

## (A) CONOSCENZA TERMINOLOGICA

Dare una breve descrizione dei termini introdotti:

- Flat file
- Sequenza di inserimento
- File ausiliario
- File degli inserimenti
- Costo di elaborazione

## (B) CONOSCENZA E COMPETENZA

Rispondere alle seguenti domande producendo anche qualche esempio

B1) Conoscenza

1. Quali sono le *operazioni logiche* su file con accesso sequenziale?
2. Cosa è un *flat file*?
3. Quali sono i *tipi di accesso* su un file ad organizzazione sequenziale?
4. Quali sono i più comuni *criteri di ricerca*?
5. Cosa si intende con il termine *aggiornamento*?

B2) Competenza

1. In quale caso su un file ad organizzazione sequenziale si può avere *accesso diretto*?
2. In quale caso è utile avere un archivio *organizzato sequenzialmente*?

## (C) ESERCIZI DI COMPrensIONE

1. In un file sequenziale i record sono disposti in ordine ..... e le operazioni possibili sono: la ....., che può essere effettuata per ....., o per ..... della chiave, e gli aggiornamenti. Per aggiornamento, si intendono le operazioni di ....., ..... e .....
2. L'organizzazione sequenziale di un archivio conviene quando l'elaborazione riguarda tutti i record. Nel caso di archivi con frequenti aggiornamenti, l'organizzazione ..... è sconsigliata.
3. In un archivio organizzato sequenzialmente, l'inserimento di un record di chiave  $k$  si esegue con l'uso di un file ....., nel quale salvare temporaneamente i record durante l'elaborazione. Per l'inserimento di più record, si può ricorrere alla ..... Del file ..... con il file degli .....
4. Associare le proposizioni di sinistra con le corrispondenti sulla destra:

	1 Accodare un record...	A ... si effettua su file ordinati
	2 L'inserimento intermedio...	B ... non richiede file ausiliario
	3 La ricerca per posizione...	C ... può usare anche l'accesso sequenziale
	4 La ricerca per valore...	D ... è agevole con l'accesso diretto

5. Completare la seguente tabella indicando il costo di elaborazione della ricerca di un dato in un archivio di  $N$  record, nei vari casi:

	In caso di successo	In caso di insuccesso	In media
Archivio non ordinato			
Archivio ordinato			

6. Dato il file primario  $f$  e il file degli inserimenti  $f_i$ , indicare dopo l'aggiornamento come è composto il file  $f$ .

20150	Info
20152	Info
20153	Info
20160	Info
20167	Info
20200	Info
20201	Info

 $f$ 

20151	Info
20163	Info
20180	Info
20202	Info

 $f_i$ 

	Info
	Info
	Info
	Info
	Info
	Info
	Info
	Info
	Info
	Info
	Info
	Info

 $f$

7. Completare le parti mancanti del seguente algoritmo e descriverne il funzionamento.  
**Tipo Ricerca (Intero chiave, Stringa nomefile)**  
**Flusso f;**  
**Tipo r;**  
**Inizio**  
 ApriFileLettura (....., .....);  
 Leggi (f, r);  
 Fintantochè (..... E ..... ) **fai**  
 Leggi(f, r);  
 Se (EOF(f)) **Ritorna** .....;  
 Altrimenti **Ritorna** r;  
 Chiudi(f);  
**Fine;**
8. Completare le parti mancanti del seguente algoritmo e descriverne il funzionamento  
**Intero** ..... (**Tipo r, Stringa nomefile**)  
**Flusso f;**  
**Inizio**  
 Se la chiave *r.k* esiste **Ritorna** .....  
 Altrimenti  
**Inizio**  
 ApriFileScrittura (f, nomefile, true);  
 Scrivi (.....);  
 Chiudi (f);  
 Ritorna 0;  
**Fine**  
**Fine;**
- 9.

