

Esercitazione

Oggi dovremo di nuovo usare funzioni sui reali. Ma il C non ha operatori nativi del linguaggio per queste operazioni, come la radice quadrata, l'elevamento a potenza o l'arrotondamento.

Per effettuare questa operazione si fa ricorso a funzioni della libreria matematica dichiarata con `#include <math.h>` che esporta le funzioni matematiche più comuni.

Alcune funzioni matematiche utili della libreria `<math.h>`:

`floor(x)`: il più grande intero non maggiore di x (per numeri maggiori di 0, la parte intera di x)

`ceil(x)`: il più piccolo intero non minore di x (per numeri maggiori di 0, la parte intera +1 di x)

`pow(x, y)`: x elevato alla y

`sqrt(x)`: radice quadrata di x

(vedere <http://it.wikipedia.org/wiki/Math.h> per ulteriori funzioni e dettagli)

Gli esercizi di oggi vanno risolti usando solamente costrutti iterativi. Per chi sapesse come usare gli array, questi non vanno usati per risolvere gli esercizi

Nella propria home directory creare una sottodirectory chiamata `es04`, in cui metteremo tutti i file C di oggi.

Esercizi:

Esercizio 1

Scrivere un programma che calcoli e visualizzi la somma degli interi pari da 2 a 1000 (estremi compresi)

Esercizio 2

Scrivere un programma che calcoli e visualizzi il prodotto degli interi multipli di 3 da 3 a 30

Esercizio 3

Scrivere un programma che permetta di caricare le coordinate di due punti nel piano e di calcolarne e stamparne la distanza euclidea tra i due punti.

Esercizio 4

Scrivere un programma che calcoli e stampi i primi N numeri della serie di Fibonacci, con N inserito da tastiera. La serie di Fibonacci inizia con 1, 1 ed ogni numero successivo è dato dalla somma dei due precedenti: 1, 1, 2, 3, 5, 8, 13, 21 . . .

Esercizio 5

Scrivere un programma che, letto un intero in input, permetta di:

- stampare a video il nome per esteso del mese;
- stabilire e stampare la stagione corrispondente

Esempio:

se l'utente digita 1, il programma deve stampare Gennaio - Inverno.

Qualora il mese sia Marzo, Giugno, Settembre e Dicembre, tutti mesi a cavallo di due stagioni, si richiada all'utente di specificare anche il giorno. Se il giorno e' compreso tra 1 e 20 si considera la stagione precedente altrimenti quella successiva.

E' necessario controllare la correttezza dell'intero in input, intendendo come correttezza il fatto che sia compreso tra 1 e 12.

Esercizio 6

Scrivere un programma tavola.c che:

- Chieda in ingresso un intero N;
- Stampi a video la tavola pitagorica per i fattori fino a N compreso.

Esempio:

```
"Inserisci il numero di righe della tavola pitagorica: 10"  
// Stampa della tavola pitagorica
```

```
TAVOLA PITAGORICA  
1 2 3 4 5 6 7 8 9 10  
2 4 6 8 10 12 14 16 18 20  
3 6 9 12 15 18 21 24 27 30  
4 8 12 16 20 24 28 32 36 40  
5 10 15 20 25 30 35 40 45 50  
6 12 18 24 30 36 42 48 54 60  
7 14 21 28 35 42 49 56 63 70  
8 16 24 32 40 48 56 64 72 80  
9 18 27 36 45 54 63 72 81 90  
10 20 30 40 50 60 70 80 90 100
```

Esercizio 7

Modificare il programma bisestile dato la volta scorsa in modo che continui a leggere valori finchè l'utente immette anni non bisestili e che si fermi quando l'utente immette un anno bisestile.

Esempio di interazione con il programma:

```
Dammi un anno (un intero): 1997  
1997 non e' bisestile!  
Dammi un anno (un intero): 1900  
1900 non e' bisestile!  
Dammi un anno (un intero): 1992  
Finalmente un anno bisestile!
```

Esercizio 8

Scrivere un programma che stampi un rettangolo la cui cornice sia costituita da caratteri asterisco e la parte interna da un carattere immesso dall'utente. Anche il numero di righe e di colonne viene deciso dall'utente.

Esempio d'uso (dati di input sottolineati):

```
Inserisci quante righe vuoi: 10  
Inserisci quante colonne vuoi: 30
```


Esempio di interazione con il programma:

Dammi un numero intero positivo compreso tra 0 e 9: -2

Dammi un numero intero positivo compreso tra 0 e 9: 5

```
1
212
32123
4321234
543212345
```

Esercizio 12

Scrivere un programma che chieda all'utente di inserire una sequenza di interi maggiori di 0 (terminata quando l'utente inserisce uno zero, che non fa parte della sequenza). Calcolare e stampare la somma e la media dei valori inseriti.

