - vendita e quindi scarico di un capo di bestiame - ricerca in base al codice - aggiornamento informazioni di un certo animale individuato in base al codice - stampa dei dati relativi ai capi presenti nell'allevamento.

Per ogni animale vengono memorizzate le seguenti informazioni:

- codice di identificazione - data di nascita (ggmmaaaa) - razza cui appartiene - peso - data prevista per la vendita

I record con queste informazioni sono memorizzati su un FILE sequenziale.

Fare analisi del testo ben dettagliata che giustifichi le scelte effettuate e la codifica C del programma.

Posizionamento in un file: funzioni fseek, ftell, rewind.

Come muoversi in un file binario mediante fseek: esempi

Trovare e usare la posizione assoluta all'intero di un file, esempio. Dimensione di un file. Calcolo del numero di byte di un file binario (e il numero di interi che contiene), mediante uso di fseek, ftell.

Tecnica di accesso diretto ad un file: campo chiave, file e relativo array di record contenente solo chiave e posizione del record nel file originario, copia del sottoinsieme del file nell'array di record corrispondente. Le modalità di apertura di file in aggiornamento rb+ e wb+. La cancellazione di un record mediante l'array ridotto usato per l'accesso diretto. Stampa ordinata di un file, mediante l'array ridotto usato per l'accesso diretto. Letto un nome e cognome, se presente aggiornare uno o più dei suoi campi del relativo record. Ricerca e modifica record scegliendo quale campo. Visualizzare il file in modo crescente di chiave (anche se in realtà resta immutato) sfruttando l'accesso diretto.

Liste semplici: concetto di lista, rappresentazione sequenziale statica, notazione grafica, nodi e archi, radice, rappresentazione collegata, inserimento in testa, intermedio, eliminazione della testa o di un intermedio, implementazione mediante puntatori in C, creazione di una lista semplice in C. Stampa di una lista di interi, funzioni di inserimento in testa alla lista e di stampa della lista, Stampa di una lista (ricorsiva). Operazioni su liste. Ricerca in una lista. Inserimento: inserimento in fondo ad una lista. Liste ordinate: inserimento ordinato in linguaggio C. Esercizio in C: Definire una funzione lunghezzaLista che data una ListaDiElementi, restituisca la sua lunghezza. Definire una funzione deallocaLista che riceve una ListaDiElementi e ne dealloca tutti gli elementi. (...free (l);). Modifica di un elemento di una lista. Chiave di ricerca. Univocità, Funzione di modifica tramite funzione di ricerca. Definizione di una funzione che data una ListaDiElementi, restituisca il puntatore al primo elemento pari nella lista (restituisce NULL se la lista e' vuota o non contiene elementi pari). Soluzione esercizio in forma iterativa ed in forma ricorsiva. Funzione lunghezza lista in forma iterativa ed in forma ricorsiva. Definizione di una procedura iterativa 'elimina' che ricevuta una ListaDiElementi e un intero X, elimini (senza deallocare) i primi X elementi e ritorni il puntatore alla testa della lista modificata. Definizione di una funzione ordinalista che modifica una ListaDiElementi data ordinandola in modo crescente. La funzione non deve usare allocazione dinamica della memoria ( malloc e free ), ne modificare il campo info degli elementi. La funzione restituisce il puntatore al primo elemento ottenuto dopo l'ordinamento. Costruzione del seguente programma in "C": lette in input due liste di numeri interi ognuna delle quali ordinata, costruire una terza lista di numeri interi ordinata, ottenuta mediante la "fusione" delle prime due. Stampare la lista.

Gestione Pila: concetti operazine di Push, operaione di Pop. Costruzione di un programma in C per la gestione dinamica di una pila di dati. Letta in input una sequenza di numeri interi positivi memorizzarla in una lista: costruire una seconda lista contenente soltanto gli elementi della prima lista che non siano numeri primi, stampare la seconda lista. Costruzione di un programma in C per la gestione di una coda dinamica.

