

## Esercizio 1

Definire un nuovo tipo chiamato BOOLEAN che può assumere solo i seguenti valori: TRUE e FALSE. Implementare quindi le seguenti operazioni:

- AND\_LOGICO: ha due parametri di tipo BOOLEAN e restituisce il risultato come un valore di tipo BOOLEAN
- OR\_LOGICO: ha due parametri di tipo BOOLEAN e restituisce il risultato come un valore di tipo BOOLEAN

Realizzare un programma che permetta all'utente di inserire due valori BOOLEAN (attraverso l'inserimento ad esempio del carattere 't' per TRUE e 'f' per FALSE) e quindi richiamare le due operazioni implementate stampando a video il risultato.

### Esempio di esecuzione

*Inserisci un valore boolean (t/f): t*

*Il valore inserito e' VERO*

*Inserisci un valore boolean (t/f): f*

*Il valore inserito e' FALSO*

*Il risultato di AND LOGICO e' FALSO*

*Il risultato di OR LOGICO e' VERO*

## Esercizio 2

Si scriva un programma che consenta di gestire l'insieme degli utenti di un sistema di messaggistica istantanea.

Ogni utente è caratterizzato da:

- Username (stringa)
- Indirizzo di posta elettronica (stringa)
- Numero di collegamenti effettuati (intero), inizialmente valorizzato a 0
- Durata totale in minuti dei collegamenti effettuati (intero), inizialmente valorizzato a 0
- Stato di connessione (tipo enumerativo con valori "online" o "offline"), inizialmente vale "offline"

Il programma deve gestire gli utenti utilizzando una struttura dati adeguata che permetta di memorizzare 10 elementi al massimo.

Realizzare le seguenti funzioni:

- Inserimento di un nuovo utente, quindi lettura da tastiera di username e email, il resto dei valori sono già inizializzati come illustrato in precedenza
- Login di un utente già registrato: il suo stato passa da offline a online
- Logout di un utente già registrato: il suo stato passa da online a offline, viene incrementato il numero di collegamenti effettuati e viene chiesto all'utente di inserire il tempo di connessione da aggiungere alla durata totale
- Cancellazione di un utente data la sua username
- Stampa della username di tutti gli utenti online in quel momento
- Stampa di tutti i dati dell'utente che ha la durata media più alta relativamente ai collegamenti effettuati

Infine implementare un menù di scelta che permette di utilizzare le funzioni a disposizione.

### Esempio di esecuzione

*1. inserisci un utente*

*2. login utente*

*3. logout utente*

*4. cancella un utente*

*5. stampa tutti gli utenti collegati*

*6. stampa l'utente con la durata media più elevata di connessioni*

*7. esci*

*Inserisci la tua scelta: 1*

*Inserisci lo username: Bart*

*Inserisci l'email: bart@simpson.com*

- 1. inserisci un utente*
- 2. login utente*
- 3. logout utente*
- 4. cancella un utente*
- 5. stampa tutti gli utenti collegati*
- 6. stampa l'utente con la durata media più elevata di connessioni*
- 7. esci*

*Inserisci la tua scelta: 2*

*Inserisci la username dell'utente che si sta connettendo: Bart*

*Utente Bart connesso*

- 1. inserisci un utente*
- 2. login utente*
- 3. logout utente*
- 4. cancella un utente*
- 5. stampa tutti gli utenti collegati*
- 6. stampa l'utente con la durata media più elevata di connessioni*
- 7. esci*

*Inserisci la tua scelta: 3*

*Inserisci la username dell'utente che si sta disconnettendo: Bart*

*Inserisci per quanti minuti è stato connesso: 10*

*Utente Bart disconnesso, in totale ha effettuato 1 connessione per un totale di 10 minuti*

*.....*

### **Esercizio 3**

Si sviluppi un programma che sia in grado di memorizzare gli orari delle lezioni gestendo i dati:

nome corso: stringa di al massimo 20 caratteri

giorno della settimana: 1 cifra da 0 a 6

orario di inizio: stringa (oppure struct per rappresentare la coppia ora-minuto)

orario di fine: stringa (oppure struct per rappresentare la coppia ora-minuto)

Un corso potrà avere più lezioni nell'arco della settimana (al massimo 4), e non sarà possibile avere più di 5 corsi in totale.

Lo studente dovrà quindi definire una opportuna struttura dati per la gestione delle informazioni e quindi implementare il programma che proponga un menù per:

- Inserire un corso (controllando di inserirne al massimo 5)
- Scelto un corso, inserire l'orario di una lezione (controllando di non inserirne più di 4 per lo stesso corso)
- Scelto un corso, stampare il relativo orario settimanale

#### Esempio di esecuzione

*1. inserisci corso*

*2. inserisci orario per corso*

*3. stampa orario per corso*

*4. esci*

*scelta: 1*

*nome nuovo corso :informaticaB*

*1. inserisci corso*

*2. inserisci orario per corso*

*3. stampa orario*

*4. esci*

*scelta: 2*

*nome corso :informaticaB*

*giorno della settimana (dom=0, .., sab=6): 2*  
*orario inizio: 8.15*  
*ora fine: 11.15*

- 1. inserisci corso*
- 2. inserisci orario per corso*
- 3. stampa orario*
- 4. esci*

*scelta: 2*  
*nome corso :informaticaB*  
*giorno della settimana (dom=0, .., sab=6): 4*  
*orario inizio: 14.15*  
*ora fine: 17.15*

- 1. inserisci corso*
- 2. inserisci orario per corso*
- 3. stampa orario*
- 4. esci*

*scelta: 3*  
*nome corso :informaticaB*  
*Corso: informaticaB*  
*lezione: 2, orario inizio: 8.15, orario fine: 11.15*  
*lezione: 4, orario inizio: 14.15, orario fine: 17.15*