

```
/*
 * Corso di Fondamenti di Informatica
 * Esercizio:
 * strutture dati per "cubo di Rubik".
 */

/* Esempio di uso dei tipi di dato user-defined: definizione delle
 * strutture dati usate da un programma relativo al gioco del cubo di Rubik.
 * (N.B. questo programma non fa nulla di utile.) */
#define LATO_FACCIA 3
/* dimensione, in numero di caselle, di un lato del cubo di Rubik */
#define NUM_FACCE 6
/* numero di facce del cubo */

int main()
{
    typedef enum {rosso, giallo, blu, verde, bianco, nero} Casella;
    typedef Casella Faccia[LATO_FACCIA][LATO_FACCIA];
    typedef Faccia CuboDirubik[NUM_FACCE];

    typedef Casella CuboDirubik2[NUM_FACCE][LATO_FACCIA][LATO_FACCIA];
    /* una definizione alternativa altrettanto valida della precedente */

    CuboDirubik UnCubo, UnSecondoCubo;
    /* questa è la dichiarazione di una VARIABILE di tipo CuboDirubik, mentre la
     * precedente era la dichiarazione del TIPO DI DATO CuboDirubik. E'
     * FONDAMENTALE avere capito la differenza tra tipi di dato e variabili! */

    /* Come accedere alla singola casella di una faccia del cubo?
     * Ad esempio così: */

    if ( rosso == UnCubo[5][2][2] )
    /* si potrebbe anche scrivere (UnCubo[5])[2][2] o ((UnCubo[5])[2])[2] */
    {
        UnCubo[5][1][1] = giallo;
    }

    CuboDirubik2 UnAltroCubo;
}
```

```
/* Attenzione: le seguenti istruzioni sono ERRATE.
 * -----
 * UnCubo = UnSecondoCubo;
 * (Assegnamento tra due array. Va invece fatto elemento per elemento,
 * usando opportuni cicli.)
 *
 * if (UnCubo == UnSecondoCubo) {...}
 * (Non ha senso fare confronti sul valore dell'identificatore di un array,
 * che è un puntatore alla prima cella di memoria da esso occupata. Se è
 * necessario confrontare il CONTENUTO di due array, ciò va fatto elemento
 * per elemento con opportuni cicli.)
 */
return 0;
}
```