

Documentazione.

Progetto su archivio dati in linguaggio C

PROGETTO

Autore: Alessio Sanna

Data: 11/08/2023

Obiettivo del progetto

Il programma in oggetto dovrà permettere la gestione e ricerca di studenti sulla base di determinati parametri.

Devono essere quindi permesse operazioni di inserimento, modifica, eliminazione, ricerca, ordinamento ed esportazione dei dati inerenti agli studenti memorizzati nel sistema.

INDICE.

1. Introduzione

- Contesto del progetto
- Scopo e obiettivo del progetto

2. Obiettivo del Progetto

- Descrizione delle funzionalità chiave del programma

3. Parte 1 - Definizione strutture ed enumerazioni

- Descrizione delle strutture RecordStudente e Data
- Descrizione dell'enumerazione Titolo
- File coinvolti: Main.c, as_library0.h, as_library0.c

4. Parte 2 - Test ed esecuzione

- Creazione di una variabile RecordStudente e assegnazione dei dati
- Stampa dei valori contenuti nella variabile
- File coinvolti: Main.c, as_library1.h, as_library1.c

5. Parte 3.1 - Inserimento Dati Studente

- Descrizione della funzione inserisciStudente
- Controlli da effettuare durante l'inserimento
- File coinvolti: Main.c, as_library2_a.h, as_library2_a.c

6. Parte 3.2 - Generazione Casuale Dati Studente

- Descrizione della funzione generaStudente
- Generazione casuale delle informazioni del record studente
- File coinvolti: Main.c, as_library2_b.h, as_library2_b.c

7. Parte 3.3 - Stampa Informazioni Studente

- Descrizione della procedura stampaStudente
- File coinvolti: Main.c, as_library2_c.h, as_library2_c.c

8. Parte 4 - Test ed Esecuzione

- Creazione di una variabile RecordStudente con assegnazione dei dati tramite acquisizione e casualmente
- Stampa dei valori contenuti nella variabile
- File coinvolti: Main.c

9. Parte 5 - Gestione elenco studenti

- Creazione di un vettore di studenti
- a. Descrizione della funzione `inserisciStudenti`
- b. Descrizione della funzione `generaStudenti`
- c. Descrizione della procedura `stampaStudenti`
- d. Descrizione della procedura `cercaStudente`
- e. Descrizione della procedura `calcolaEntrate`
- f. Descrizione della procedura `titoloStudio`
- g. Descrizione della procedura `stampaEdizione`
- File coinvolti: `Main.c`, `as_library3_c.h`, `as_library3_c.c`

10. Parte 6 - Test ed Esecuzione

- Avvio del programma, generazione dei dati e stampa
- File coinvolti: `Main.c`, `as_library3_c.h`, `as_library3_c.c`

11. Parte 7 - Gestione Cifratura e Stampa Ordinata

- Implementazione delle funzioni **`criptaStudente`** e **`decriptaStudente`**
- Implementazione delle procedure **`criptaDati`** e **`decriptaDati`**
- Implementazione della procedura **`stampaOrdineAlfabetico`**
- Implementazione della procedura **`stampaOrdineAnnoNascita`**
- Note sulla cifratura delle informazioni
- Presentazione dello stesso (min 5' e max 10') tramite registrazione dello schermo e commento delle varie parti salienti con demo dello stesso.

12. Note

- Spiegazione delle istruzioni di elaborazione progetto nei file header
- Note sulle istruzioni contenute nei file header e nel file `Main.c`
- Sigle di verifica e test

Comando codice parte 1.

Parte 1 - Definizione strutture ed enumerazioni

Nome struttura: RecordStudente

Ogni record registrato dovrà contenere (esclusivamente) le seguenti informazioni:

- a) nome (vettore char 64 caratteri, compreso terminatore)
- b) cognome (vettore char 64 caratteri, compreso terminatore)
- c) codiceFiscale (vettore char 17 caratteri, compreso terminatore)
- d) titoloStudio (enumerazione, vedi sotto)
- e) dataNascita (struttura, vedi sotto)
- f) edizione (int)
- g) inCorso (booleano)
- h) sommaPagata (float)

Nome struttura: Data

Ogni posizione dovrà contenere (esclusivamente) le seguenti informazioni:

- a) anno (int)
- b) mese (int)
- c) giorno (int)

Nome enumerazione: Titolo

L'enumerazione dovrà contenere i seguenti valori: diploma, laurea triennale, laurea magistrale, altro.

