

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

/*
Testo dell'esercizio:
Scrivere un programma C che legga da un file una sequenza di non più di 500 caratteri alfabetici minuscoli, senza spazi,
ordinati alfabeticamente. I caratteri possono eventualmente essere ripetuti. Il programma deve scrivere su un file un
insieme di righe, ciascuna contenente un carattere e il numero di volte in cui compare nella sequenza. I caratteri devono
essere riportati in ordine alfabetico inverso. Ad esempio se la sequenza letta Ã
aaacccfffrfrttuuuzz
il programma scriverÃ
z2 \n u3 \n t1 \n r2 \n f4 \n c2 \n a3
*/

#include <stdio.h>

#define NOME_FILE_IN ".\\file-es11_04\\input.txt"
/* nome del file di input, path incluso */
#define NOME_FILE_OUT ".\\file-es11_04\\output.txt"
/* nome del file di output, path incluso */

int main(void)
{
    int NumCarletti ['z'-'a'+1];
    /* l'elemento di indice K-'a' (dove K Ã
 un generico dato di tipo char)
    * dell'array contiene il numero di occorrenze del carattere K nel file di
    * input. In altri termini, l'elemento di indice J dell'array contiene il
    * numero di occorrenze del (J+1)-esimo carattere dell'alfabeto */

    int Carletto;
    /* valore, sotto forma di intero (come fornito da fgetc e atteso da
    * fputc), dell'ultimo carattere letto dal file di ingresso */
    FILE *pFileIn;
    /* identifica lo stream dal quale leggere dati */
    FILE *pFileOut;
    /* identifica lo stream nel quale scrivere dati */
    char Cursore;
    /* usato per scorrere NumCarletti */
}
```

```
/**** apertura file di output ****/
pfileOut = fopen(NOME_FILE_OUT, "w");
if ( NULL == pfileOut )
{
    printf("\n\nERRORE nell'apertura del file di output \"%s\".",
        NOME_FILE_OUT);
    printf("\n\nIl programma termina.\n\n");
    return(0); /* terminazione della funzione main */
}

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

/**** 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 senza eseguire alcuna operazione.\n\n");
}

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

return (0);
}

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

/**** inizializzazione di NumCarletti ****/
for (Cursore = 'a'; Cursore <= 'z'; ++Cursore)
{
    NumCarletti[Cursore-'a'] = 0;
}

/* Nota. L'indice di un array Ã un intero. Tuttavia, sfruttando il meccanismo
* di conversione automatica tra il tipo char e il tipo int che il C possiede,
* e' possibile usare direttamente il carattere Cursore per
```

```
* identificare un elemento dell'array. Dato che Cursor e ' un char, e di
* conseguenza un numero intero senza segno rappresentato da un singolo byte,
* i valori assunti dall'intero in cui viene automaticamente convertito
* Cursor sono compresi tra 0 e 255. */
```

```
**** Lettura dei dati e conteggio dei caratteri ****/
Carletto = fgetc(pFileIn);
```

```
while ( ('a' <= Carletto) && (Carletto <= 'z') )
/* questa condizione esclude il caso in cui si e' raggiunta la fine del file
* (quando fgetc restituisce EOF), ed elimina eventuali caratteri CR (Carriage
* Return) e/o LF (Line Feed, corrispondente a '\n') eventualmente
* presenti al termine del file di testo, in coda ai caratteri utili.
* Molti editor di testo inseriscono tali caratteri anche se non richiesto
* dall'utente: e' possibile verificarlo decommentando la linea di codice
* situata immediatamente dopo il ciclo. Ad esempio Gedit inserisce sempre
* un carattere '\n' in coda al file.*/
```

```
{
++NumCarletti[Carletto-'a'];
Carletto = fgetc(pFileIn);
}
```

```
/*
printf("\n###DEBUG: l'ultimo carattere letto da file ha codice %d. ",
Carletto);
printf("I codici di 'a', 'z' e '\\n' sono rispettivamente %d, %d e %d.",
'a', 'z', '\n');
*/
```

```
/* codice usato per debugging, onde verificare il contenuto dell'array:
for (Cursor = 'a'; Cursor <= 'z'; ++Cursor)
```

```
{
printf("\n###DEBUG: NumCarletti['%c'] = %d", Cursor,
NumCarletti[Cursor-'a']);
}
```

```
*/
**** scrittura nel file di output ****/
for (Cursor = 'z'; Cursor >= 'a'; --Cursor)
{
```

```
if (NumCarLetti[ cursore-'a'] > 0)
{
    fprintf(pfFileOut, "%c%d\n", cursore, NumCarLetti[ cursore-'a']);
}
}

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

if ( 0 != fclose(pfFileOut) )
{
    fprintf("ERRORE nella chiusura file di uscita.");
}

printf("\n\n");

return(0);
}
```