

```
/*  
 * Corso di Fondamenti di Informatica  
 * Esercizio:  
 * tema d'esame (2017-02-23)  
 */
```

```
/*  
 Scrivere un programma per il calcolo di statistiche sui caratteri presenti in un file di testo il cui nome Ã¨ specificato  
 come primo parametro sulla riga di comando. Il file contiene un testo, suddiviso in una o piÃ¹ linee di lunghezza pari al  
 piÃ¹ a 100 caratteri cadauna. Ogni linea e' terminata da un carattere 'a capo'.  
 Nel calcolo delle statistiche, il programma considera i caratteri letti dal file AD ECCEZIONE DEGLI SPAZI E DEGLI 'A CAPO'.  
 Esso fornisce in uscita:
```

- il numero di linee presenti nel file;
- il numero totale di caratteri;
- il numero massimo e il numero medio di caratteri per linea;
- il testo completo della linea piÃ¹ lunga presente nel file.

```
E' consentito l'uso della libreria string.h
```

```
Ad esempio, dato il file contenente il seguente testo:
```

```
Prova di contenuto di file  
per avere un esempio.
```

```
Non e' importante quello che c'e' scritto  
presente.
```

```
il programma visualizza:
```

```
Linee: 4  
Car: 84  
Max, Med: 35, 21  
Linea max: Non e' importante quello che c'e' scritto  
*/
```

```
#include <stdio.h>  
#include <string.h>  
#define LUNG_STR 101
```

```
/* #define DEBUG */  
/* commentare via questa direttiva una volta finito il debugging */
```

```
int main(int argc, char *argv[])  
{  
    FILE* fileInput;
```

```
/* stream da cui prelevare il testo */
char LineaCorrente[LUNG_STR];
/* ultima linea di testo letta dal file */
char LineaMax[LUNG_STR];
/* linea di testo di lunghezza massima letta dal file */
int LungCorrente, LungMax;
/* lunghezza (in caratteri) della linea di testo corrente e della
   linea di lunghezza massima */
int NumCaratteri;
/* numero totale di caratteri letti dal file, esclusi ' ' e '\n' */
int NumLinee;
/* numero di linee di testo lette dal file */
int Cursore;
/* scorre una stringa durante il suo esame */

if ( 1 == argc )
{
    printf("\n\nProgramma lanciato non correttamente!\nE' richiesto un argomento che indichi il path del file di
    ingresso.\nIl programma termina senza aver eseguito alcuna operazione.\n\n");
    return 1;
}

#ifdef DEBUG
    printf("\nDEBUG: nome file di input = '%s', argv[1]);
#endif
FileInput = fopen(argv[1], "r");

if ( NULL == FileInput )
{
    printf("\nErrore di apertura file di ingresso!");
}
else
{
    printf("\nFile di ingresso aperto correttamente.\n");
    NumCaratteri = 0;
    LungMax = 0;
    NumLinee = 0;
}
```

```
while (NULL != fgets(LineaCorrente, LUNG_STR, FileInput))
{
    #ifdef DEBUG
        printf("\n\nDEBUG: letta linea di testo '%s'", LineaCorrente);
    #endif

    NumLinee = NumLinee + 1;
    LungCorrente = strlen(LineaCorrente);
    /* si noti che questo numero (per ora) include gli spazi e i '\n' */

    for (Cursore = 0; '\0' != LineaCorrente[Cursore]; ++Cursore)
        /* sottrae dalla lunghezza della linea di testo il contributo di spazi e
        * '\n' */
        {
            if ( (' ' == LineaCorrente[Cursore]) ||
                ('\n' == LineaCorrente[Cursore]) )
            {
                --LungCorrente;
                #ifdef DEBUG
                    printf("\n\nDEBUG: eliminato dal calcolo carattere con codice %d",
                        LineaCorrente[Cursore]);
                #endif
            }
        }

        #ifdef DEBUG
            printf("\n\nDEBUG: la linea comprende %d caratteri utili", LungCorrente);
        #endif

        NumCaratteri = NumCaratteri + LungCorrente;

        if (LungMax < LungCorrente)
        {
            strcpy(LineaMax, LineaCorrente);
            LungMax = LungCorrente;
        }
    }

    printf("\nLinee:      %d", NumLinee);
    printf("\nCar:      %d", NumCaratteri);
    printf("\nMax, Med:  %d, %d", LungMax, NumCaratteri/NumLinee);
}
```

```
printf("\nLinea max: %s", LineaMax);  
if ( 0 != fclose(FILEInput) )  
{  
    printf("\nErrore di chiusura file di ingressi!");  
}  
return 0;  
}
```