

ESERCIZIO 1

Si scriva un programma C che chieda all'utente un numero intero N (si controlli sia minore di 6) e legga da input N dati relativi a un torneo di pallavolo, secondo il seguente formato:

- nazione1 (max. 50 caratteri)
- nazione 2 (max 50 caratteri)
- set vinti dalla squadra 1 (int) SI CONTROLLI CHE SIANO COMPRESI TRA 0 e 3
- set vinti dalla squadra 2 (int) SI CONTROLLI CHE SIANO COMPRESI TRA 0 e 3

Si controlli inoltre che i set vinti dalla squadra 1 e dalla squadra 2 non siano uguali e li memorizzi in un vettore di strutture **PARTITE** (di dimensione massima 6).

Il vettore **PARTITE** contiene strutture di tipo **partita** (usare typedef) del tipo:

- **nazione 1**
- **nazione 2**
- **set 1**
- **set 2**
- **vincitore: squadra che ha riportato un punteggio (numero di set) maggiore**

Si noti che una squadra può comparire più volte nel vettore, una per ogni partita fatta.

ESERCIZIO 1

Si scriva un programma che, tramite la procedura **find_vinte** crei e stampi un vettore **VINTE** che ha lo stesso numero di elementi del vettore **PARTITE**. Ogni elemento di **VINTE** contiene strutture di tipo **vincitore** così composte

- **nazione**
- **scarto** che conterrà la differenza tra i set della squadra vincente meno i set della squadra perdente che rappresentano la squadra che ha vinto la partita corrispondente nel vettore **PARTITE** e lo scarto.

```
void find_vinte(partita P[], int N, vincitore V[]);
```

Il secondo parametro rappresenta la dimensione dei vettori **P[]** e **V[]** (parametro di ingresso).

ESERCIZIO 2

Si scriva un programma C che:

a) legga da input i dati relativi a 4 libri, secondo il seguente formato:

- titolo del libro (stringa di max. 20 caratteri)
- cognome autore (stringa di max. 20 caratteri)
- prezzo del libro (float)
- numero di pagine (intero) si controlli che il numero di pagine sia maggiore di 100
- codice univoco (intero)

e li memorizzi in un vettore di strutture **LIBRI** (di dimensione 4). Il vettore **LIBRI** contiene strutture di tipo **libro**:

- titolo
- autore
- prezzo
- codice
- num_pagine

ESERCIZIO 2

Chieda all'utente un numero di pagine di soglia S e stampi il titolo del libro, l'autore e il prezzo per pagina di ogni libro contenuto nel vettore il cui numero di pagina è maggiore di S . La stampa deve essere fatta da una procedura

```
void stampa(struct libro L[], int dim, int S);
```

che come parametri ha il vettore, la sua dimensione e la soglia S . La procedura stampa deve usare al suo interno una funzione **prezzopag** che calcola il prezzo per pagina di un singolo libro

```
float prezzopag(int num_pag, float prezzo);
```