Laboratorio di programmazione

25 ottobre 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

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. Come suggerimento, per leggere i due caratteri utilizzate la funzione `scanf( "%c %c", ... )` (sostituendo ai puntini gli opportuni nomi di variable preceduti da `&`). 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`).

Definizione di costanti

Durante una delle scorse lezioni abbiamo visto 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 π (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.

Tipi e dimensioni dei dati

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`.

Come spiegato a lezione, il file di intestazione `limits.h` contiene (tra le altre cose) le definizioni (ottenute grazie alla direttiva `#define` spiegata in precedenza) dei valori limite dei tipi fondamentali per l'architettura corrente. Ad esempio, il seguente programma stampa l'intervallo di valori possibili per le variabili di tipo intero con segno.

```c
#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.

Extraterrestri

L'esistenza di civiltà extraterrestri evolute è una delle questioni più affascinanti e dibattute fra gli scienziati di tutto il mondo, e la possibilità che tali civiltà esistano ha dato origine a un vero e proprio filone letterario e cinematografico. Il professor Frank Drake, oggi astrofisico all'Università di Santa Cruz, in California, ha tentato nel 1961 di stimare il numero N di civiltà extraterrestri con cui potremmo entrare in contatto; tale valutazione è basata su una formula, oggi nota come *formula di Drake*:

$$N = R^* \times f_p \times n_e \times f_\ell \times f_i \times f_c \times L.$$

In questa formula compaiono alcuni dati la cui valutazione è possibile in modo abbastanza accurato, e altri che possono essere solo ipotizzati:

- R^* è la frequenza media con cui si formano delle stelle nella nostra galassia; secondo la NASA e l'Agenzia Spaziale Europea, questo dato è circa 6 stelle/anno;
- f_p è la frazione di stelle che hanno un sistema di pianeti; il valore è molto incerto: secondo Drake $f_p = 0.5$ (cioè, il 50% delle stelle hanno un sistema planetario), ma fino ad ora si è potuto stabilire solo $f_p = 0.1$ (è però probabile che ciò sia dovuto semplicemente al fatto che le tecnologie di cui disponiamo non consentono di rilevare sistemi planetari molto piccoli o molto distanti);
- n_e è il numero medio di pianeti che possono ammettere la nascita della vita in ciascun sistema planetario; è molto difficile da stimare: l'assunzione di Drake è che $n_e = 2$, ma secondo alcuni si tratta di un'ipotesi troppo ottimistica;
- f_ℓ è la probabilità che, in un pianeta che può ammettere la nascita della vita, si sviluppi effettivamente qualche forma di vita; secondo Drake $f_\ell = 1$, mentre recenti studi di carattere statistico hanno stabilito che comunque $f_\ell > 0.13$, posto che si assuma che il pianeta in questione sia esistito per almeno un miliardo di anni;
- f_i è la probabilità che, in un pianeta in cui si sviluppi la vita, si abbia la nascita di vita intelligente; Drake stimò $f_i = 0.01$;
- f_c è la probabilità che una civiltà intelligente sia disponibile e in grado di comunicare; Drake stimò $f_c = 0.01$;
- L è il numero di anni per cui una civiltà intelligente e in grado di comunicare può sopravvivere; secondo Drake $L = 10000$.

Scrivete un programma che calcoli il valore della stima di Drake, prendendo in input i valori dei parametri incerti (definite i parametri per i quali ritenete che il valore sia sufficientemente certo come delle costanti, usando la direttiva `#define`), e fornendo in output N. Fate girare il programma alcune volte, prima inserendo i valori supposti da Drake, e poi inserendo le vostre valutazioni personali, che potranno essere più ottimistiche o pessimistiche di quelle ipotizzate da Drake. Che risultati ottenete?

Frazioni continue

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

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

e viene chiamata *frazione continua*. Ad esempio,

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

Ovviamente, $[a_0, a_1, \ldots, a_n]$ 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 $a_0, a_1, \ldots, a_n$ e stampi in output $[a_0, a_1, \ldots, a_n]$. In particolare, usate questo programma per calcolare $[1, 1, \ldots, 1]$. Che valore ottenete? Riuscite a immaginare che "celebre" valore rappresenta la frazione infinita $[1, 1, 1, \ldots]$?

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

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

quindi...