



Esercitazione di laboratorio n. 1

Scadenza per upload su Portale: martedì 14 ottobre ore 23:59

Esercizio n. 1: manipolazione di una matrice

Classificazione: tipo: *ricerca*; ambito: *numerico*; riferimento al testo: cap. 4.2.2 *dati vettoriali (matrici), problemi iterativi*

Competenze C: *argomenti al main, uso di funzioni, passaggio di parametri a funzioni, lettura/scrittura di file, manipolazioni di matrici statiche.*

Un file di testo contiene una matrice di interi con il seguente formato:

- la prima riga del file specifica le dimensioni reali della matrice (numero di righe *nr* e numero di colonne *nc*). Si assuma che entrambi i valori siano comunque al più pari a 20
- ciascuna delle *nr* righe successive contiene gli *nc* valori corrispondenti a una riga della matrice, separati da uno o più spazi.

Si scriva un programma C che:

- legga tale matrice dal file di ingresso
- generi una nuova matrice, in cui il valore di ciascun elemento è dato dalla media aritmetica delle (al più) 8 caselle adiacenti all'elemento corrispondente della matrice di ingresso
- scriva la matrice così ottenuta su un file di uscita, con lo stesso formato del file di ingresso.

Il nome dei due file sia passato al programma sulla riga di comando.

Esempio

Sia il contenuto del file di ingresso il seguente:

```
3 4
1 2 3 4
5 6 7 8
9 10 11 12
```

Il file di uscita scritto dal programma deve dunque avere il contenuto seguente:

```
3 4
4.3 4.4 5.4 6.0
5.6 6.0 7.0 7.4
7.0 7.6 8.6 8.7
```

in cui il primo valore è ottenuto con il calcolo $(2+5+6)/3$, il secondo come $(1+3+5+6+7)/5$, etc.

Esercizio n. 2: decompressione di un testo

Classificazione: tipo: *selezione*; ambito: *elaborazione testi*; riferimento al testo: cap. 4.4.3 *dati vettoriali, elaborazione di testo mediante stringhe, codifica/decodifica*

Competenze C: *uso di funzioni, passaggio di parametri a funzioni, lettura/scrittura di file, I/O di stringhe.*

Un'applicazione per la compressione di testi opera su un file di testo, individuando anzitutto un insieme di parole del testo (tipicamente lunghe e ripetute) che verranno poi elaborate come segue:

- a ciascuna delle parole dell'insieme così costruito viene associato un intero univoco, con valore sempre compreso nell'intervallo $[0 - 99]$. L'applicazione riporta in un secondo file di testo ogni associazione intero/parola (una per riga)



- nel file di testo originale, infine, ogni parola che appartiene all'insieme viene sostituita dall'intero ad essa associato (preceduto dal carattere '\$').

Si scriva un programma C che, dati due file (il file di testo trasformato `codificato.txt` e il corrispondente file delle associazioni `dizionario.txt`), ricostruisca il file di testo originale. Si può assumere che ogni riga del file "compresso" `codificato.txt` sia lunga al più 80 caratteri.

Esempio

Sia il contenuto del file di testo `codificato.txt` il seguente:

Questo e' solo un \$2 di quello che il \$5 deve ottenere.

L'\$2 e' corto, ma si potrebbe facilmente produrre un \$2 piu' lungo.

Il \$5 deve \$19 in grado di generare esattamente questo file.

Si noti che il \$5 deve anche rispettare il formato del file.

Per \$2, nel file potrebbero \$19 presenti piu' caratteri di spaziatura contigui, come qui, ed essi devono \$19 riportati senza modifiche!

e il contenuto del file `dizionario.txt` il seguente:

5 programma

2 esempio

19 essere

Il programma deve quindi scrivere nel file `decodificato.txt` il seguente contenuto:

Questo è solo un esempio di quello che il programma deve ottenere.

L'esempio è corto, ma si potrebbe facilmente produrre un esempio più lungo.

Il programma deve essere in grado di generare esattamente questo file.

Si noti che il programma deve anche rispettare il formato del file.

Per esempio, nel file potrebbero essere presenti più caratteri di spaziatura contigui, come qui, ed essi devono essere riportati senza modifiche!

SUGGERIMENTO: si usi un vettore di stringhe (realizzato come matrice di caratteri) per memorizzare l'associazione tra un intero (indice del vettore, ovvero indice di riga della matrice) e la parola (elemento del vettore, corrispondente a una riga della matrice) ad esso collegata. Il file di ingresso, inoltre, sia letto per righe (o, in alternativa, per caratteri).

Esercizio n. 3: stringhe periodiche

Classificazione: tipo: *selezione*; ambito: *elaborazione testi*; riferimento al testo: cap. 3.3.2 *dati scalari, elaborazione di stringhe carattere per carattere*

Competenze C: *I/O di stringhe, funzioni sulle stringhe.*

Una stringa di caratteri si dice periodica con periodo minimo k quando è composta dalla concatenazione di almeno 2 sottostringhe identiche di lunghezza k . Ad esempio, la stringa "abcabcabcabc" ha periodo minimo 3 in quanto composta da 4 ripetizioni della stringa "abc", "zzzzzz" ha periodo minimo 1, "k21k21" ha periodo minimo 3, "vivaviva" ha periodo minimo 4. La stringa "a" non è periodica poiché non presenta ripetizioni essendo composta da un solo carattere, così come non lo è la stringa "abcabcabca", in quanto l'ultimo periodo non risulta concluso.

Si scriva un programma in grado di:



- leggere da tastiera un numero indefinito di stringhe, ciascuna composta da soli caratteri alfanumerici e di lunghezza massima pari a 30. La lettura termina quando viene introdotta la parola “stop”
- indicare, per ciascuna stringa acquisita, se essa è “periodica” oppure no e, nel caso lo sia, il suo periodo minimo.

Esempio

Il seguente è un esempio di esecuzione del programma (il testo sottolineato è inserito dall'utente):

Stringa? pcpcpcpc

La stringa pcpcpcpc' periodica (periodo 2).

Stringa? A

La stringa A non e' periodica.

Stringa? 901801901801

La stringa 901801901801 e' periodica (periodo 6).

Stringa? mdm

La stringa mdm non e' periodica.

Stringa? stop

Programma terminato.