



Esercitazione di laboratorio n. 2

(Caricamento sul portale entro le 23.59 del 03/11/2017 dell'es. 2 e di uno tra gli es. 1 e 3)

Esercizio n. 1: Calcolo di media mobile su vettore

Competenze: lettura/scrittura di file, manipolazioni di vettori statici.

Categoria: problemi numerici iterativi (Dal problema al programma: 4.1)

Un main dichiara e legge da tastiera un vettore V di intero di al più 30 elementi. Dopo aver letto da tastiera un intero k il cui significato sarà spiegato oltre, il main richiama una funzione `mediaMobile` per calcolare la media mobile di finestra k dei contenuti del vettore, memorizzata in un vettore M di float di al più 30 valori, anch'esso dichiarato nel main.

La finestra i -esima di ampiezza k del vettore è composta dalla cella in posizione i e dalle $k-1$ celle seguenti. In caso non siano disponibili k celle, allora non si proceda al calcolo della media per quella finestra.

Il prototipo della funzione sia:

```
int mediaMobile(int V[], int v, int k, float M[]);
```

La funzione riceve in ingresso il vettore V , la sua lunghezza v , l'ampiezza dell'intervallo da considerare k e il vettore M in cui memorizzare le medie. La funzione ritorna il numero di medie calcolate e memorizzare in M .

Esempio

Sia V

1	1	2	3	6	0	1	1
---	---	---	---	---	---	---	---

con $k = 3$,
allora M sarà

1.333	2.000	3.667	3.000	2.333	0.667
-------	-------	-------	-------	-------	-------

Esercizio n. 2: Occorrenze di parole

Competenze: lettura/scrittura di file, manipolazioni di testi, ricerca in tabelle di nomi/stringhe

Categoria: problemi di elaborazione testi mediante stringhe (Dal problema al programma: 4.4.3)

Si scriva un programma in grado di localizzare, all'interno di un generico testo, le occorrenze di ogni parola iniziante con un prefisso dato. Per parola si intende una sequenza di caratteri alfanumerici contigui (identificati dalla funzione `isalnum`), separata da spazi o segni di punteggiatura



(identificati dalle funzioni `isspace` e `ispunct`). Per prefisso di lunghezza k di una parola si intendono i primi k caratteri da sinistra della stessa.

Input:

- file `sequenze.txt`: sulla prima riga contiene il numero totale di sequenze, al più 20, sulle righe successive, una per riga, le sequenze da ricercare. La lunghezza delle singole sequenze è limitata a massimo 5 caratteri. Non si faccia distinzione tra maiuscole e minuscole;
- file `testo.txt`: contiene il testo. Il numero di righe non è noto a priori. Si assuma che la lunghezza di ogni riga sia al più pari a 200 caratteri. Si assuma inoltre che nessuna parola del testo sia più lunga di 25 caratteri.

Funzionalità: visualizzare, per ognuno dei prefissi, quali parole hanno quel prefisso e dove si trovano nel file in termini di conteggio delle parole.

Esempio

Sia il file con l'elenco delle sequenze di inizio da ricercare il seguente:

```
4
no
Al
per
s
```

Si consideri inoltre il seguente testo come riferimento:

Non sempre si capisce un esercizio alla prima lettura, ma prestando attenzione al testo e all'esempio non dovrebbe essere impossibile scrivere codice funzionante nonostante i dubbi iniziali. Se ancora non si capisce, allora basta chiedere all'esercitatore di turno.

A titolo di esempio, la sequenza `no` è prefisso di `Non` (parola in posizione 1 nel testo), non (posizione 18), nonostante (posizione 25), non (posizione 31) ma non di turno.

Il programma deve produrre un output simile al seguente:

```
no
Non 1
non 18
nonostante 25
non 31
```

```
Al
alla 7
al 13
all 16
allora 34
all 37
```

```
s
sempre 2
si 3
scrivere 22
Se 29
si 32
```



Esercizio n.3: Somme su matrici

Competenze: lettura/scrittura di file, manipolazioni di matrici statiche.

Categoria: problemi numerici iterativi (Dal problema al programma: 4.2.2)

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 n_r e numero di colonne n_c). Si assuma che entrambi i valori siano comunque al più pari a 20
- ciascuna delle n_r righe successive contiene gli n_c 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 e la visualizzi
- chieda ripetutamente all'utente un indice di riga e di colonna e calcoli la somma degli elementi nelle quattro sottomatrici individuate considerando gli indici introdotti dall'utente come frontiere, trascurando gli elementi appartenenti alle frontiere stesse. Si osservi che le sottomatrici potrebbero essere vuote
- termini quando l'utente inserisce coordinate errate, ad esempio -1 , -1

Esempio:

Si consideri la seguente matrice 3×4 , e si assuma che l'utente abbia specificato $r = 1$ e $c = 2$

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

Le somme associate alle quattro sottomatrici sono:

$$NO = 1 + 2 = 3$$

$$NE = 4$$

$$SO = 9 + 10 = 19$$

$$SE = 12$$