

Laboratorio 11 - File

Marco Alberti



**Dipartimento
di Matematica
e Informatica**

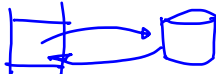


**Università
degli Studi
di Ferrara**

Programmazione e Laboratorio, A.A. 2021-2022

Ultima modifica: 8 dicembre 2021

Attenzione! Questo materiale didattico è per uso personale dello studente ed è coperto da copyright.
Ne sono vietati la riproduzione e il riutilizzo anche parziale, ai sensi e per gli effetti della legge sul diritto d'autore.



Creazione rubrica

Nella seguente tabella sono riportati i dati di una rubrica telefonica, in cui ogni persona è rappresentata da

- nome (stringa di max 20 caratteri senza spazi, incluso terminatore)
- indirizzo (stringa di max 30 caratteri senza spazi, incluso terminatore)
- numero (stringa di max 15 caratteri senza spazi, incluso terminatore)

STRUCT
Persona

TELEFONO

0532 423456
532123456

Nome	Indirizzo	Telefono
Mario_Rossi	Via_Roma_15	333_1234567
Giulia_Bianchi	Piazza_Milano_23	347_7654321
Elisa_Verdi	Via_Italia_44	348_11122233
Maria_Russo	Viale_Cavour_18	320_3849234
Carlo_Esposito	Via_Mazzini_25	380_98765432

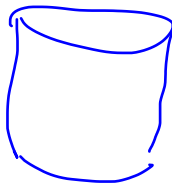
Scrivere un programma che crei un file di testo di nome **rubrica.txt** e un file binario di nome **rubrica.dat**, entrambi contenenti questi dati.

~~***~~

M	a	r	i	o	_	R	o	s	s	i	
---	---	---	---	---	---	---	---	---	---	---	--

Memoria

1000	Mario Rossi
	Via Roma 15
1006	333 .



Ricerca in rubrica

Per ognuno dei due file creati all'esercizio precedente, si scriva un programma che

FILE DI RECORD

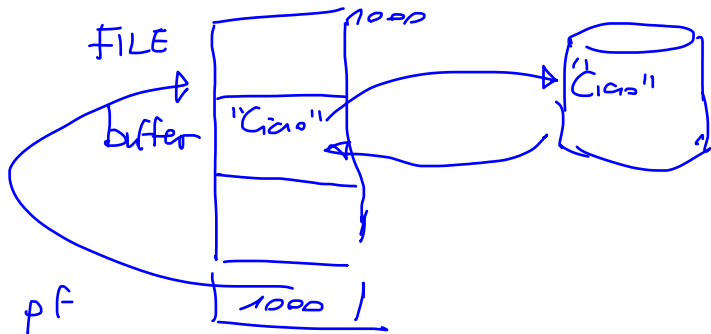
- 1 legga da tastiera un nome;
- 2 cerchi la persona corrispondente, cioè quella con nome uguale alla stringa data in input, nel file della rubrica;
- 3 visualizzi sullo schermo i dati della persona (se trovata).

Variante: mostrare i dati delle persone per cui la stringa in input è contenuta nel nome (ad esempio quelli di **Mario_Rossi** se l'input è **Mario**).



fflush()

BUFFERED I/O

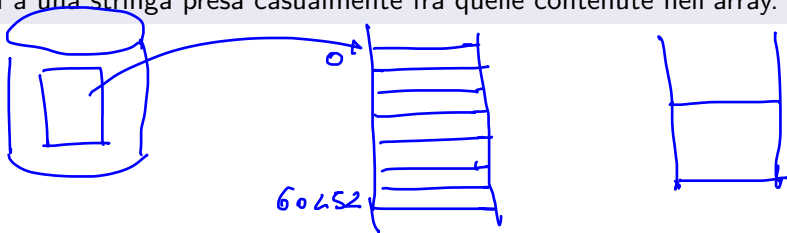


fprintf(pf, "Ciao")

Parola casuale

Scrivere un programma che stampi a video una parola presa casualmente dal file allegato `111/words.italian.txt`. In particolare:

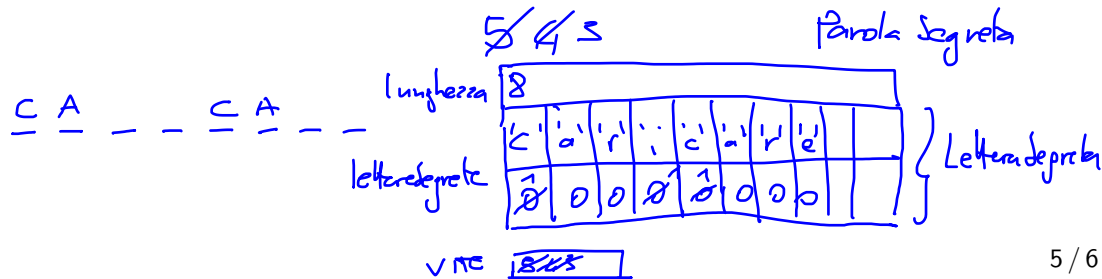
- definire un array in grado di contenere al massimo 100.000 stringhe di 31 caratteri (compreso il terminatore);
- caricare nell'array il contenuto del file *utilizzando la redirection dell'input*;
- definire e usare una procedura di nome **parolaCasuale** che imposti uno dei suoi parametri a una stringa presa casualmente fra quelle contenute nell'array.



Indovina

Si scriva un programma che richiede all'utente di indovinare una parola presa a caso dal vocabolario italiano.

Per indovinare la parola, l'utente inserisce una lettera ad ogni turno. Se la lettera appartiene alla parola, il programma visualizza le lettere già indovinate della parola, nelle relative posizioni. Altrimenti (se la lettera non compare nella parola), il programma segnala l'errore. Dopo 5 errori, l'utente perde. Se, invece, riesce ad indovinare tutte le lettere della parola, vince.





- Rappresentare la parola da indovinare con una struttura di nome **ParolaSegreta**, contenente due campi:
 - un intero contenente la lunghezza della parola
 - un array di strutture di nome **LetteraSegreta**, una per ogni lettera della parola, contenente la lettera stessa e un flag che indica se è già stata indovinata o meno.
- Definire e usare una funzione di prototipo **ParolaSegreta**
nuovaParolaSegreta(char s[]) che imposti una parola segreta con le lettere della stringa **s**, tutte non indovinate.
- Definire e usare una funzione di prototipo **int contiene(ParolaSegreta *p, char c)** che restituisce **1** se ***p** contiene **c** e **0** altrimenti, modificando ***p** per impostare le eventuali nuove lettere indovinate.
- Dopo ogni tentativo, stampare la parola indicando con **'_'** le lettere non ancora indovinate e mostrando quelle già indovinate (ad esempio **_e__a_a** se la parola è **ferrara** e sono state indovinate le lettere **a** ed **e**).