File parte 1:

Main. *Fare riferimento a :* Main.c

Librerie. *Fare riferimento a :* as_library0.h, as_library0.c

Comando codice parte 1 . *Fare riferimento a :* as_library0.h

Comando codice parte 2.

Parte 2 – Test ed esecuzione

Dichiarare nel main una variabile di tipo RecordStudente e fare una prima versione

assegnando i dati in fase di dichiarazione, una seconda versione

assegnando i dati con

l'operatore di assegnamento (campo per campo), una terza versione

mediante l'utilizzo della

funzione scanf().

Per ogni versione stampare i valori contenuti all'interno della variabile dichiarata nel main.

File parte 2:

Main. *Fare riferimento a :* Main.c

Librerie. *Fare riferimento a :* as_library1.h, as_library1.c

Comando codice parte 1 . *Fare riferimento a :* as_library1.h

Comando codice parte 3.

Parte 3.1

1. Creazione di una funzione chiamata **inserisciStudente** che permetta l'inserimento dei dati di uno studente. La procedura deve effettuare i seguenti controlli (direttamente nella procedura o utilizzando subroutine esterne – scelta consigliata):
 - Il nome e il cognome possono contenere solo spazi e lettere (maiuscole o minuscole);
 - Il codice fiscale deve contenere esattamente 16 caratteri alfanumerici maiuscoli;
 - il titolo di studio può essere scelta solo tra i valori dell'enumerazione;
 - la data di nascita deve rispettare i vincoli canonici (non si può inserire il 30 febbraio il 31 aprile... ma non fare caso agli anni bisestili!);
 - l'edizione deve essere maggiore di zero;
 - la sommaPagata non può essere negativa;

Comando codice parte 3.1 . Fare riferimento a : `as_library2_a.h`

Main. *Fare riferimento a :* `Main.c`

Librerie. *Fare riferimento a :* `as_library2_a.h`, `as_library2_a.c`

Parte 3.2

2. Creazione di una funzione chiamata **generaStudente** che generi casualmente tutte le informazioni relative al record dello studente seguendo le specifiche riportate al punto precedente e restituisca un elemento completo.

Comando codice parte 3.2 . *Fare riferimento a : as_library2_b.h*

Main. *Fare riferimento a : Main.c*

Librerie. *Fare riferimento a : as_library2_b.h, as_library2_b.c*

Parte 3.3

3. Creazione di una procedura chiamata **stampaStudente** che stampi tutte le informazioni, in modo ordinato e leggibile relative all'elemento passato come parametro.

File parte 3:

Comando codice parte 3.3 . *Fare riferimento a : as_library2_c.h*

Main. *Fare riferimento a : Main.c*

Librerie. *Fare riferimento a : as_library2_c.h, as_library2_c.c*

Comando codice parte 4.

Parte 4 – Test ed esecuzione

Dichiarazione nel main una variabile di tipo RecordStudente e fare una prima versione assegnando i dati tramite acquisizione e una seconda versione assegnando i dati casualmente;

per ogni versione stampare i valori contenuti all'interno della variabile dichiarata nel main tramite l'apposita procedura di stampa.

N.b.: Sarebbe bene iniziare da subito a suddividere il progetto in più file.

HINT: partire prima dalla generazione, implementando poi l'acquisizione e solo in ultimo aggiungere i controlli all'acquisizione.

File parte 4:

Comando codice parte 4.

Comando codice parte 4. *Fare riferimento a :* Main.c

Main. *Fare riferimento a :* Main.c

Comando codice parte 5.

Parte 5

- Creazione nel main di un vettore di 50 elementi chiamato “elenco” di tipo RecordStudente;

a- Creazione di una funzione chiamata **inserisciStudenti** che permetta l’inserimento dei dati di tutti gli studenti utilizzando la funzione “inserisciStudente”; la procedura deve ricevere in ingresso il vettore degli studenti e la sua dimensione.

b- Creazione di una funzione chiamata **generaStudenti** che generi casualmente tutte le informazioni relative a tutti gli studenti utilizzando la funzione “generaStudente”; la procedura deve ricevere in ingresso il vettore degli studenti e la sua dimensione.

c- Creazione di una procedura chiamata **stampaStudenti** che stampi tutte le informazioni di ogni singolo studente utilizzando la funzione “stampaStudente”; la procedura deve ricevere in ingresso il vettore degli studenti e la sua dimensione.

