

PROVA PRATICA DI FONDAMENTI DI INFORMATICA A 4 LUGLIO 2002 – DOCENTE MICHELA MILANO

TEMPO A DISPOSIZIONE 1 ORA E 45 MINUTI

SCRIVERE NOME, COGNOME, MATRICOLA E TIPO DI COMPITO SU DUE DISCHETTI IN CUI COPIARE IL FILE SORGENTE E L'ESEGUIBILE

Si scriva un programma C che legga due serie di dati e li memorizzi in due vettori di strutture. Nel primo vettore TERNE (di dimensione 3) vengono memorizzate strutture (**struct terna**) del tipo:

- cognome arbitro (stringa di lunghezza massima 20, senza spazi intermedi)
- cognome primo segnalinee (stringa di lunghezza massima 20, senza spazi intermedi)
- cognome secondo segnalinee (stringa di lunghezza massima 20, senza spazi intermedi)

Nel secondo vettore PARTITE (di dimensione 3) vengono memorizzate strutture (**struct partita**) del tipo:

- codice identificativo prima squadra (stringa di lunghezza massima 20, senza spazi intermedi)
- codice identificativo seconda squadra (stringa di lunghezza massima 20, senza spazi intermedi)
- cognome arbitro (stringa di lunghezza massima 20, senza spazi intermedi)

Si scriva un programma che:

1. tramite due procedure **leggiterna** e **leggipartita** legga a terminale i dati da inserire nei due vettori;

```
void leggiterna(int n, struct terna T[]);  
void leggipartita(int n, struct partita P[]);
```

dove n è la dimensione del vettore secondo parametro;

2. si chieda all'utente il codice identificativo di una squadra **team**;
3. per ogni elemento del vettore PARTITE contenente **team** (come prima o come seconda squadra) si cerchi nel vettore TERNE gli elementi che hanno lo stesso arbitro della partita considerata e si stampino a video (cioè: si stampino a video le terne arbitrali nelle quali compare lo stesso arbitro delle partite giocate dalla squadra **team**).

Per effettuare il confronto tra stringhe e' possibile fare uso delle funzioni di libreria fornite dal C (*string.h*).

```

#include <stdio.h>
#include <string.h>

#define DIMT 3
#define DIMP 3

struct terna{
    char arbitro[20];
    char segnalinee1[20];
    char segnalinee2[20];
};

struct partita{
    int squadra1;
    int squadra2;
    char arbitro[20];
};

void leggiterna(int n, struct terna T[]);
void leggipartita(int n, struct partita P[]);

main()
{
    int i,j;
    struct terna TERNE[DIMT];
    struct partita PARTITE[DIMP];
    int team;

    leggiterna(DIMT,TERNE);
    leggipartita(DIMP,PARTITE);

    printf("\nInserire il codice di una squadra: ");
    scanf("%d",&team);

    for (i=0; i<DIMP; i++)
        if ( (PARTITE[i].squadra1 == team) || (PARTITE[i].squadra2 ==
team) )
            for (j=0; j<DIMT; j++)
                if (strcmp(PARTITE[i].arbitro,TERNE[j].arbitro) == 0)
                    printf("\nArbitro: %s, segnalinee: %s ,%s",
TERNE[j].arbitro,TERNE[j].segnalinee1,TERNE[j].segnalinee2);
}

void leggiterna(int n, struct terna T[]){
    int i;
    for(i=0;i<n;i++){
        printf("arbitro:");
        scanf("%s",T[i].arbitro);
        printf("segnalinee1:");
        scanf("%s",T[i].segnalinee1);
        printf("segnalinee2:");
        scanf("%s",T[i].segnalinee2);
    }
}

```

```
void leggipartita(int n, struct partita P[])
{
    int i;
    for(i=0;i<n;i++){
        printf("squadra1:");
        scanf("%d",&P[i].squadra1);
        printf("squadra2:");
        scanf("%d",&P[i].squadra2);
        printf("arbitro:");
        scanf("%s",P[i].arbitro);
    }
}
```