Esercizio 13

Scrivere un programma che chieda all'utente di inserire una sequenza di interi maggiori di 0 (terminata quando l'utente inserisce uno zero, che non fa parte della sequenza). Stampare il minimo e il massimo della sequenza.

Esercizio 14

Sulla base dell'esercizio visto durante la precedente esercitazione, scrivete un programma che calcola la media pesata dei voti di uno studente (la media pesata si calcola moltiplicando ogni voto per il suo peso in crediti, sommando tutti questi valori e dividendo per la somma del numero di crediti).

Quindi uno studente con due esami, fisica 1 (voto: 24, crediti:15) e informatica (voto: 28, crediti:6), avra' una media pesata circa uguale a $(24*15 + 28*6)/(15+6) = 25,14$.

Il programma deve quindi chiedere ad uno studente di inserire i voti degli esami e il loro peso in crediti, uno per volta. Lo studente dovrà inserire 0 per segnalare che ha terminato l'inserimento. Il programma quindi calcola e stampa la sua media pesata sui crediti. Nota: si tenga conto che la votazione del singolo esame e il numero di crediti sono interi. Inoltre sono votazioni valide per il superamento di un esame solo quelle comprese tra 18 e 30 (estremi inclusi) e il numero di crediti di un esame deve essere maggiore di 0.

Esercizio 15

Scrivere un programma alfabeto che chiede all'utente una sequenza di caratteri alfabetici minuscoli verificando che ogni carattere letto sia maggiore o uguale ai precedenti (secondo l'ordine alfabetico). Il primo carattere inserito può essere un qualsiasi carattere minuscolo. La sequenza termina quando l'utente immette un carattere non alfabetico o maiuscolo oppure se immette un carattere minore di uno di quelli letti precedentemente. Terminata la lettura dei caratteri il programma deve stampare il numero di caratteri minuscoli diversi appartenenti alla sequenza (il carattere che causa la terminazione non è considerato parte della sequenza). Se la sequenza è vuota, cioè non viene immesso alcun carattere minuscolo, allora il programma stampa solo un avvertimento.

Esempi di esecuzione:

Dammi un carattere: X

La sequenza di lettere minuscole e' vuota

Dammi un carattere: a

Dammi un carattere: r
Dammi un carattere: r
Dammi un carattere: 3
Totale lettere minuscole ordinate e diverse: 2

Dammi un carattere: a
Dammi un carattere: a
Dammi un carattere: a
Dammi un carattere: d
Dammi un carattere: z
Dammi un carattere: 4
Totale lettere minuscole ordinate e diverse: 3

Esercizio 16

Scrivere un programma rappresentazione che legge una sequenza di 0 e di 1 di dimensione prefissata K e stampa il numero intero la cui rappresentazione in complemento a 2 su K cifre è la sequenza letta.

Esempio di esecuzione :

Digitare una sequenza di 0 e 1 lunga 5: 0 1 1 1 0
Il numero intero e': 14

Nota: il programma deve definire K con un'istruzione define e deve poter funzionare anche con valori diversi da 5 semplicemente cambiando questa define.

Esercizio 17

Nella morra due giocatori si sfidano scegliendo un simbolo ciascuno tra sasso, forbici e carta: due simboli uguali pareggiano, mentre il sasso batte le forbici, le forbici battono la carta, e la carta batte il sasso. Scrivere il programma morra che gestisce una sfida tra PC e utente: (a) generando un numero casuale da 1 a 3 così definiti: 1: sasso, 2: forbici, 3: carta (utilizzare il costrutto #define per rendere leggibile l'associazione tra il numero e il simbolo) (b) leggendo un carattere ('s': sasso, 'f': forbici, 'c': carta) (c) stampando l'esito del confronto. Se l'utente immette un carattere diverso da 's', 'f' e 'c' allora perde comunque.

Esercizio 18

Si dice che una sequenza di interi non negativi è bilanciata se contiene lo stesso numero di pari e di dispari. Si scriva una funzione iterativa che legga una sequenza di interi non negativi (senza memorizzarla) che termina quando vengono immessi due numeri uguali e che, supponendo di non considerare l'ultimo come parte della sequenza, restituisca:

- 0 se la sequenza non è bilanciata, né ordinata in modo crescente;
- 1 se la sequenza è bilanciata, e non crescente;
- 2 se la sequenza non è bilanciata, ma è crescente;
- 3 la sequenza è bilanciata e crescente