

Capitolo 10 - File di testo

1. Scrittura interi casuali

Scrivi un programma che genera un file di testo contenente N numeri interi casuali compresi tra A e B , separati dal carattere "spazio". I valori N , A , B ed il *nome* del file sono forniti dall'utente. Nel caso in cui non sia possibile creare il file, il programma deve mostrare un opportuno messaggio di errore.

2. Scrittura float casuali in modalità "aggiunta"

Scrivi un programma che genera un file di testo contenente N numeri **float** casuali compresi tra 0 e 1, separati dal carattere "a capo". Il valore N ed il *nome* del file sono forniti dall'utente. Se il file esiste già, il programma deve aggiungere i nuovi numeri alla fine del file esistente. Esempio: se $N=3$, il contenuto del file di output può essere:

```
0.8039451
0.5739421
0.178321
```

3. Scrittura float casuali (bis)

Modifica il programma precedente in modo da generare numeri **float** casuali nell'intervallo tra A e B (A e B sono due valori float forniti dall'utente). Esempio: input $N=3$, $A=10.5$ $B=11.5$, output :

```
10.678531
11.003481
10.456199
```

4. Scrittura password casuali

Generare un file di password casuali, in formato CSV, con la seguente struttura:

```
utente;password
u0001;USKEL
u0002;SLMDO
u0003;QODGE
```

Il programma richiede all'utente:

- Nome del file da generare. Se il file esiste già occorre avvertire l'utente.
- Numero di password da inserire nel file.

- Lunghezza della password. Tutte le password avranno la stessa lunghezza e la lunghezza minima è 4.

Ogni password è composta esclusivamente da lettere maiuscole. Il file di esempio è stato generato con: `numero_password = 3` , `lunghezza_password = 5`.

5. Scrittura password casuali (bis)

Modifica l'esercizio precedente in modo che ogni password generata contenga sempre:

- Almeno una lettera maiuscola
 - Almeno una lettera minuscola
 - Almeno una cifra
-

6. Lettura dei primi 10 caratteri di un file

Scrivi un programma che stampa i primi 10 caratteri di un file specificato dall'utente. Se il file non esiste, mostrare un opportuno messaggio di errore. Se il file ha meno di 10 caratteri, mostrare il messaggio "FINE" dopo aver letto l'ultimo carattere disponibile.

7. Lettura di un file di lunghezza indeterminata

Scrivi un programma che legge un file di testo contenente numeri interi, separati da spazi. Il programma deve stampare tutti i numeri presenti nel file e infine mostrare la loro somma. Nota: all'utente deve essere richiesto solo il nome del file, e non il numero di valori contenuti!

8. Statistiche

Il file `input.txt` contiene un elenco di numeri reali compresi tra 0 e 1. Scrivi un programma che analizza il file `input.txt` (che dovrà trovarsi nella cartella corrente del programma) e genera il file `output.txt` contenente le seguenti informazioni:

- Numero massimo
- Numero minimo
- Percentuale di valori minori di 0.5
- Percentuale di valori maggiori o uguali a 0.5

Esempio

`input.txt`

```
0.9
0.001
```

```
0.4
0.25
```

output.txt

```
Massimo = 0.9
Minimo = 0.001
PercentualeMinori = 75%
PercentualiMaggiori = 25%
```

Spiegazione: il file di input contiene quattro valori, e i valori minori di 0.5 sono tre, quindi la percentuale è $(3/4)*100$.

9. Lettura file CSV

Il file `input.csv` contiene due numeri reali su ogni riga, separati dal carattere ";" . Esempio:

```
2.5;0.5
1.5;3.5
3.2;1.0
```

Scrivi un programma che legge il file e mostra la somma della prima e della seconda colonna. Nell'esempio, la prima somma è 7.2 e la seconda è 5.

10. Conteggio parole

Scrivi un programma che, dato un file di testo specificato dall'utente, calcola il numero di parole contenute nel file e la lunghezza media di una parola.

11. Frequenza dei caratteri

Dato un file di testo specificato dall'utente, mostrare il numero di occorrenze e la frequenza relativa di ogni Carattere ASCII. L'elenco deve essere ordinato per frequenza *decrescente*. I caratteri con frequenza nulla non devono comparire nell'elenco. Esempio:

```
Inserisci il nome del file: input.txt
OUTPUT
Carattere B: occorrenze=10 frequenza=50%
Carattere C: occorrenze=6 frequenza=30%
Carattere A: occorrenze=4 frequenza=20%
```

Nota: la frequenza assoluta è il numero di occorrenze (quante volte compare quel carattere), mentre la frequenza relativa è data da $(\text{numero_occorrenze}/\text{numero_totale_caratteri})*100$.

12. File persone (CON STRUTTURE)

Il file `input.txt` memorizza su ogni riga il nome, il cognome e l'età di una persona. Nome e cognome sono sempre presenti, mentre l'età può mancare. Il file contiene al massimo 1000 persone. Esempio:

input.txt

```
Luca Verdi 50
Federico Bianchi 80
Mario Rossi 25
Elisa Rossi
Anna Bianchi 28
```

Scrivi un programma che analizza il file di input e crea il file `output.txt` con le seguenti caratteristiche:

- a. Le persone sono ordinate per età crescente. Se l'età non è presente è da considerare come 0.
- b. Per ogni persona occorre riportare prima il cognome e poi il nome

output.txt

```
Rossi Elisa 0
Rossi Mario 25
Bianchi Anna 28
Verdi Luca 50
Bianchi Federico 80
```

Suggerimento: definire la struttura persona e caricare tutti i dati in un array di strutture.
