

# Laboratorio di programmazione

9 novembre 2007

## Gatti

Scrivete un programma che esibisca il comportamento schematizzato dalle due seguenti esecuzioni (in grassetto sono indicati i dati introdotti dall'utente):

Quanti gatti in tutto? **35**  
Quanti gatti in ogni fila? **6**  
35 gatti in fila per 6 col resto di 5

Quanti gatti in tutto? **128**  
Quanti gatti in ogni fila? **7**  
128 gatti in fila per 7 col resto di 2

## Lire del vecchio conio...

Scrivete un programma che dato in ingresso un prezzo in Euro (rappresentato per semplicità come un numero in virgola mobile a singola precisione), ne stampi l'equivalente in Lire. Si ricordi che 1 Euro = 1936,27 Lire.

## Ordine alfabetico

Scrivete un programma che date in ingresso due lettere maiuscole stampi la loro distanza nell'ordine alfabetico. Ad esempio, su ingresso **A C**, il programma deve stampare 3, su ingresso **F B**, il programma deve stampare 5. Ricordate che i caratteri in C sono rappresentati come (piccoli) interi e osservate che, secondo la codifica ASCII, non c'è soluzione di continuità e non ci sono altri caratteri tra la A e la Z (cosa di cui potete convincervi con il comando `man ascii`).

## Cerchio

La direttiva `#define` che, tra l'altro, consente di "dare un nome" alle costanti, ossia di associare al valore di una costante un identificatore che può comparire ovunque sarebbe stato legale che comparisse quella costante.

La possibilità di denominare le costanti è così comune che ci sono file che contengono, tra l'altro, la definizione di molte costanti utili, come, ad esempio, le costanti matematiche. Il file `math.h` (che potete istruire il precompilatore ad includere nel vostro sorgente con la direttiva `#include <math.h>`), ad esempio, definisce la costante `M_PI` pari al valore di  $\pi$  (il rapporto tra la circonferenza ed il diametro).

Facendo riferimento a quanto spiegato sin qui, scrivete un programma che legga (in una variabile di tipo `float`) il raggio di un cerchio e ne stampi l'area.

## Intervallo di tempo

Scrivete un programma che calcoli l'intervallo di tempo compreso tra due orari. Assumete che sia gli orari che l'intervallo di tempo siano rappresentati nel formato a 24 ore `hh:mm:ss`. E' possibile usare `short int` invece che `int`?

## Numeri interi e intervalli

Per ciascuno dei punti seguenti realizzare un programma che, una volta letto un numero intero (con una opportuna `scanf`), scriva (con una opportuna `printf`):

1. se il numero è positivo (o nullo), oppure negativo;
2. se il numero è negativo o, se positivo (o nullo), se è pari, o dispari;
3. se il numero appartiene, o meno, all'intervallo  $[10, 20]$ ;
4. la radice quadrata del numero (ottenuta con la funzione `sqrt`), se positivo (o nullo), oppure (per i numeri negativi) che non è possibile estrarre la radice quadrata;
5. “primo” se il numero è minore di 6, “secondo” se il numero appartiene a  $[6, 12]$ , “terzo” se appartiene a  $[13, 20]$  e “ultimo” se maggiore di 20;
6. “primo” se il numero appartiene a  $(0, 10]$ , “secondo” se appartiene a  $[5, 7]$  e “ultimo” se non appartiene a nessuno dei due precedenti intervalli;
7. se il numero è esterno all'intervallo  $(0, 2)$ .

## Classificazione triangoli

Scrivete un programma che, dati tre numeri, stabilisce se questi possono definire le lunghezze dei lati di un triangolo e, in caso affermativo, classifica il triangolo come equilatero, isoscele o scaleno.

## Massimo tra due numeri

Scrivete un programma che, dati due numeri interi, stabilisca qual è il massimo. Provate ad usare l'operatore condizionale `expr1 ? expr2 : expr3`

## Lettere

Scrivete un programma che, data una lettera minuscola, stabilisca se è una vocale dell'alfabeto italiano, una consonante dell'alfabeto italiano, una lettera che non fa parte dell'alfabeto italiano.

