

Laboratorio di programmazione

23 novembre 2007

Bilancio finanziario

Scrivete un programma per la gestione semplificata di un conto finanziario. Il programma dovrà mostrare un menu di 5 scelte: svuotare il conto, accreditare una somma, addebitare una somma, mostrare il saldo, uscire dal programma. Le scelte vanno rappresentate rispettivamente dagli interi 0, 1, 2, 3, 4. In base alla scelta dell'utente, il programma dovrà chiedere eventuali informazioni aggiuntive e quindi eseguire l'operazione richiesta. All'inizio dell'esecuzione, potete assumere che il conto sia vuoto.

Esempio di funzionamento

```
Operazioni: 0 = svuota, 1 = accredita, 2 = addebita, 3 = stampa saldo, 4 = esci
Scegli operazione: 1
Somma da accreditare: 1052.30
Scegli operazione: 3
Saldo: 1052.30
Scegli operazione: 2
Somma da addebitare: 333.10
Scegli operazione: 2
Somma da addebitare: 1000.00
Scegli operazione: 3
Saldo: -280.80
Scegli operazione: 4
```

Espressioni ben parentesizzate

Una successione di caratteri è un'espressione ben parentesizzata se, per ogni prefisso della successione stessa (cioè, per ogni possibile segmento iniziale della successione), il numero di parentesi aperte "(" è maggiore o uguale al numero di parentesi chiuse ")", e se, complessivamente, il numero di parentesi aperte è uguale al numero di parentesi chiuse.

Scrivete un programma che legga (mediante la funzione `getchar()`) una sequenza di caratteri terminata da `.` e determini se essa è un'espressione ben parentesizzata. In caso negativo, il programma dovrà stampare in quale posizione ha identificato un errore, e il tipo di errore.

Esempio di funzionamento

```
Stringa: ((1)abb(3(2a)4(b))5).
La stringa è un'espressione ben parentesizzata
```

```
Stringa: ((1)abb(3(2a)))4(b)5.
La stringa non è un'espressione ben parentesizzata:
Carattere 19: mancano parentesi chiuse alla fine!
```

```
Stringa: ((1)abb(3))(2))a4(b))5).
La stringa non è un'espressione ben parentesizzata:
Carattere 15: troppe parentesi chiuse!
```

Fattoriale di un numero intero

Il fattoriale di un intero n , denotato col simbolo $n!$, è definito come segue:

$$0! = 1 \quad \text{e} \quad n! = n \cdot (n-1) \cdot (n-2) \cdots 1 \text{ per ogni } n \geq 1.$$

Ad esempio, $4! = 4 \cdot 3 \cdot 2 \cdot 1 = 24$ e $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$.

Scrivete un programma che calcoli il fattoriale di un numero dato in input.

Suggerimento. Calcolate in maniera incrementale il prodotto dei numeri tra n e 1, memorizzando man mano il prodotto parziale in una variabile. Partite da n , quindi calcolate $n \cdot (n-1)$, poi $n \cdot (n-1) \cdot (n-2)$ e via dicendo fino ad ottenere il fattoriale cercato.

Somma di esattamente 10 numeri non nulli

Scrivete un programma che legga una serie di numeri e ne calcoli la somma; il programma deve terminare dopo aver letto esattamente 10 numeri diversi da 0.

Medie

Data una sequenza di numeri reali x_1, x_2, \dots, x_n , sono dette rispettivamente

1. *media geometrica* il valore $\sqrt[n]{x_1 x_2 \dots x_n} = \exp(\log(x_1 x_2 \dots x_n)/n)$,

2. *media quadratica* il valore $\sqrt{(x_1^2 + x_2^2 + \dots + x_n^2)/n}$,

3. *media armonica* il valore $n / \left(\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n} \right)$.

Scrivete un programma che legga una sequenza di numeri reali positivi in input, terminati da 0, e ne stampi le tre medie. Per scrivere il programma, potete usare le funzioni `sqrt()`, `exp()` e `log()`, che calcolano, rispettivamente, la radice quadrata, l'esponenziale e il logaritmo naturale del loro argomento. Per potere utilizzare queste funzioni, dovete inserire la dichiarazione `#include <math.h>`. Ricordatevi inoltre di compilare il programma con l'opzione `-lm`.

Esempio di funzionamento

2 4 5 -1

Media geometrica: 3.419952

Media quadratica: 3.872983

Media armonica: 3.157895

Divisori e numeri primi

1. Scrivete un programma che stampi la sequenza decrescente dei numeri divisori di n , dove n è un numero inserito in input. Modificate il programma in modo che stampi anche il numero di divisori di n .
2. Scrivete un altro programma che, usando un ciclo `for`, stabilisca se un numero n è primo (ovvero ha per divisori solo 1 e se stesso) oppure no. Cercate di ridurre il numero di istruzioni da eseguire! Scrivete una nuova versione del programma precedente usando un ciclo `while`.
3. Scrivete un programma che scomponga in fattori primi un numero dato in input.

Istogramma

Scrivete un programma che legga (con `getchar()`) una sequenza di caratteri terminata da `.` e che visualizzi un istogramma che rappresenti la frequenza di vocali, consonanti, cifre, spazi bianchi (blank, a capo, tabulazione) o altri caratteri. Per ciascuno dei tipi di caratteri presenti nel testo dovrà essere visualizzata una barra lunga quanto il numero delle occorrenze dei caratteri di quel tipo. Il programma non deve fare distinzione tra lettere maiuscole e minuscole e non deve visualizzare le barre dei tipi di caratteri mancanti.

Esempio di funzionamento

123: stella.

```
vocali      **
consonanti  ****
cifre       ***
spazi bianchi *
altro       *
```

C'era un RAGAZZO, che come me, amava i BEATLES e i ROLLING STONES.

```
Vocali: *****
Consonanti: *****
Spazi: *****
Altro: ***
```

Suggerimento. Dato un carattere alfabetico `c` per trasformarlo nel corrispondente carattere maiuscolo potete usare la funzione `toupper(c)`: essa restituisce il carattere maiuscolo corrispondente a `c` (se `c` è già maiuscolo, restituisce `c` stesso); per usare questa funzione è sufficiente includere il file `<ctype.h>`.