

Esercitazione di Laboratorio - 2

Pagina del corso :

<http://didawiki.cli.di.unipi.it/doku.php/fisica/inf/start>

Oggi facciamo esercitazione su

- numeri reali (float, double, long double)
- rappresentazione dei numeri
- espressioni condizionali (if-else)

Il makefile da usare

- L'altra volta abbiamo usato questo makefile

```
# makefile
CC=gcc
CFLAGS=-Wall -g -O -pedantic -Wformat=2 -Wextra
```

- Stavolta dobbiamo modificarlo leggermente

```
# makefile
CC=gcc
CFLAGS=-Wall -g -O -pedantic -Wformat=2 -Wextra -lm
```

- Aggiungiamo l'opzione `-lm` che indica al linker di inserire la libreria matematica. Perché?

Radice quadrata ed elevamento a potenza

- Oggi dovremo usare funzioni sui reali.
- Ma il C non ha operatori nativi del linguaggio per la radice quadrata o l'elevamento a potenza
- Per effettuare questa operazione si fa ricorso a funzioni della libreria matematica dichiarata con `#include <math.h>` che esporta le funzioni matematiche più comuni.

Radice quadrata ed elevamento a potenza (2)

- In particolare:

```
double sqrt(double x);
```

```
double pow(double x, double y);
```

- La prima restituisce la radice quadrata di x
- La seconda restituisce x elevato alla y .
- In quest'ultimo caso, poichè y può essere un qualunque numero reale (nei limiti di rappresentabilità) la stessa funzione permette di calcolare anche le radici reali di x .
- E' ovvio, nel caso in cui il valore di y non corrisponda ad un numero intero, bisogna fare molta attenzione al segno di x .

printf

- A lezione sono stati introdotti un certo numero di segnaposto di formato di printf.
- I più comuni sono facili da ricordare
 - %d → interi
 - %f → float
 - %c → caratteri
- Ad ogni modo, ecco un elenco completo:
http://it.wikipedia.org/wiki/Printf#Segnaposti_del_formato_printf

Esercizi proposti

Nella propria home directory creare una sottodirectory chiamata es02, in cui metteremo tutti i file C di oggi.

- 1) Scrivere il programma `due_variabili` che dichiara due variabili, e gli assegna due valori in virgola mobile (a scelta) per inizializzarle, e ne stampa la somma, la differenza, il prodotto e la media.
I valori stampati devono essere accompagnati da opportuni commenti.

Esercizi proposti (2)

Esercizio proposto la scorsa volta (chi non l'ha fatto, lo faccia ora)

- Scrivere un programma **area_rettangolo** che dichiara due variabili che rappresentano i lati di un rettangolo, assegna a tali variabili due valori e stampa il perimetro e l'area del rettangolo risultante.
- Modificare il programma precedente in modo da leggere dall'input i valori delle dimensioni del rettangolo.

Eseguire il programma per un rettangolo di dimensioni 3 x 2.

- 2) Modificare il programma `area_rettangolo` per calcolare e stampare anche l'area di un quadrato avente lo stesso perimetro ed il perimetro di un quadrato avente la stessa area.
Usare la funzione **sqrt** per la radice quadrata
- 3) Modificare ancora il programma precedente in modo che verifichi che i valori immessi in input siano validi.
Se i valori inseriti non sono validi, il programma deve stampare un messaggio d'errore invece di effettuare i calcoli.

Esercizi proposti (3)

- 4) Scrivere il programma `tri_isoscele` che chiede all'utente due interi che rappresentano la base e l'altezza di un triangolo isoscele, li legge e stampa il perimetro e l'area del triangolo rappresentato.
Per calcolare la radice quadrata utilizzare la funzione **`sqrt`**.
- 5) Scrivere il programma `multiplo` che legga un intero e determini se è multiplo di 7, di 5 o di entrambi.
- 6) Un anno bisestile è identificato da un intero maggiore di 1584 che sia divisibile per 4 ma non per 100 oppure che sia divisibile per 400.
Scrivere il programma `bisestile` che letto un anno determini se tale anno è bisestile.

Esercizi proposti (4)

- 7) Scrivere il programma `ordina_tre` che legge tre interi e stamparli in ordine decrescente facendo il minimo numero di confronti possibile.
- 8) Scrivere il programma `voto_2_giudizio` che legge un valore intero v che rappresenta un voto e a seconda del valore letto stampa il giudizio corrispondente come specificato dalla tabella riportata sotto:

$30 < v$	"Valore illecito!"
$24 < v \leq 30$	"Ottimo"
$21 < v \leq 24$	"Buono"
$18 \leq v \leq 21$	"Sufficiente"
$0 \leq v < 18$	"Insufficiente"
$v < 0$	"Valore illecito!"

Esercizi proposti (5)

- 9) Scrivere il programma calcolatrice che legge un valore di tipo double, uno di tipo char e poi ancora uno di tipo double e infine fa in modo che:
- se il carattere letto è '+' allora stampa la somma dei due valori numerici;
 - se il carattere è '*' allora stampa il prodotto dei due valori numerici;
 - se è un altro carattere allora stampa il primo valore.
- 10) Scrivere il programma tipo_triangolo che legge tre valori interi che rappresentano le lunghezze dei lati di un triangolo, stabilisce se sono dati attendibili (la somma di due lati non minore dell'altro lato) e, in caso affermativo, se si tratta di un triangolo scaleno, isoscele o equilatero, stampando un opportuno messaggio.
- 11) Ripetere l'esercizio precedente utilizzando un algoritmo che conta il numero di lati uguali.

Esercizi proposti (6)

12) 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.c` 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)

(Per generare un numero pseudo-casuale, vedere la prossima slide)

(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.

Numeri pseudo-casuali

- Questo è il codice per generare un numero pseudo-casuale fra 1 e 3.

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>

int numero=0;
srand(time(NULL)); /*inizializza il generatore di numeri*/
numero=(rand()%3)+1;
```

- Notate l'operazione di modulo (%) : rand() restituisce un numero fra 0 e RAND_MAX.
- Senza "+1" il numero sarebbe compreso fra 0 e 2
- srand inizializza il generatore di numeri casuali con time(NULL), il numero di secondi trascorsi dall' "Epoca": [http://it.wikipedia.org/wiki/Tempo_\(Unix\)](http://it.wikipedia.org/wiki/Tempo_(Unix))

Esercizi proposti (7)

13) Ripetere l'esercizio precedente facendo barare il pc in modo che vinca i $\frac{2}{3}$ delle partite.

Spunto di riflessione: cosa significa "i $\frac{2}{3}$ delle partite" se il giocatore umano non gioca a caso?

14) Scrivere il programma tipo_bevanda che legge un valore in virgola mobile g che rappresenta la gradazione alcolica di una bevanda e a seconda del valore letto stampa il messaggio della tabella :

$50 < g$	"Roba forte!"
$33.3 < g \leq 50$	"Superalcolico"
$20 < g \leq 33.3$	"Alcolico"
$15 < g \leq 20$	"Vino liquoroso"
$15 < g \leq 20$	"Vino forte"
$10.5 < g \leq 12$	"Vino normale"
$0 < g \leq 10.5$	"Vino leggero"
$g = 0$	"Bevanda analcolica"