

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 `es03`, in cui metteremo tutti i file C di oggi.

1) Scrivere il programma calcolatrice che legge un valore di tipo `double`, uno di tipo `char` e poi ancora uno di tipo `double`. Se il carattere letto è uno dei quattro segni di operazioni aritmetiche (`+`, `-`, `*`, `/`) il programma esegue

l'operazione
sui due valori e stampa il risultato.
Altrimenti stampa un messaggio di errore.
Stampare i risultati con al max 2 cifre decimali.

2) 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.

3) Ripetere l'esercizio precedente facendo barare il pc in modo che vinca i 2/3 delle partite.

Spunto di riflessione: cosa significa "i 2/3 delle partite" se il giocatore umano non gioca a caso?

Modificare eventualmente il programma perchè ogni umano, comunque bene giochi, perda sempre i 2/3 delle partite.

4) Scrivere un programma che esegue istruzioni for per stampare le seguenti

serie di numeri:

a) 1, 2, 3, 4, 5, 6, 7

b) 3, 8, 13, 18, 23

c) 20, 14, 8, 2, -4, -10

d) 19, 27, 35, 43, 51

5) Modificare il programma precedente in modo da usare solo

istruzioni
while per i cicli

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

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

8) 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.

9) 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.

10) Scrivere un programma rettangolo che legge due interi positivi A e B e quindi stampa un rettangolo di dimensioni AxB usando il carattere '*'.

Esempio di rettangolo 5*3:

```
*****  
*****  
*****
```

11) Scrivere un programma isoscele che chieda all'utente un intero n e stampi un triangolo isoscele di asterischi, di altezza lunga n e base lunga $2n-1$.
(Se il valore letto è negativo si consideri il suo valore assoluto).

Esempio di interazione con il programma:
Inserisci l'altezza: 6

```

      *
     ***
    *****
   *********
  ***********
 *****
*****
```

Suggerimento: per allineare correttamente le righe, stampare su ogni riga il giusto numero di spazi prima degli asterischi

12) Modificare il programma precedente per stampare un rombo vuoto.

Nota: accettare solo numeri dispari come altezza.

Esempio di interazione con il programma:

Inserisci l'altezza (dispari): 8

Inserisci l'altezza (dispari): 9

```

      *
     * *
    *  *
   *   *
  *    *
 *     *
*      *
 *     *
  *    *
   *   *
    *  *
     * *
```

13) Modificare il programma 17 per stampare un triangolo isoscele di altezza

lunga n (con n compreso tra 0 e 9) e base lunga $2n-1$ fatto come sotto.

Se il valore letto è non è compreso tra 0 e 9 si chieda all'utente un nuovo

valore finché l'intero immesso non appartenga all'intervallo richiesto.

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

14) Un triangolo rettangolo può avere tutti i lati di lunghezza intera.

Un insieme di tre valori interi per i lati di un triangolo rettangolo è

chiamato Tripletta Pitagorica. Questi tre lati devono soddisfare la condizione che

la somma del quadrato di due dei lati deve essere uguale al quadrato del terzo.

Scrivere un programma che trovi e stampi tutte le triplette pitagoriche in cui tutti

e tre i lati sono minori di 500.

Suggerimento: usare tre for annidati ed eseguire la ricerca in modo esaustivo, cioè

coprendo tutte le possibili triple.

15) Modificare il programma bisestile dato la volta scorsa in modo che continui

a leggere valori finché 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 è bisestile!

Dammi un anno (un intero): 1900

1900 non è bisestile!

Dammi un anno (un intero): 1992

Finalmente un anno bisestile!

16) La media pesata dei voti di uno studente 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 $(24*15 + 28*6)/(15+6) = 25,14$.

Scrivere il programma `calcola_la_media` che chiede ad uno studente 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.

17) Modificare il programma precedente per chiedere il numero di crediti del prossimo esame che lo studente deve sostenere. Il programma quindi calcola quale voto lo studente dovrebbe prendere perchè la media migliori raggiungendo la votazione (intera) immediatamente superiore alla media attuale. Se tale voto fosse maggiore di 30, il programma deve stampare il messaggio "Mi dispiace, non si puo'".

18) Esercizio proposto a lezione:
"Algoritmo di Euclide con i resti per il calcolo del MCD".
La descrizione informale dell'algoritmo è:

```
leggi m ed n
while m ed n sono entrambi != 0 {
    sostituisci il maggiore tra m ed n con il resto della divisione
del maggiore
    per il minore
}
stampa il numero tra i due che e' diverso da 0
```

Implementare questo algoritmo in C nel programma euclide.

19) 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: f
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

20) 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.

21) Esercizio proposto a lezione:
scrivi un programma che stampa la tavola pitagorica (come matrice quadrata, con righe da 1 a 10 e colonne da 1 a 10).
Sulle slide dell'ultima lezione c'è una possibile soluzione, chi non se la ricorda provi a ricostruirla.
Riuscite a stampare la tavola correttamente spaziata? (Suggerimento: guardate i dettagli del formato di printf).
Usate la barra verticale | per separare ogni colonna dalla successiva
e sequenze di segni meno - per separare ogni riga dalla successiva.

A lezione è stata proposta una soluzione con due cicli for annidati.

Riuscite a scrivere il programma con un unico ciclo for?
Ne vale la pena? Quali sono gli svantaggi di quest'ultima soluzione?

