



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Fondamenti di Informatica T

Lista di Orari

Esercizio: Lista di Orari

Si vuole realizzare un programma per gestire una lista di orari

Ogni orario è caratterizzato dalle seguenti informazioni

- hh: un intero (ore)
- mm: una intero (minuti)

Si definisca un tipo struttura atto a contenerlo

- Si usi "typedef" per creare un alias con il nome "orario"



Esercizio: Lista di Orari

Si definiscano le funzioni:

```
void leggi_orario(orario *o);
```

- Che legga da terminale le informazioni su un orario...
- ...E li inserisca nella struttura passata come parametro

```
void stampa_orario(orario o)
```

- Che stampi a terminale le informazioni su un orario

Si scriva un main di prova che legga e stampi un orario



Esercizio: Lista di Orari

Si definisca la funzione:

```
int confronto_orario(orario o1, orario o2)
```

- Che confronti due strutture orario o1 ed o2 e restituisca:
 - 0 se sono uguali
 - un valore *negativo* se o1 precede o2
 - un valore *positivo* se o1 segue o2
- Si estenda il main in modo da leggere un secondo orario e:
 - Inserire tale orario e quello dell'esercizio precedente
 - ...In un array ordinato (Suggerimento: basta un if!)
 - Stampare a video il contenuto dell'array



Esercizio: Lista di Orari

Si definisca la funzione:

```
void inserisci_orario(orario *l, int *n, orario o)
```

Che, dato un orario "o"

- Lo inserisca nell'array ordinato "l"
- In una posizione tale da mantenere l'ordinamento di "l"
- ...E aggiorni la dimensione logica di "l" (passata per riferimento)

Si realizzi un main di prova che:

- Legga ed inserisca tre orari nell'array del passo precedente
- Stampi a video il nuovo contenuto dell'array





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Fondamenti di Informatica T

Omonimie

Esercizio: Omonimie

Si vuole realizzare un programma per gestire una lista di persone

Ogni persona è caratterizzata dalle seguenti informazioni

- nome: una stringa senza spazi di al più 255 caratteri
- cognome: una stringa senza spazi di al più 255 caratteri

Si definisca un tipo struttura atta a contenerla

- Si usi "typedef" per creare un alias con il nome "persona"



Esercizio: Omonimie

Si definiscano le funzioni:

```
void leggi_persona(persona *p);
```

- Che legga da terminale le informazioni su una persona...
- ...E li inserisca nella struttura passata come parametro

```
void stampa_personone(persona *l, int n);
```

- Che stampi a terminale le informazioni su una persona

Si scriva un main che legga e stampi un array di persone



Esercizio: Omonimie

Si definisca la funzione:

```
int uguali(persona *p1, persona *p2);
```

- Che confronti due persone (passate per riferimento)
- ...E restituisca "vero" se sono uguali ("falso" altrimenti)

Si scriva un main che confronti le prime due persone dell'array costruito in precedenza



Esercizio: Omonimie

Si definisca la funzione:

```
int presente(persona *lista, int n, persona p);
```

- Che, data una persona "p"
- ...Verifichi se ve ne sia una omonima nell'array "lista", di dimensione "n"
- La funzione deve restituire "vero" in caso affermativo

Si scriva un main che:

- Legga da terminali i dati di una nuova persona
- ...E ne verifichi la presenza nell'array precedente



Esercizio: Omonimie

Si definisca la funzione:

```
void trova_unici(persona *lista, int n,  
                 persona *unici, int *nu);
```

- Che, data un'array di persone "lista" di dimensione "n"
- ...Riempia un array (preallocato) "unici" con il contenuto di "lista" *a meno dei duplicati*.
- La dimensione logica di "unici" è memorizzata in "nu" (passato per riferimento)

Si usi la funzione per rimuovere i duplicati dall'array precedente





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Fondamenti di Informatica T

Minuscole

Esercizio: Minuscole

Si definisca la funzione:

```
char carattere_minuscolo(int c);
```

Che, dato un carattere "c"

