

Esercizi per l'ottavo laboratorio

Antonio Miele Ettore Speziale Michele Tartara

30 novembre 2010

2 dicembre 2010

3 dicembre 2010

In questo laboratorio dovrete utilizzare i sottoprogrammi per organizzare e riutilizzare il codice scritto. Per questo motivo, assieme ad un esercizio nuovo, proponiamo una variante dell'esercizio della rubrica, con l'obiettivo di svolgerlo di nuovo utilizzando i sottoprogrammi e mostrare come la soluzione risulti più chiara ed immediata.

Vocabolario inglese-italiano - dictionary

Scrivere un programma per la gestione di un vocabolario inglese-italiano. Il vocabolario è composto da una serie di parole inglese (massimo 50) e per ciascuna parola fornisce la sua traduzione in italiano. Il dizionario elettronico permette di effettuare le seguenti operazioni:

- Inserire una nuova parola inglese con la relativa traduzione in italiano. Si noti che non ci possono essere parole inglesi duplicate; perciò, quando si inserisce una parola già presente, il programma deve chiedere se sostituire la precedente traduzione o meno.
- Visualizzare tutte le parole contenute con le relative traduzioni.
- Cercare una parola inglese e visualizzarne la traduzione.
- Eliminare una parola dal dizionario (Leggere il suggerimento su come implementare l'eliminazione riportato nelle note).
- Salvare il dizionario su file.
- Caricare il dizionario da file.

Note

- Si richiede di strutturare il codice utilizzando sottoprogrammi.
- La parola inglese e la sua traduzione italiana possono essere memorizzate utilizzando una struttura composta da due stringhe (lunghe al massimo 20 caratteri). Il dizionario è quindi rappresentato tramite un array di queste voci.

elemento	house	car	pen	table	...	numVoci = 4
posizione	0	1	2	3	...	

↓

elemento	house	table	pen	...	numVoci = 3
posizione	0	1	2	...	

Figura 1: Eliminazione della parola “car”.

- Si consiglia di definire altri sottoprogrammi di supporto come, per esempio, `mostraMenu()`.
- Si ricorda che la libreria standard C fornisce la funzione

```
int strcmp(char str1[], char str2[])
```

che restituisce 0 se due stringhe sono uguali, e la funzione

```
strcpy(char destination[], char source[])
```

che copia nel primo array la stringa contenuta nel secondo array.
- Si consiglia la definizione di un sottoprogramma `cercaVoce()` che riceve in ingresso l’array con il contenuto del dizionario, il numero di parole contenute nel dizionario e una parola inglese da cercare e restituisce la posizione dell’array in cui è memorizzata la parola. Nel caso la parola non è stata trovata la funzione restituisce -1.
- Per eliminare una voce dal dizionario la si può sovrascrivere con l’ultima parola valida e decrementare di uno il numero di voci presenti nel dizionario; nel caso la parola da eliminare è l’ultima valida basta decrementare di uno il numero di voci presenti nel dizionario. Nell’esempio nella Figura 1, si mostra come eliminare da un dizionario contenente 4 voci (la traduzione italiana è omessa per semplicità) la parola “car” che si trova in posizione 1: l’ultima parola valida (“table” memorizzata nella posizione 3) è copiata nella posizione 1; infine il numero di parole contenute è decrementato di uno.
- Si consiglia di far restituire il numero aggiornato di parole contenute nel dizionario alle funzioni `aggiungiVoce()` e `eliminaVoce()`.

Rubrica telefonica

Scrivere un programma che gestisce una rubrica. Essa è composta da un insieme di elementi, ognuno dei quali rappresenta una persona (vedi Tabella 1). La rubrica contiene, al massimo, i dati di 30 persone e permette di effettuare le seguenti operazioni (a scelta dell’utente):

- Inserire i dati di una nuova persona. Attenzione: non possono esserci degli omonimi, quindi se l’utente cerca di aggiungere un utente con lo

Tabella 1: caratteristiche di una persona

Campo	Tipo di Dato	Note
nome	stringa	massimo 16 caratteri
cognome	stringa	massimo 16 caratteri
telefono	stringa	massimo 8 caratteri

stesso nome e cognome di un altro già presente nella rubrica, il programma deve chiedere se sovrascrivere o meno il vecchio numero di telefono. Inoltre va controllato che il numero di telefono contenga solo caratteri che rappresentano numeri (0, 1, ...,9).

- Ricerca e stampa a video del numero di telefono in base al nome e al cognome.
- Eliminare una voce nella rubrica specificando il nome ed il cognome.
- Salvare su file la rubrica.
- Caricare da file la rubrica.
- **(facoltativo)** Ricerca e stampa a video del numero di telefono in base al cognome.

Note

- Si richiede di strutturare il codice utilizzando sottoprogrammi.
- E' possibile rappresentare una persona tramite un tipo di dato strutturato con le caratteristiche elencate in Tabella 1; la rubrica sarà quindi memorizzata in un array mentre il numero di persone effettivamente contenute può essere memorizzato in una variabile intera.
- Si consiglia di definire altri sottoprogrammi di supporto come, per esempio, `mostraMenu()`, `numeroValido()`.
- Si ricorda che la libreria standard C fornisce la funzione `int strcmp(char str1[], char str2[])` che restituisce 0 se due stringhe sono uguali, e la funzione `strcpy(char destination[], char source[])` che copia nel primo array la stringa contenuta nel secondo array.
- Si consiglia la definizione di un sottoprogramma `cercaVoce()` che riceve in ingresso l'array con il contenuto della rubrica, il numero di voci contenute, il nome ed il cognome della persona da cercare. La funzione restituisce la posizione dell'array in cui è memorizzata la parola. Nel caso la persona non è stata trovata la funzione restituisce -1.
- Per eliminare una voce dalla rubrica seguire lo stesso consiglio dato per l'esercizio precedente.

- Si consiglia di far restituire il numero aggiornato di voci contenute nella rubrica alle funzioni `aggiungiVoce()` e `eliminaVoce()`.