

```
/*  
 * Corso di Fondamenti di Informatica  
 * Esercizio:  
 * tema d'esame  
 */
```

```
/* Sia dato un file di testo, "Romanzo.txt", contenente il testo completo di  
 * un romanzo. Il romanzo è diviso in capitoli e l'inizio di ogni capitolo  
 * (incluso il primo) è segnalato dalla presenza del carattere '@'.  
 * Il contenuto di Romanzo.txt va copiato in un secondo file di testo,  
 * RomanzoFormatto.txt, aggiungendo all'inizio di ciascun capitolo tre  
 * righe vuote, poi la frase "Capitolo N" (dove N è il numero del capitolo  
 * corrente) ed infine un'altra riga vuota.  
 * Successivamente in RomanzoFormatto va inserito il testo del capitolo,  
 * modificandone i caratteri della sola prima parola nel modo seguente:  
 * - se il carattere e' una lettera maiuscola, va trasformata in minuscola;  
 * - se il carattere e' una lettera minuscola, va trasformata in maiuscola;  
 * - se il carattere non e' una lettera esso va lasciato non modificato.  
 * E' consentito l'uso della libreria ctype.h.  
 * Il programma deve stampare un messaggio di errore ogniqualvolta si  
 * verifica un di errore di apertura o di chiusura file.  
 */
```

```
/* NOTA: l'operazione di trasformazione di lettere da maiuscole a minuscole o  
 * viceversa e' abbastanza frequente, anche nei temi d'esame. A volte e'  
 * consentito, per svolgerla, l'uso delle apposite funzioni  
 * int toupper(int lettera)  
 * (che restituisce la versione maiuscola di una lettera minuscola) e  
 * int tolower(int lettera)  
 * (che restituisce la versione minuscola di una lettera maiuscola).  
 * Entrambe le funzioni si limitano a restituire in uscita il carattere  
 * in ingresso, non modificato, quando esso non appartiene alla categoria  
 * da trasformare (lettere minuscole per toupper, maiuscole per tolower).  
 * Per utilizzare queste funzioni includere la libreria C ctype.h con  
 * #include <ctype.h>  
 */
```

```
*
*   Notare che le due funzioni sopra citate rappresentano i caratteri, sia
*   in ingresso che in uscita, tramite interi. Grazie alle funzionalità di
*   conversione automatica tra tipi di dato fornite dal C, è comunque
*   possibile passare direttamente un char alle funzioni e/o assegnare
*   il valore restituito direttamente ad un char senza che cio' causi
*   alcun problema.
*/
```

```
#include <stdio.h>
/* per input/output */
#include <ctype.h>
/* per conversioni tra lettere maiuscole e minuscole */

#define NOME_FILE_IN "./file-es11_02/Romanzo.txt"
/* nome del file di input, path incluso */
#define NOME_FILE_OUT "./file-es11_02/RomanzoFormattato.txt"
/* nome del file di output, path incluso */
#define TERMINATORE_CAPITOLO '@'
/* carattere che segnala il termine del singolo capitolo */

int main(void)
{
    typedef enum {false, true} boolean;

    int Carletto;
    /* valore, sotto forma di intero (come fornito da fgets e atteso da
     * fputs), dell'ultimo carattere letto dal file di ingresso */
    int NumCapitoloCorrente;
    /* il prossimo numero di capitolo da scrivere nel file di output */
    boolean PrimaParola;
    /* vale true se la parola corrente è la prima del capitolo */
    FILE* pFileIn;
    /* identifica lo stream dal quale leggere dati */
    FILE* pFileOut;
    /* identifica lo stream nel quale scrivere dati */

    /**** apertura file di input ****/
    pFileIn = fopen(NOME_FILE_IN, "r");
    if ( NULL == pFileIn )
    {
        printf("\n\nERRORE nell'apertura del file di input \"%s\"", NOME_FILE_IN);
    }
}
```

```
printf("\n\nIl programma termina.\n\n");
return (0); /* terminazione della funzione main */
}

printf("\nAperto in lettura il file %s.", NOME_FILE_IN);

/**** apertura file di output ****/
pFileOut = fopen(NOME_FILE_OUT, "w");
if ( NULL == pFileOut )
{
printf("\n\nERRORE nell'apertura del file di uscita \"%s\".",
NOME_FILE_OUT);
printf("\n\nIl programma termina.\n\n");
}

/* chiusura del file di ingresso */
if ( 0 != fclose(pFileIn) )
{
printf("ERRORE nella chiusura file di ingresso.");
}

return(0); /* terminazione della funzione main */
}

printf("\nAperto in scrittura il file %s.", NOME_FILE_OUT);

/**** lettura e scrittura dei dati, carattere per carattere ****/
NumCapitoloCorrente = 1;
Carletto = fgetc(pFileIn);

while ( (EOF != Carletto) )
{
if ( TERMINATORE_CAPITOLO == Carletto )
/* inizia un nuovo capitolo */
{
PrimaParola = true;
fprintf(pFileOut, "\n\nCapitolo %d\n\n", NumCapitoloCorrente);
++NumCapitoloCorrente;
}
else
/* copia il carattere dal file d'ingresso al file di uscita, modificandolo
```

```

* se necessario */
{
    if ( ( ' ' == Carletto) || ('\n' == Carletto) )
    {
        PrimaParola = false;
        /* quando si incontra uno spazio o un a capo, la prima parola e'
         * terminata. Notare che anche quando la prima parola e' seguita da un
         * segno di interpunzione, il carattere successivo a tale segno e'
         * sempre uno spazio o un a capo: dunque la rilevazione del termine
         * della prima parola opera correttamente anche in questo caso. */
    }

    if (true == PrimaParola)
        /* se Carletto e' una lettera, va modificato */
    {
        if (('a' <= Carletto) && ('z' >= Carletto))
            /* se e' una lettera minuscola, va resa maiuscola */
        {
            Carletto = toupper(Carletto);
        }
        else
            /* e' una lettera maiuscola, da rendere minuscola; oppure e' un altro
             * simbolo, che la funzione tolower lascerà invariato */
        {
            Carletto = tolower(Carletto);
        }
    }

    fputc(Carletto, pFileOut);
    /* aggiunge il carattere al file di output corrente */
}

Carletto = fgetc(pFileIn);
}

/**** chiusura dei file in uso ****/
if ( 0 != fclose(pFileIn) )
{
    printf("ERRORE nella chiusura file di ingresso.");
}
}
```

```
if ( 0 != fclose(pfFileOut) )  
{  
    printf("ERRORE nella chiusura file di uscita.");  
}  
printf("\n\n");  
return(0);  
}
```

www.u