

Laboratorio di programmazione

16 novembre 2007

Sequenze

Si ricorda qui per comodità degli studenti che la notazione $[a, b]$ indica l'*intervallo* di numeri (interi, o reali, a seconda del contesto) compresi tra a e b estremi inclusi (ossia l'insieme di numeri maggiori, o uguali, ad a e minori, o uguali, a b); se nella notazione dell'intervallo una parentesi tonda sostituisce quella quadrata, si intende che l'estremo adiacente alla parentesi tonda non è compreso nell'intervallo, ad esempio: $[a, b)$ indica l'insieme di numeri maggiori, o uguali, ad a e (strettamente) minori di b .

Si ricorda inoltre che una sequenza di numeri x_1, x_2, \dots, x_n si dice (*strettamente*) *crescente* qualora $x_1 < x_2 < \dots < x_n$ e (*strettamente*) *decrescente* qualora $x_1 > x_2 > \dots > x_n$ (per completezza, si ricorda che qualora non tutte le disuguaglianze siano strette, ma valgano come \leq nel primo caso e \geq nel secondo, le sequenze si dicono, rispettivamente, *non decrescente* e *non crescente*).

Per ciascuno dei punti seguenti realizzare due programmi, uno basato su un ciclo `for` e l'altro su un ciclo `do-while`, che scrivano:

1. la sequenza crescente dei numeri interi compresi nell'intervallo $(-10, -2]$;
2. la sequenza decrescente dei numeri interi compresi nell'intervallo $[2, 10]$;
3. la sequenza decrescente dei numeri pari compresi nell'intervallo $[1, 99]$;

Per ciascuno dei punti seguenti realizzare un programma che scriva:

1. la sequenza crescente dei primi 10 quadrati perfetti (ossia numeri x tali che $x = y^2$ per qualche numero naturale y), realizzando il programma in modo che sia possibile ottenere da esso la sequenza dei primi 20 quadrati perfetti pari modificandolo in *un solo* punto nelle prime righe del programma;
2. la sequenza crescente di 11 numeri reali equispaziati nell'intervallo $[0, 1]$, ossia la sequenza

0.0 0.1 0.2 ... 0.9 1.0

realizzando il programma in modo che sia possibile ottenere da esso la sequenza crescente di 31 numeri reali equispaziati nell'intervallo $[0, 10]$ modificandolo solo nella parte di intestazione.

3. la sequenza crescente dei quadrati perfetti minori o uguali di un numero n inserito dall'utente.

Somma di numeri

Scrivete un programma che calcoli la somma di una serie di interi inseriti dall'utente, assumendo che la serie termini con 0.

Esempio di funzionamento

```
Inserisci i numeri (0 per terminare): 8 23 71 5 0
La somma è 107.
```

Calendario

Scrivete un programma che stampi un calendario mensile. L'utente deve specificare il nome del mese e il giorno della settimana in cui comincia il mese. Per semplicità considerate solo anni non bisestili...

Esempio di funzionamento

```
Inserisci il numero del mese (1 = gennaio, ..., 12 = dicembre): 2
Inserisci il giorno di inizio mese (1 = lunedì, ..., 7 = domenica): 4
L  M  M  G  V  S  D
   1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28
```

Il cifrario di Cesare

Svetonio nella *Vita dei dodici Cesari* racconta che Giulio Cesare usava per le sue corrispondenze riservate un codice di sostituzione molto semplice, nel quale la lettera chiara veniva sostituita dalla lettera che la segue di tre posti nell'alfabeto: la lettera A è sostituita dalla D, la B dalla E e così via fino alle ultime lettere che sono cifrate con le prime.

Più in generale si dice codice di Cesare un codice nel quale la lettera del messaggio chiaro viene spostata di un numero fisso k di posti, non necessariamente tre.

Dovete scrivere un programma che legga (usando `scanf`) in input il numero k (chiave di cifratura) e il testo da cifrare, sotto forma di una sequenza di caratteri terminata da `.` (usando `getchar`), e che emetta in output il testo cifrato; il programma deve cifrare solo le lettere dell'alfabeto, mantenendo minuscole le minuscole, e maiuscole le maiuscole, mentre deve lasciare inalterati gli altri simboli.

Per provare se il programma funziona, cifrate un messaggio con una certa chiave k e poi applicate al risultato una nuova cifratura con chiave $26 - k$: il risultato dovrebbe essere la stringa originale.

Esempio di funzionamento

```
5 mamma li Turchi.
rfrrf qn Yzwhmn
```

```
21 rfrrf qn Yzwhmn.
mamma li Turchi
```

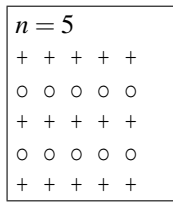
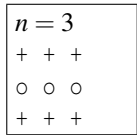
Disegni

Questo esercizio richiede di scrivere dei programmi che stampino delle figure, secondo gli schemi sotto definiti. L'utente dovrà inserire un intero n che definisce la dimensione della figura da disegnare.

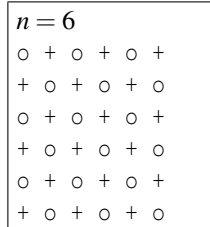
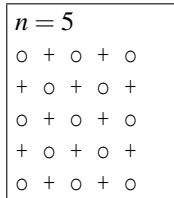
1. Quadrato: la figura è ottenuta da n righe costituite da n caratteri uguali:

$n = 3$ + + + + + + + + +	$n = 5$ +
------------------------------------	--

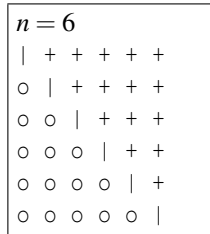
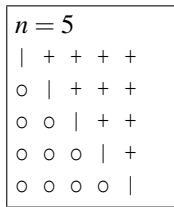
2. Righe alternate: la figura è ottenuta alternando righe costituite solo da + e righe costituite solo da o:



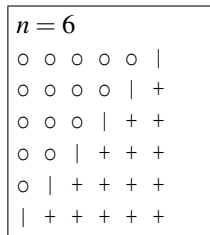
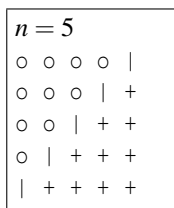
3. Caratteri alternati: la figura è ottenuta alternando caratteri o e +:



4. Triangolo: il triangolo sotto la diagonale con direzione alto/sx verso basso/dx è formato da o, il triangolo sopra la diagonale da +, la diagonale da |.



5. Triangolo bis: il triangolo sopra la diagonale con direzione bass/sx verso alto/dx è formato da o, il triangolo sotto la diagonale da +, la diagonale da |.



6. Rombo: la figura rappresenta un rombo avente i lati nei punti medi dei lati del quadrato. Notate che queste figure sono definite solo per n dispari!

