

Esercitazione di Laboratorio - 3

Pagina del corso :

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

Oggi facciamo esercitazione su

- Costrutti iterativi

while, do while, for

- Istruzioni di assegnamento

++(pre), --(pre), (post)++, (post)--,

+=, -=, *=, /=, %=

- Per chi sapesse usare gli array:
gli esercizi di oggi vanno risolti **SENZA** array.

Il makefile da usare

- Anche oggi useremo lo stesso makefile della scorsa volta

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

ossia quello modificato con l'aggiunta dell'opzione `-lm` (meno elle emme) per la libreria matematica.

- Copiatelo dalla cartella della scorsa volta in una nuova cartella `es03` della vostra home directory, dove metteremo i file di oggi.

printf

- Abbiamo usato l'istruzione printf già molte volte
- Anche printf (come scanf) ha un formato complesso degli specificatori di tipo:

`% [flag] [larghezza] [.precisione] [lunghezza] tipo`

- Il tipo è una delle consuete lettere:
d→interi, f→float, c→caratteri, ...
- Gli altri elementi sono opzionali ma talvolta molto utili.
- Un riferimento completo lo potete trovare qui:
http://it.wikipedia.org/wiki/Printf#Segnaposti_del_formato_printf

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))

Funzioni matematiche utili della libreria `<math.h>`

- `floor (x)` : il più grande intero non maggiore di x
- `ceil (x)` : il più piccolo intero non minore di x
- `trunc (x)` : la parte intera di x
- `round (x)` : l'intero più “vicino” ad x
- `fmod (x,y)` : resto della divisione intera fra numeri in virgola mobile
- `sqrt (x)` : radice quadrata di x
- `cbrt (x)` : radice cubica di x
- `pow (x,y)` : x elevato alla y

Per un elenco completo, <http://en.wikipedia.org/wiki/Math.h>

Esercizi proposti

1) 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).

2) 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`.

Esercizi proposti

- 3) Modificare il programma dell'esercizio 1 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".
- 4) Modificare ancora il programma precedente in modo da calcolare il numero minimo di esami (con lo stesso numero di crediti del prossimo da dare) in cui lo studente dovrebbe prendere 30 per alzare la media come nell'esercizio precedente. Inoltre individuato tale numero di esami, il programma stampi anche il voto minimo che è possibile ottenere in tutti questi esami garantendo comunque il raggiungimento del risultato (cioè alzare la media all'intero immediatamente superiore).

Esercizi proposti

5) 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.

Esempio di esecuzione nella prossima slide.

Esercizi proposti

5) (continua)

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

Esercizi proposti

- 6) 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

- 7) Scrivere un programma rettangolo che legge due interi positivi A e B e quindi stampa un rettangolo di dimensioni AxB usando il carattere '*'. Disegnare solamente il bordo del rettangolo.

Esempio di rettangolo 7x3:

* *

Esercizi proposti

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

- 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.

Esercizi proposti

8) Modificare il programma morra in modo che gestisca tornei tra il computer e l'utente al meglio delle 9 partite, secondo le seguenti regole:

a) una partita vinta vale 1 punto;

b) una partita pareggiata o persa vale zero punti;

c) vince il torneo il giocatore che raggiunge 5 punti (in questo caso il torneo si ferma anche se non tutte le 9 partite sono state disputate);

d) se la condizione precedente non si è verificata, vince il torneo il giocatore che, al termine delle 9 partite, ha totalizzato più punti;

Prima di uscire il programma indica il numero delle partite disputate, il punteggio ottenuto da ciascun giocatore e proclama il vincitore.

9) Modificare il programma precedente in modo che termini e proclami il vincitore, anche se non ha raggiunto i 5 punti, appena sia certo quale dei due giocatori vincerà'.

Esercizi proposti

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

10) Modificare il programma bisestile 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): 1853
```

```
1853 non e' bisestile!
```

```
Dammi un anno (un intero): 1992
```

```
Finalmente un anno bisestile!
```

Esercizi proposti

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
```

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

Esercizi proposti

12) Modificare il programma precedente 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
```

Esercizi proposti

13) 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).

A lezione è stata proposta una soluzione con due ciclo for annidati. Riuscite a scrivere il programma con un unico ciclo for?

Ne vale la pena? Quali sono gli svantaggi di quest'ultima soluzione?

Esercizi proposti

14) Scrivere un programma inverti che legga 9 numeri interi fra 0 e 9 e che li stampi in ordine inverso DOPO che l'utente ha terminato l'inserimento.

Il programma **non deve** usare array e **nemmeno** 9 variabili per ricordare i numeri.

Suggerimento: pensate alla notazione posizionale: cosa succede se moltiplico un numero per 10 e gliene sommo un altro compreso fra 0 e 9?

Esempio di interazione con il programma

```
Inserisci un intero (ne mancano ancora 9) : 4
Inserisci un intero (ne mancano ancora 8) : 6
Inserisci un intero (ne mancano ancora 7) : 3
Inserisci un intero (ne mancano ancora 6) : 2
Inserisci un intero (ne mancano ancora 5) : 1
Inserisci un intero (ne mancano ancora 4) : 0
Inserisci un intero (ne mancano ancora 3) : 8
Inserisci un intero (ne mancano ancora 2) : 4
Inserisci un intero (ne mancano ancora 1) : 5
```

```
Eccoti i tuoi numeri invertiti: 5 4 8 0 1 2 3 6 4
```

Esercizi proposti

15) Modificate il programma precedente perche' stampi la lista di numeri uno per riga e con accanto un equivalente numero di asterischi (vedi sotto). Esempio di esecuzione:

```
Inserisci un intero (ne mancano ancora 9) : 1
Inserisci un intero (ne mancano ancora 8) : 2
Inserisci un intero (ne mancano ancora 7) : 3
Inserisci un intero (ne mancano ancora 6) : 4
Inserisci un intero (ne mancano ancora 5) : 5
Inserisci un intero (ne mancano ancora 4) : 6
Inserisci un intero (ne mancano ancora 3) : 7
Inserisci un intero (ne mancano ancora 2) : 8
Inserisci un intero (ne mancano ancora 1) : 9
```

Eccoti i tuoi numeri asteriscati:

```
9 |*****
8 |*****
7 |*****
6 |*****
5 |*****
4 |****
3 |***
2 |**
1 |*
```