Laboratorio di informatica

Ingegneria meccanica

Esercitazione 11 – 12 dicembre 2007

Operazioni base sui file

- **Apertura: fopen**
  - **Parametri: nome del file, modalità di apertura del file**
  - **Esempio:**

```
FILE *ptrfile
ptrfile = fopen("A:\\dati.txt", "r");
if (ptrfile == NULL)  printf ("errore apertura
  file\n");
else {...}
```

- **Chiusura: fclose**
  - **Parametro: puntatore al file**
  - **Esempio: fclose(ptrfile);**
- **Riposizionamento all'inizio del file: rewind**
  - **Parametro: puntatore al file**
  - **Esempio: rewind(ptrfile);**

Operazioni sui file: lettura

- **Lettura di un carattere: fgetc**
  - **Parametro: puntatore al file**
  - **Esempio: carattere = fgetc(ptrfile);**
  - **fgetc(stdin) legge un carattere da standard input**
- **Lettura di una stringa: fgets**
  - **Parametri: stringa, numero massimo di caratteri della stringa, puntatore al file**
  - **Esempio: fgets(stringa, 80, ptrfile);**
- **Lettura con formato: fscanf**
  - **come scanf ma con un parametro in più, il puntatore al file**
  - **Esempio: fscanf(ptrfile , "%c" , &ch);**

Operazioni sui file: scrittura

- **Scrittura di un carattere: fputc**
  - **Parametri: carattere, puntatore al file**
  - **Esempio: fputc(car, ptrfile);**
  - **fputc('a', stdout) equivale a putchar('a')**
- **Scrittura di una stringa: fputs**
  - **Parametri: stringa, puntatore al file**
  - **Esempio: fputs("Prova", ptrfile);**
- **Scrittura con formato: fprintf**
  - **come printf ma con un parametro in più, il puntatore al file**
  - **Esempio: printf(ptrfile , "Valore = %d" , numero);**

Operazioni sui file: test di fine file

- Confrontare il carattere letto con la costante EOF (definita in stdio.h, vale –1)

  Esempi:

```
while ( (ch = fgetc(ptrfile)) != EOF)
while ( fscanf (ptrfile, "%c", &ch) != EOF)
```

- Usare la funzione: feof
  - Parametri: puntatore al file
  - Restituisce 0 se non è stata raggiunta la fine del file, altrimenti un valore diverso da 0

  Esempio:

```
while ( !feof(ptrfile))
while (feof(ptrfile)==0)
```

Esempio: conteggio caratteri in un file

```
#include <stdio.h>
int main( ) {
  FILE *fPtr ;
  char ch ;
  long int cont = 0 ;
  if (( fPtr = fopen( "provar" , "r" )) != NULL ) {
        while( feof(fPtr)==0 ) {
           fscanf( fPtr , "%c" , &ch );
           cont++ ;
        }
        printf( "cont = %ld\n" , cont ) ;
        /* cont = numero caratteri in "provar" */
        fclose( fPtr ) ;
  }
  else printf( "errore apertura file" ) ;
  return 0;
}
```


Esercizio 1 (1/2)

Scrivere un programma che legge da stdin
- il nome di un file (una stringa)
- un intero d
- d numeri reali

e li memorizza nel file un numero per riga.

Estendere il programma precedente in modo tale che:

- legga dal file riempito precedentemente i numeri reali e li memorizzi in un array
- stampi su standard output il numero di valori effettivamente inseriti nell'array e i valori stessi

Attenzione: la dimensione massima dell'array deve essere dichiarata come costante MAX_D con una direttiva per il preprocessore, e il programma deve interrompere la lettura da file se il numero di valori letti eccede MAX_D o se viene raggiunta la fine del file.

Esercizio 1 (2/2)

Modificare il programma precedente in modo tale che la lettura da file e memorizzazione nell'array sia realizzata da una funzione che riceve come argomenti:

- una stringa (array di char), per il nome del file
- un array di reali
- un intero, per la dimensione dell'array

e restituisce il numero di elementi letti.

Esercizio 2 (1/2)

Progettare una funzione che, ricevuti
- una stringa s (array di char terminato con '\0')
- due char c1 e c2

sostituisca ogni occorrenza di c1 in s con c2 e restituisca il numero totale n di sostituzioni eseguite in s

Inserire la funzione in un programma contenente quanto necessario per verificare se il comportamento della funzione e' corretto.
Il programma di prova deve scrivere su file la sequenza di caratteri che costituiscono la stringa prima e dopo la sostituzione su due linee distinte, e sulla terza linea il numero di sostituzione effettuate.

Esercizio 2 (2/2)

Verificare la funzione almeno con i seguenti dati di test
s = ['a', 'b', 'a', 'r', 'o', 'b', 'b', '\0'],
c1 = 'b', c2 = 'c'
s = ['a', 'b', 'a', 'r', 'o', 'b', 'b', '\0'],
c1 = 'x', c2 = 'c'
producendo due file distinti per i risultati nei due casi

NOTA: fare in modo che il programma legga da standard input i dati necessari all'esecuzione (nome del file di output, stringa iniziale e i due caratteri)
In questo modo sarà più agevole effettuare il test del programma

Esercizio 3

- Scrivere un programma che, a partire da due file esistenti, crei un nuovo file il cui contenuto è la concatenzione del contenuto dei primi due. I nomi dei tre file devono essere letti da standard input
- Esempio:

Primo file di input

```
Primo file di
input!
```

Secondo file di input

```
Secondo file di
input!
```

File di output

```
Primo file di
input!
Secondo file di
input!
```

Esercizio 4

- Scrivere un programma che conti quante volte ciascuna lettera dell'alfabeto compare all'interno di un file e scriva il risultato su un altro file. Il nome del file deve essere letto da standard input. Il programma non deve fare distinzioni tra lettere minuscole o maiuscole; inoltre i caratteri che non sono lettere non devono essere considerati.

File di input

```
Questo e' un
file di prova!
```

File di output

```
a = 1
b = 0
c = 0
d = 1
e = 3
f = 1
…
```