- Se il carattere rappresenta ad un lettera maiuscola...
- ...Ne restituisca la corrispondente minuscola

Nota:

- Per convertire una lettera maiuscola in minuscola...
- ...Basta sottrarre ('A' - 'a'), i.e. la distanza tra la sequenza delle codifiche maiuscole e minuscole



Esercizio: Minuscole

Si definisca la funzione:

```
char carattere_minuscolo(int c);
```

Che, dato un carattere "c"

- Se il carattere rappresenta ad un lettera maiuscola...
- ...Ne restituisca la corrispondente minuscola
 - Per convertire una lettera maiuscola in minuscola...
 - ...Basta sottrarre ('A' - 'a'), i.e. la distanza tra la sequenza delle codifiche maiuscole e minuscole

Si collaudi la funzione nel main



Esercizio: Minuscole

Si definisca la funzione:

```
void stringa_minuscola(char *src, char *dst);
```

Che, data una stringa "src"

- La copi nella stringa "dst", con le maiuscole convertite in minuscole

Si collaudi la funzione nel main



Esercizio: Minuscole

Si scriva un programma che:

- Accetti input sotto forma di argomenti del main
- Stampi a video gli argomenti, convertiti in formato minuscolo
 - Gli argomenti stampati devono essere separati da ", "





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Fondamenti di Informatica T

Somma di Orari

Esercizio: Somma di Orari

Si vuole realizzare un programma per sommare orari

Ogni orario è caratterizzato dalle seguenti informazioni

- hh: un intero (ore)
- mm: una intero (minuti)

Si definisca un tipo struttura atto a contenerlo

- Si usi "typedef" per creare un alias con il nome "orario"



Esercizio: Somma di Orari

Si definisca la funzione:

```
orario leggi_orario(char *s);
```

Che, data una stringa "s", nel formato "hh:mm"

- Legga *dalla stringa* le informazioni sull'orario
- In caso di errori di lettura, si stampi un messaggio e si interrompa il programma

Si collaudi la funzione nel main



Esercizio: Somma di Orari

Si definisca la funzione:

```
orario somma_orari(orario o1, orario o2);
```

Che, dati due orari o1 ed o2

- Calcoli e restituisca la loro somma
- Si gestiscano correttamente i riporti, e.g.:
 - $12:34 + 0:40 = 13:14$
 - $23:00 + 5:00 = 4:00$

Si collaudi la funzione nel main



Esercizio: Somma di Orari

Si scriva un programma che:

- Accetti input sotto forma di argomenti del main
- Ogni argomento deve essere un orario, nel formato "hh:mm"
- Calcoli e stampi a video la somma degli orari





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Fondamenti di Informatica T

Somma di Argomenti

Esercizio: Somma di Argomenti

Si scriva un programma che:

- Accetti input sotto forma di argomenti del main
- Ogni argomento deve essere un numero intero
- Calcoli e stampi a video la somma degli argomenti
- In caso un argomento non sia un intero, il programma deve stampare un messaggio di errore e terminare





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Fondamenti di Informatica T

Massimo di Argomenti

Esercizio: Massimo di Argomenti

Si scriva un programma che:

- Accetti input sotto forma di argomenti del main
- Ogni argomento deve essere un numero intero
- Calcoli e stampi a video il massimo degli argomenti
- In caso un argomento non sia un intero, il programma deve stampare un messaggio di errore e terminare
- In caso non sia passato alcun argomento, il programma deve stampare un messaggio di errore e terminare





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Fondamenti di Informatica T

Andrea Acquaviva <andrea.acquaviva@unibo.it>

Michele Lombardi <michele.lombardi2@unibo.it>

Andrea Borghesi <andrea.borghesi3@unibo.it>

Giuseppe Tagliavini <giuseppe.tagliavini@unibo.it>

Allegra De Filippo <allegra.defilippo@unibo.it>