**(D) ESERCIZI DI APPLICAZIONE**

1. **Esercizio risolto.** Considerando un file ordinato *f*, formato da record di tipo *Rec*, memorizzato in *nomefile*, scrivere la procedura per realizzare l'inserimento ordinato di un record *r*.  
 Per scrivere la funzione richiesta, facciamo le seguenti osservazioni:
- trattandosi di un file organizzato sequenzialmente, l'inserimento richiede un file ausiliario *aux*;
  - dovendo mantenere l'ordinamento, si tratta di un inserimento, in genere, intermedio.
  - dovendo controllare la non esistenza di *r* in *f*, facciamo uso della funzione  
**Tipo Ricerca (Intero chiave, Stringa nomefile)**

La procedura richiesta può essere la seguente:

**Intero** Inserisci (**Tipo r, Stringa nomefile**)

**Inizio**

```

Tipo buf; // buffer per I/O
Flusso aux, f;
Se (Ricerca(r.chiave, nomefile)!=null) // se record è presente ...
  Ritorna -1; // ... termina inserimento
Altrimenti
  Inizio // ... altrimenti inserisce
  ApriFileLettura(f, nomefile); // apre file dati e ...
  ApriFileScrittura(aux, "temp", false); // ... ausiliario
  Leggi(f, buf);
  Fintantochè (buf.chiave<r.chiave) fai // fintantochè le chiavi sono più piccole
  Inizio
  Scrivi(aux, buf); // ... trascrive i record
  Leggi(f, buf);
  Fine
  Scrivi(aux, r); // inserisce il nuovo record
  Fintantochè(!Eof(f)) fai // copia restanti
  Inizio
  Scrivi(aux, buf);
  Leggi(f, buf);
  Fine
  Chiudi(f); Chiudi(aux);
  ApriFileLettura(aux, "temp"); // apre aux
  ApriFileScrittura(f, nomefile, false); // apre file dati in riscrittura
  Leggi(aux, buf);
  Fintantochè(!Eof(aux)) fai // ricopia tutto aux in f

```

```

Inizio
  Scrivi(f, buf);
  Leggi(aux, buf);
Fine
Ritorna 0;

```

**Fine****Fine.**

2. **Esercizio risolto.** Considerando un file ordinato  $f$ , formato da record di tipo  $Rec$ , memorizzato in  $nomefile$ , scrivere la procedura per realizzare la modifica di un record  $r$ , tramite la sua sostituzione con un record  $new$ . Per scrivere la funzione richiesta, facciamo le seguenti osservazioni:
- trattandosi di un file organizzato sequenzialmente, la sostituzione richiede un file ausiliario  $aux$ ;
  - si deve mantenere l'ordinamento;
  - dovendo controllare la non esistenza di  $r$  in  $f$ , facciamo uso della funzione

**Rec Ricerca (Intero chiave, Stringa nomefile)**

La procedura richiesta può essere la seguente:

```

Intero modifica (Rec r, Stringa nomefile, Rec new)
{
  Rec buf;
  Flusso aux, f;
  Se (Ricerca(r.chiave, nomefile)==null)
    Ritorna -1;
  Altrimenti
    Inizio
      ApriFileLettura(f, nomefile);
      ApriFileScrittura(aux, "temp", false);
      Leggi(f, buf);
      Fintantochè (buf.chiave<r.chiave) fai // trova la posizione
      Inizio
        Scrivi(aux, buf);
        Leggi(f, buf);
      Fine
      Scrivi(aux, new);
      Fintantochè(!Eof(f)) fai // copia restanti
      Inizio
        Scrivi(aux, buf);
        Leggi(f, buf);
      Fine
      Chiudi(f); Chiudi(aux);
      ApriFileLettura(aux, "temp");
      ApriFileScrittura(f, nomefile, false);
      Leggi(aux, buf);
      Fintantochè(!Eof(aux)) fai // ricopia aux in f
      Inizio
        Scrivi(f, buf);
        Leggi(aux, buf);
      Fine
      Ritorna 0;
    Fine