d- Creazione di una procedura chiamata **cercaStudente** che stampi tutte le informazioni dello studente il cui codice fiscale corrisponde a quello inserito dall’utente.

e- Creazione di una procedura chiamata **calcolaEntrate** che calcoli e stampi l’introito totale sommando le somme pagate da tutti gli studenti.

f- Creazione di una procedura chiamata **titoloStudio** che stampi il numero di studenti per ogni tipologia di titolo di studio.

g- Creazione di una procedura chiamata **stampaEdizione** che stampi tutte le informazioni di ogni singolo studente per l’edizione specifica utilizzando la funzione “stampaStudente”; la procedura deve ricevere in ingresso il vettore degli studenti e la sua dimensione nonché l’edizione di riferimento.

File parte :

Comando codice parte 5 . *Fare riferimento a :* as_library3_c.h

Main. *Fare riferimento a :* Main.c

Librerie. *Fare riferimento a :* as_library3_c.h, as_library3_c.c

Comando codice parte 6.

Parte 6 – Test ed esecuzione.

Avviare il programma, generare i dati e stamparli.

Comando codice parte 6 . *Fare riferimento a :* as_library3_c.h

Main. *Fare riferimento a :* Main.c

Librerie. *Fare riferimento a :* as_library3_c.h, as_library3_c.c

Comando codice parte 7.

- Implementata una funzione **criptaStudente** che cripti nome, cognome e codice fiscale di una variabile RecordStudente data in ingresso restituisca la sua versione criptata.
 - Implementata una funzione **decriptaStudente** che decripti nome, cognome e codice fiscale di una variabile RecordStudente data in ingresso restituisca la sua versione criptata.
 - Implementata una procedura **criptaDati** che cripti tutti gli studenti utilizzando la funzione “criptaStudente”; la procedura deve ricevere in ingresso il vettore degli studenti e la sua dimensione.
 - Implementata una procedura **decriptaDati** che decripti tutti gli studenti utilizzando la funzione “decriptaStudente”; la procedura deve ricevere in ingresso il vettore degli studenti e la sua dimensione.
 - Creazione di una procedura chiamata **stampaOrdineAlfabetico** che stampi tutte le informazioni di ogni singolo studente in ordine alfabetico utilizzando la funzione “stampaStudente”; la procedura deve ricevere in ingresso il vettore degli studenti e la sua dimensione.
 - Creazione di una procedura chiamata **stampaOrdineAnnoNascita** che stampi tutte le informazioni di ogni singolo studente dal più piccolo al più grande utilizzando la funzione “stampaStudente”; la procedura deve ricevere in ingresso il vettore degli studenti e la sua dimensione.
- Note cifratura: le ‘a’ diventano ‘z’, le ‘b’ diventano ‘y’... le ‘y’ diventano ‘b’, le ‘z’ diventano ‘a’.

NOTE.

- Nei file header del progetto possono essere contenute le istruzioni di elaborazione progetto all'interno di sezioni commento in verde . Queste si potranno trovare seguendo le indicazioni nelle sezioni " File parte n / Comando codice parte n " della presente documentazione.
- Le istruzioni sono state inserite all'interno dei file header solo ai fini orientativi; in caso di progettazione di un qualsiasi software si sarebbe valutata una scelta diversa per motivi di efficienza/memoria.
- La dicitura contenuta nelle istruzioni del programma nei file header.h o nel file main.c :

```
/** Test eseguiti: ok **/
```

starà ad indicare che il programmatore ha effettuato dei test di verifica di corretto funzionamento dei processi con effetto di successo.
- Nella funzione **inserisciStudente()** si potrà verificare un' impossibilità di verifica di corretto dato inserito al campo struct Recordstudente.**inCorso**. Questo è dovuto alla definizione del dato **inCorso** come variabile booleana. In questo modo all'inserimento da runtime di valori minori di 0 si otterrà sempre e comunque uno 0, così anche all'inserimento di valori maggiori di 1 si otterrà sempre un 1. In questo modo , la funzione di controllo **controllo_inCorso()** non potrà mai ricevere valori oltre questi range e il controllo non potrà mai davvero essere effettuato: se l'utente inserisce un valore al di fuori del range 0/1 verrà comunque acquisito un **true** o un **false**.
Vedi file: as_library2_a.c