

```
/*
 * Corso di Fondamenti di Informatica
 * Esercizio: tema d'esame (leggermente modificato per rendere piÃ¹ semplici le
 * parti non richieste all'esame)
 */
/*
 * Un array di caratteri e' utilizzato per rappresentare una sequenza di 1000
 * cifre binarie. Ciascun elemento dell'array assume valore '0' qualora la cifra da
 * rappresentare sia 0, e '1' altrimenti. Si consideri il problema di codificare la
 * sequenza binaria usando la codifica Run Length Code (RLC) mediante una lista
 * sequenziale di valori interi, ciascuno dei quali rappresenta la lunghezza di una
 * sequenza di '0' consecutivi o di '1' consecutivi compresi nell'array. Si suppone
 * che la codifica RLC di un array cominci -per convenzione- sempre con il numero
 * di cifre binarie consecutive pari a 0 che si trovano all'inizio dell'array.
 * Si scriva una funzione che riceve in ingresso un
 * array binario e restituisca la sua codifica RLC. Ad esempio l'array
 * 00000011100111100000111110... ha codifica RLC 7 3 2 4 6 5...
 * mentre l'array
 * 11100111100000111110... ha codifica RLC 0 3 2 4 6 5...
 */
#include <stdio.h>

#define NUM_CIFRE 1000
/* numero di cifre che compongono la sequenza binaria da convertire */
#define NON_USATO -1
/* valore attribuito agli elementi non utilizzati dell'array contenente la
 * codifica RLC */

void CalcolaRLC(char* Binario, int* RLC)
/* Binario e' un array i cui primi NUM_CIFRE caratteri contengono le cifre della
 * sequenza binaria di cui calcolare la codifica RLC, dove ciascuna cifra
 * binaria e' rappresentato con il carattere '0' o '1' a seconda del valore; RLC
 * Ã¨ un array di NUM_CIFRE+1 interi utilizzato per contenere la codifica RLC
 * dei dati in Binario, i cui elementi non utilizzati (in coda) hanno valore
 * NON_USATO */
{
    int ContBin, ContRLC;
    /* contatori usati per scorrere gli array Binario e RLC rispettivamente */

```

```
int CifreUguali;
/* numero di cifre binarie consecutive uguali tra loro trovate durante l'esame
di Binario */
ContBin = 0;

/* Gestiamo il caso particolare in cui la prima cifra della sequenza binaria
non Ã 0, il che richiede che la codifica RLC inizi con uno 0 (si veda il
testo dell'esercizio */
if ('0' != Binario[0])
{
    RLC[0] = 0;
    ContrLC = 1;
}
else
{
    ContrLC = 0;
}

/*popola la prima parte di RLC con la codifica */
while (ContBin < NUM_CIFRE)
{
    CifreUguali = 1;

    while ( (ContBin+CifreUguali < NUM_CIFRE) &&
            (Binario[ContBin+CifreUguali] == Binario[ContBin]) )
        /* il ciclo viene eseguito fino a che l'array contiene ancora cifre utili e
        queste risultano identiche a quella di indice ContBin */
        /* NOTA: l'ordine in cui sono poste le due condizioni in AND tra loro non
        e' rilevante e non puo' essere invertito. Se venisse invertito, quando
        ContBin+CifreUguali vale NUM_CIFRE verrebbe letto l'elemento di array
        Binario[ContBin+CifreUguali] = Binario[NUM_CIFRE], che NON appartiene
        agli elementi utili dell'array e puo' addirittura trovarsi oltre
        l'ultimo elemento dell'array */
        ++CifreUguali;
    }
}

RLC[ContrLC] = CifreUguali;
++ContrLC;
ContBin = ContBin + CifreUguali;
```

```
}
}
/* popola la seconda parte di RLC con valori pari a NON_USATO */
while (ContrLC < NUM_CIFRE + 1)
{
    RLC[ContrLC] = NON_USATO;
    ++ContrLC;
}
}

/*****
NOTA: il blocco main non era richiesto all'esame; e' stato introdotto per
consentire il debugging della funzione EseguiCodificaRLC.
*****/

int main()
{
    char Originale[NUM_CIFRE+1];
    /* array contenente i dati binari da convertire; ciascun bit e' rappresentato
    con il carattere '0' o '1' a seconda del valore. La dimensione e' accresciuta
    di 1 rispetto a NUM_CIFRE per tenere conto che eseguiamo la lettura con
    scanf, che inserisce un carattere '\0' in coda agli elementi letti */
    int CodicereLC[NUM_CIFRE+1];
    /* array contenente la codifica RLC; il numero di elementi e' NUM_CIFRE + 1
    perche' occorre poter gestire il caso pessimo. Questo caso e' quello in cui il
    contenuto della sequenza binaria cambia valore ad ogni bit (e dunque la sua
    codifica RLC richiede tanti interi quante sono le cifre binarie di partenza)
    e la sequenza comincia con 1 (dunque la codifica RLC richiede uno 0 aggiuntivo
    in testa) */
    int Contatore;
    /* usato per scorrere gli elementi dell'array */

    printf("\nInserisci una successione di esattamente %d cifre binarie espresse come successione di '0' e '1', poi premi il
    tasto <enter>.\n", NUM_CIFRE);
    scanf("%s", Originale);

    printf("\nla sequenza binaria da convertire e': \n'%s'", Originale);

    CalcolaRLC(Originale, CodicereLC);
}
```

```
printf("\n\nla corrispondente codifica RLC e' (i dati sono separati da punti): \n");  
for (Contatore = 0;  
    (CodiceRLC[Contatore] != NON_USATO) && (Contatore < NUM_CIFRE+1);  
    ++Contatore)  
    /* La condizione (Contatore < NUM_CIFRE) serve a tenere conto del caso limite  
    in cui la sequenza binaria cambia valore ad ogni bit e quindi tutte le  
    cifre dell'array CodiceRLC sono utilizzate, senza alcun elemento pari a  
    NON_USATO in coda ad esse */  
    {  
        printf("%d. ", CodiceRLC[Contatore]);  
    }  
    printf("\n\n");  
return 0;  
}
```