```

**Fine**

3. Scrivere l'algoritmo di ricerca di un record di chiave  $k$ , in un file  $f$ , organizzato sequenzialmente, con accesso sequenziale e ordinato rispetto alla chiave.
4. Scrivere l'algoritmo di inserimento di un singolo record di chiave  $k$ , in un file  $f$ , organizzato sequenzialmente, con accesso sequenziale e ordinato rispetto alla chiave.
5. Scrivere l'algoritmo per inserire i record presenti in un file degli inserimenti, in un file  $f$ , organizzato sequenzialmente, con accesso sequenziale e ordinato rispetto alla chiave.
6. Si vuole automatizzare la gestione di un archivio anagrafico, in cui ogni record deve riportare i seguenti dati di una persona:
 

a. codice fiscale;	e. cap;
b. cognome;	f. città di residenza;
c. nome;	g. provincia.
d. indirizzo;	
7. Un'azienda deve automatizzare il calcolo dello stipendio mensile dei dipendenti. Per ciascun dipendente sono riportati:
 

h. numero di matricola del dipendente;	k. numero di ore mensili straordinarie;
i. numero di ore mensili ordinarie;	l. retribuzione oraria per il lavoro straordinario;
j. retribuzione oraria per il lavoro ordinario;	m. percentuale di trattenuta fiscale.

Oltre agli algoritmi relativi alle operazioni utente, fornire un algoritmo che calcoli e stampi, per ogni dipendente, la retribuzione lorda e quella netta.

8. In un archivio di pezzi di ricambio per autoveicoli, ogni record contiene:
- marca;
  - tipo;
  - modello;
  - anno di costruzione;
  - codice del pezzo di ricambio;
  - descrizione del pezzo di ricambio, di vari tipi di autovetture.
- Oltre alle operazioni utente, si vuole implementare una procedura per reperire tutti i pezzi relativi ad un certo modello di autovettura (ad esempio "Fiat", "Tipo", "Cabrio") di un dato anno di fabbricazione.
9. Si devono gestire le visite dei pazienti di un'attività sanitaria. Si hanno a disposizione i seguenti campi:
- codice sanitario del paziente;
  - cognome;
  - nome;
  - codice del servizio offerto (chirurgia, otorino, oculistica, ecc);
  - data della prestazione;
  - codice del medico che ha effettuato la prestazione;
- Occorre organizzare il file con ordinamento rispetto al codice sanitario del paziente. Oltre alle operazioni utente, fornire le seguenti procedure:
- elenco dei pazienti afferenti ad un dato medico, di cui si immette il codice;
  - cancellazione logica di un paziente (a prestazione effettuata);
16. Si vuole realizzare un *text formatter*, ossia un programma che leggendo caratteri da un testo di input, crei in uscita un file di output con determinate caratteristiche di stampa. Per ottenere le caratteristiche desiderate, il testo di input viene creato inserendo delle stringhe particolari, dette *sequenza di controllo*, che sono formate da un punto (.) seguito da una coppia di lettere indicanti sinteticamente il comando da eseguire. Le sequenze di controllo vanno poste su una riga separata rispetto al testo e sono:

Sequenza	Default	Effetto
.pn n	1	Imposta il numero di pagina ad <i>n</i>
.br	no	Arresta l'elaborazione del testo
.ce n	1	Centra le successive <i>n</i> righe di testo
.fo s	vuota	Pone la stringa <i>s</i> come piè di pagina
.he s	vuota	Pone la stringa <i>s</i> come intestazione
.ls n	1	Imposta l'interlinea ad <i>n</i> linee
.pl n	66	Imposta il numero di linee per pagina ad <i>n</i>
.rm n	60	Imposta il margine destro a <i>n</i> caratteri
.lm n	0	Imposta il margine sinistro a <i>n</i> caratteri
.hl n	0	Imposta il numero di linee per l'intestazione
.fl n	0	Imposta il numero di linee per il piè di pagina
.un	false	Attiva la sottolineatura
.nu	true	Disattiva la sottolineatura

17. Un file sequenziale contiene i parametri corporei di alcuni atleti: in particolare, il volume del tronco  $V_t$  e la lunghezza degli arti  $L$ . La relazione:

$$I = (V_t)^{1/3} / L * 1000$$

esprime l'indice morfologico dell'atleta (un valore intero solitamente compreso tra 150 e 300), che influenza la sua performance sportiva in alcuni sport maggiormente che in altri. Si vuole calcolare: la media aritmetica  $M_a$  dei valori di  $I$ , la moda  $M$  della distribuzione (il valore che presenta la massima frequenza) e la mediana  $M_d$ , sapendo che:  $M_a - M = 3 * (M_a - M_d)$