

Fondamenti di Informatica ◇ 2019-20

Laboratorio #3 del 14-11-2019

Operazioni da file

Scrivere un programma che legga dal file `operazioni.txt` un elenco a priori di lunghezza ignota di operazioni **tra valori interi** da eseguire, una per riga. L'operazione tra due operandi interi è presentata nel formato:

`<operando1> <operazione> <operando2>`

Tra il primo operando, l'operatore e il secondo operando c'è esattamente uno spazio, e gli operatori utilizzati sono solamente `+`, `-`, `*` o `/` (non effettuare controlli, è senz'altro così).

Il programma visualizza i risultati delle operazioni presenti nel file. Nella soluzione avvalersi di un sottoprogramma (che dovrete sviluppare) che riceve in ingresso i due operandi e l'operatore e restituisce al chiamante il risultato dell'operazione. Dopo la sequenza di risultati visualizzata, mettere un 'a-capo'.

Per esempio, se il contenuto del file `operazioni.txt` è il seguente

```
5 + 4
10 / 9
4 * 7
1520 - 5567
184 * 2
```

il programma visualizza: 9 1 28 -4047 368

Ingresso/Uscita:

input: file di testo contenente stringhe

output: una sequenza di interi

Valore mediano di un array

Il file di testo `mediana.txt` contiene una sequenza di al più 100 interi, preceduta da un intero `n` che rappresenta il numero di valori successivamente presenti nel file (non ci saranno inconsistenze, se `n` vale 12, nel file ci sono poi 12 valori interi).

Scrivere un programma che visualizzi il valore mediano `M` dei numeri presenti nel file. Il valore mediano cercato `M` è il valore che occupa la posizione centrale nella distribuzione ordinata dei valori.

Per risolvere il problema, sviluppare due sottoprogrammi: `ordina` e `mediano`, il primo per ordinare un insieme di valori, il secondo per calcolare il valore mediano.

Dopo il valore visualizzato, mettere un `'a-capo'`.

Per esempio, se il contenuto del file è il seguente:

```
10
3
5
97
45
68
32
15
20
1000
24
```

il programma visualizza 32.

Ingresso/Uscita:

input: file di testo contenente valori interi

output: un intero (seguito da un carattere `'a-capo'`)

Prodotto tra matrici

Scrivere un programma che acquisisce i dati di due matrici di valori interi da un file binario `mats.bin`, e ne effettua il prodotto, visualizzando il risultato (seguito dal carattere 'a-capo'). Il contenuto del file binario è il seguente:

```
numero di righe matrice 1
numero di colonne matrice 1
numero di colonne matrice 2
valori matrice 1
valori matrice 2
```

(Si noti che il numero di colonne della matrice 1 è pari al numero di righe della matrice 2 affinché il prodotto possa essere fatto.) Le matrici hanno senz'altro dimensione non superiore a 50x50.

Per esempio, se il file contiene (qua visualizzato non in binario):

```
3
4
2
12 54 99 5
2 45 68 33
10 22 888 4
197 -85
264 -8
487 1
-4 -8
```

il programma visualizza:

```
64813 -1393
45258 -726
440218 -170
```

Ingresso/Uscita:

input: un file binario

output: una sequenza di interi organizzati su più righe

Ruota immagine a destra

Un'immagine in bianco e nero è rappresentata in un sistema di calcolo come un array bidimensionale di valori interi tra compresi nell'intervallo $[0, 255]$, ciascuno rappresentante il valore di intensità del pixel in quella posizione. Scrivere un programma che acquisita un'immagine da un file il cui nome viene chiesto all'utente (ed è al più di 30 caratteri), la ruota di 90 gradi in senso orario. L'immagine ha una dimensione massima di 100x100 pixel.

Il formato del file contenente l'immagine è il seguente:

```
<numero righe immagine> <numero colonne immagine>  
<valori dei pixel dell'immagine>
```

Per esempio, se il file contiene:

```
4 6  
2 55 79 3 218 24  
41 5 46 233 96 9  
73 34 11 0 109 68  
35 85 17 45 20 219
```

il programma visualizza:

```
35 73 41 2  
85 34 5 55  
17 11 46 79  
45 0 233 3  
20 109 96 218  
219 68 9 24
```

Ingresso/Uscita:

input: un file di interi

output: sequenze di interi organizzati per righe

Somma dei quadrati delle differenze tra immagini

Confrontare due immagini (o parti di immagini) e capire quanto sono simili è un problema molto comune in Computer Vision. Un modo per capire quanto due immagini sono simili consiste nel calcolare la somma dei quadrati delle differenze pixel per pixel, metodo noto con il nome di Sum of Squared Differences o SSD.

Scrivere un programma che dati due file contenenti due immagini (con le stesse dimensioni di al più 50x50 pixel) calcoli l'indice SSD, ovvero:

$$SSD = \frac{1}{n} \sum_{i=0}^{n_r} \sum_{j=0}^{n_c} (I_1(i, j) - I_2(i, j))^2 \quad (1)$$

dove n è il numero di pixel di una immagine, n_r è il numero di righe, n_c il numero di colonne e $I_x(i, j)$ è il valore del pixel (i, j) nell'immagine x . Si calcoli il valore di SSD tramite un sottoprogramma opportuno che riceve in ingresso le due immagini e restituisce il valore calcolato. Il programma chiede all'utente il nome dei due file contenenti le immagini (ciascuno al più 30 caratteri).

Il formato del file contenente un immagine è:

```
<numero righe immagine> <numero colonne immagine>  
<valori dei pixel dell'immagine>
```

Per esempio, date le immagini:

```
3 4  
25 65 48 88  
100 251 64 95  
65 10 25 45
```

```
3 4  
28 65 55 94  
100 254 70 95  
77 20 30 45
```

il valore visualizzato è 9547.083008

Ingresso/Uscita:

input: due file di testo contenente interi

output: un valore reale (seguito da un carattere 'a-capo')