Java: caratteristiche generali, ambienti di programmazione, struttura dei programmi, identificatori e parole chiave, variabili e costanti. Tipi di dato: primitivi, riferimento. il casting, operatori, commenti e documentazione. La gestione dell'I/O. Programma Java che calcola l'area di un rettangolo, Programma Java che calcola le potenze di un numero intero con anche uso di Math.pow() . Java: la sequenza, la selezione, operatori di confronto, operatori booleani, la ripetizione (while, do-while, for). Esempi svolti in Java. Java: strutture dati array, matrice. Le eccezioni. Esempi svolti. La programmazione orientata agli oggetti OOP, classi ed oggetti, attributi, metodi. struttura oggetto, diagramma delle classi, esempio classe automobile, UML, incapsulamento. Java: dichiarazione di una classe, istanza e dichiarazione di un oggetto, esempio della costruzione della classe cerchio per il calcolo dell'area conoscendo il raggio. Esecuzione del programma. Dichiarazione degli attributi, livelli di visibilità public, private, protected. Static, Final.

Classi e oggetti, orientamento agli oggetti e confronto con la programmazione procedurale, la classe, attributi, metodi, l'oggetto e l'istanza, incapsulamento, diagramma delle classi, UML, costruttore. Java: sintassi classe ed oggetti. Java: dichiarazione degli attributi, livelli di visibilità public, private, protected, static, final. Dichiarazione dei metodi, tipo del valore di ritorno, elenco dei parametri, livelli di visibilità di un metodo. Java: dettagli istanza di classe, dichiarazione oggetto, allocazione oggetto, costruttore, riferimenti nulli, uguaglianza tra oggetti. Utilizzo degli oggetti, this, attributi e metodi static. Information hiding, interfaccia. Esempi di loro realizzazione in Java. Programma Java per la creazione della classe frazione (per la riduzione ai minimi termini MCD). Programmi object-oriented in Java, programmi con più classi. Java: Array di oggetti. Ereditarietà: classe base, derivata, sottoclasse, superclasse, gerarchia di classi, grado di gerarchia, estensione, ridefinizione, ereditarietà singola e multipla, librerie. Dichiarazione in Java di una sottoclasse e suo utilizzo, esercizio della creazione della classe per il calcolo del volume di un Cilindro utilizzando la classe Cerchio già realizzata. Polimorfismo: overriding, overloading. Esempio svolto: creazione classe per gestire un conto corrente che permette il versamento e prelievo di importi limitati utilizzando una classe conto corrente già esistente. Librerie.

Alcuni ulteriori esercizi svolti:

Derivare dalla classe poligono regolare una classe prisma retto e sviluppare anche un semplice main per la gestione di oggetti di tipo poligono e di tipo prisma.

Con riferimento alla classe base creata nel problema della classe calcolatrice, derivare una calcolatrice per calcolare le seguenti funzioni: la somma dei quadrati dei due numeri, la differenza tra il quadrato del numero maggiore e il quadrato del minore.

Creazione della classe per il calcolo del volume di un Cilindro utilizzando una classe Cerchio già realizzata.

Costruzione in Java della classe per il programma che calcola il volume e la superficie di un parallelepipedo a base quadrata estendendo la classe di base Quadrato.

Costruire una classe per rappresentare uno studente, estendendo la classe base Anagrafica (già realizzata) con un nuovo attributo per la matricola. Ridefinire il metodo per la stampa dei dati, visualizzando anche l'informazione sulla matricola.

Costruire una classe per rappresentare un dipendente, estendendo la classe base Anagrafica, con gli attributi per lo stipendio e il livello (da 1 a 7). Prevedere un metodo per gestire l'incremento del livello: ad ogni scatto deve corrispondere un aumento del 10% dello stipendio.

Scrivere un programma Java per realizzare una classe "resistore" e una classe "generatore".

Successivamente sia previsto di istanziare n oggetti resistore (max 5) e un oggetto generatore.

Considerando in serie tutti i resistori, creare un menu di scelta per calcolare:

la resistenza totale RTOT;

la corrente I;

la tensione ai capi di ogni resistore.

Gli insegnanti

gli studenti

Fabrizio Postacchini .....  
Luigi Romagnoli .....

Fermo 4 giugno 2021