## Conversione orario

Scrivete un programma che, dato un orario in formato a 24 ore hh:mm, fornisca il corrispondente orario nel formato AM/PM e viceversa. Ricordate che secondo il formato AM/PM, le 24 ore del giorno sono suddivise in due periodi chiamati AM (ante meridiem) e PM (post meridiem): ogni periodo consiste di 12 ore numerate con 12 (usato come 0), 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11. L'orario 12:00 AM indica la mezzanotte, l'orario 12:00 PM indica il mezzogiorno.

## Indovina l'animale

Scrivete un programma che invita l'utente a scegliere un animale tra

anguilla, cane, delfino, gatto, leone, mucca, sogliola, tartaruga marina

poi indovini la scelta fatta dall'utente ponendo delle domande con risposta si/no. Cercate di ridurre al minimo il numero di domande!

## Operatore sizeof

L'operatore unario `sizeof` applicato ad una variabile ha per valore la dimensione della variabile a cui è applicato espressa in byte. Così, ad esempio (sull'architettura delle macchine presenti in laboratorio), se la variabile `x` è di tipo `int`, allora l'espressione `sizeof(x)` ha valore 4.

Scrivete un programma che dichiari una variabile per ciascuno dei tipi fondamentali e delle sue rispettive varianti `long` e `short` (qualora ammissibili), e ne stampi la dimensione in byte ottenuta tramite l'operatore `sizeof`.

## limits.h

Il file di intestazione `limits.h` contiene (tra le altre cose) le definizioni (ottenute grazie alla direttiva `#define`) dei valori limite, per l'architettura corrente, dei tipi interi e carattere. Ad esempio, il seguente programma stampa l'intervallo di valori possibili per le variabili di tipo intero con segno.

---

```
#include <stdio.h>
#include <limits.h>

int main( void ) {
    printf( "%d..%d\n", INT_MIN, INT_MAX );

    return 0;
}
```

---

Le definizioni dei limiti per alcuni degli altri tipi fondamentali si chiamano: `SCHAR_MIN`, `SCHAR_MAX` e `UCHAR_MAX` per il tipo carattere con e senza segno, `SHRT_MIN`, `SHRT_MAX` e `USHRT_MAX` per il tipo intero breve, con e senza segno. Modificate il programma precedente perché stampi l'intervallo di valori possibili per tali tipi.

## Frazioni continue

Se  $a_0$  è un numero intero qualsiasi, e  $a_1, a_2, \dots, a_n$  sono interi positivi, la notazione  $[a_0, a_1, \dots, a_n]$  sta per l'espressione

$$[a_0, a_1, \dots, a_n] = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{\ddots}}}}}$$

e viene chiamata *frazione continua*. Ad esempio,

$$[-1, 5, 2, 4] = -1 + \frac{1}{5 + \frac{1}{2 + \frac{1}{2 + \frac{1}{4}}}}$$

Ovviamente,  $[a_0, a_1, \dots, a_n]$  è un numero razionale, e si può dimostrare che per ogni numero razionale esiste una sola frazione continua che lo rappresenti a meno di eventuali 1 in fondo (poiché, ad esempio,  $[-1, 5, 2, 4] = [-1, 5, 2, 3, 1]$ ). Inoltre, per ogni numero irrazionale esiste un'unica frazione continua infinita che lo rappresenta.

Scegliete un valore di  $n$  (p.es.,  $n = 4$ ) e scrivete un programma che riceva in input  $a_0, a_1, \dots, a_n$  e stampi in output  $[a_0, a_1, \dots, a_n]$ . In particolare, usate questo programma per calcolare  $[1, 1, \dots, 1]$ . Che valore ottenete? Riuscite a immaginare che “celebre” valore rappresenta la frazione infinita  $[1, 1, 1, \dots]$ ?

**Suggerimento:** Se chiamate  $x$  il valore della frazione continua  $[1, 1, 1, \dots]$ , vale che

$$x = 1 + \frac{1}{x},$$

quindi...