

Laboratorio 1: Bash Programming

Lo scopo di questo laboratorio è di creare una semplice utility che permetta di gestire una **rubrica** (Address book) da riga di comando.

I dati della rubrica sono salvati in un database contenuto in un file **CSV** dal nome `address-book-database.csv` come quello fornito in esempio.

L'utente interagisce con la rubrica con il programma `Bash address-book.sh` fornendo diversi comandi per visualizzare, ricercare, inserire o cancellare voci dalla rubrica.

Di seguito l'elenco delle funzionalità che la rubrica deve implementare.

1. Visualizzazione

address-book.sh view

Stampa il contenuto della rubrica, allineando i campi in modo che i campi della stessa colonna siano allineati. Si ordinino le voci per mail, ricordando che la prima riga deve essere l'header.

A titolo di esempio, l'output per il database di esempio deve essere

name	surname	phone	mail	city	address
andrea	verdi	3426875623	andrea@units.it	monfalcone	via roma 11
luca	rossi	334515642	luca@libero.it	trieste	via giulia 23
paolo	bianchi	3124568123	paolo@virgilio.it	trieste	via milano 53

Suggerimento: si considerino i comandi `column`, `head` e `tail`.

Nota: tutte le entry del database devono essere visualizzate (non solamente le prime)

2. Ricerca

address-book.sh search <string>

Si stampino le voci che contengono la stringa passata come argomento. Si ricerchi la stringa in ogni campo delle voci. L'output deve avere il seguente formato.

\$address-book.sh search trieste

```
Name: luca
Surname: rossi
Phone: 334515642
Mail: luca@libero.it
City: trieste
Address: via giulia 23
```

```
Name: paolo
Surname: bianchi
Phone: 3124568123
Mail: paolo@virgilio.it
City: trieste
```

Address: via milano 53

In caso nessuna voce contenga il parametro di ricerca, il programma deve stampare Not found.

Suggerimento: si consideri il comando grep e si veda l'utilizzo del ciclo for in bash e il ruolo della variabile speciale shell IFS. (si consiglia di settarla IFS=\$'\n')

3. Inserimento

address-book.sh insert

Inserimento di una nuova voce in rubrica. Il programma deve richiedere all'utente di inserire i vari campi tramite tastiera.

Nota: in caso l'indirizzo mail risulti già presente, il programma non deve effettuare l'inserimento e stampare un messaggio di errore.

Il programma deve stampare quanto segue:

\$address-book.sh insert

Name: **giovanni**
Surname: **pautasso**
Phone: **3124531285**
Mail: **giovanni@gmail.com**
City: **aurisina**
Address: **via venezia 23**
Added

Il testo in nero è stampato dall'applicazione, mentre quello in rosso è inserito dall'utente.

Suggerimento: si consideri il comando read e il parametro -n del comando echo. Si presti attenzione a verificare correttamente che la mail non sia presente (si veda ad esempio l'uso di grep -x)

4. Cancellazione

address-book.sh delete <mail>

Cancella la voce associata alla mail fornita come argomento. In caso di cancellazione avvenuta con successo stampa Deleted mentre se nessuna voce è associata alla mail stampa Cannot find any record.

Nota: si assicuri di non cancellare l'header del file CSV in caso di input errato.

Istruzioni per la consegna

Il programma fornire un output con lo **stesso esatto formato** usato negli esempi.

Il programma deve poter lavorare con qualsiasi dato di input. La correzione consisterà nell'esecuzione automatizzata del programma con un input generato dal docente.

Per la consegna del programma, usare la funzione *Assignments* su Microsoft Teams, facendo l'upload di **un singolo file** nell'*assignment* relativo a questa esercitazione. Nel caso l'esercitazione richieda un programma in C, va consegnato il file sorgente e non il programma compilato.