

Aula V di Matematica

Un po' di ripasso + qualcosa di nuovo

Igor Melatti

- Scrivere un programma che legge un numero non negativo x da tastiera e stampa su schermo a quanti giorni, ore, minuti e secondi corrispondono x secondi
- Scrivere un programma che legge da tastiera 4 numeri non negativi d , h , m , s e dice quanti secondi ci sono in d giorni, h ore, m minuti e s secondi

```
#include <stdio.h>

int main()
{
    unsigned x;
    printf("Immettere il numero di secondi: ");
    scanf("%u", &x);
    printf("x secondi corrispondono a %u giorni, %u ore, %u minuti e %u secondi\n",
        x/(60*60*24), x/(60*60)%24, x/60%60, x%60);
}
```

```
#include <stdio.h>

int main()
{
    unsigned x;
    printf("Immettere il numero di secondi: ");
    scanf("%u", &x);
    printf("x secondi corrispondono a %u giorni, %u ore, %u minuti e %u secondi\n",
        x/60/60/24, x/60/60%24, x/60%60, x%60);
}
```

```
#include <stdio.h>

int main()
{
    unsigned d, h, m, s;
    printf("Immettere il numero di giorni: ");
    scanf("%u", &d);
    printf("Immettere il numero di ore: ");
    scanf("%u", &h);
    printf("Immettere il numero di minuti: ");
    scanf("%u", &m);
    printf("Immettere il numero di secondi: ");
    scanf("%u", &s);
    printf("%u giorni, %u ore, %u minuti e %u secondi corrispondono a %u secondi\n",
        d, h, m, s, d*60*60*24 + h*60*60 + m*60 + s);
}
```

- Modificare i programmi di cui sopra per far sì che, se un certo numero è 1, allora il corrispondente sostantivo è declinato bene (facile...)
- Modificare i programmi di cui sopra in maniera tale che continui a ripetere il calcolo finché non viene immesso 0
 - nel caso della seconda traduzione, quando viene immesso 0 a tutti e 4 gli input
 - fatelo prima con un `while`
 - poi con un `do...while`
 - infine con un `for`
 - se vi volete fare del male, provate anche con il `goto`

- Scrivere un programma che stampi a video quanti bit e quanti bytes occorrono per i tipi predefiniti del C: int, unsigned, long, float, char...
- Scrivere dei programmi (iterativi) che chiedano da tastiera 2 interi x e y e poi stampino:
 1. il prodotto $x \times y$, ma *senza usare* l'operatore *
 2. la potenza x^y , ma *senza usare* la funzione `pow`
 3. il resto e il quoziente di x/y , ma *senza usare* né l'operatore `/`, né l'operatore `%`
- Riscrivere i programmi 1 e 2 in maniera che uno tra x e y sia reale (nel caso 2, deve essere per forza la base); stampare il risultato sia in notazione decimale, sia in notazione scientifica (virgola mobile), sia limitando il numero di cifre decimali dopo la virgola a 2 (questo sia per la notazione decimale che per quella scientifica)

```
#include <stdio.h>

int main()
{
    unsigned x, y, prod;
    printf("Immettere il moltiplicando: ");
    scanf("%u", &x);
    printf("Immettere il moltiplicatore: ");
    scanf("%u", &y);
    printf("%u*%u = ", x, y);
    for (prod = 0; y > 0; y--)
        prod += x;
    printf("%u\n", prod);
}
```



```
#include <stdio.h>

int main()
{
    unsigned x, y, pot;
    printf("Immettere la base: ");
    scanf("%u", &x);
    printf("Immettere l'esponente: ");
    scanf("%u", &y);
    printf("%u^%u = ", x, y);
    for (pot = 1; y > 0; y--)
        pot *= x;
    printf("%u\n", pot);
}
```

```
#include <stdio.h>

int main()
{
    unsigned x, y, div, mod;
    printf("Immettere il dividendo: ");
    scanf("%u", &x);
    printf("Immettere il divisore: ");
    scanf("%u", &y);
    printf("%u/%u = ", x, y);
    for (div = 0; (div + 1)*y <= x; div++); /* attenzione: il blocco di istruzioni
                                                all'interno del for e' vuoto! */
    printf("%u\t%u%%u = %u\n", div, x, y, x - div*y);
}
```

- `printf("%... ");`
- Al posto dei puntini, ci può essere il mondo, e tutto per stampare un solo valore numerico
- Per i nostri scopi, almeno per ora ci limitiamo a quanto segue
- Ci deve sempre essere uno dei seguenti caratteri: `d`, `u`, `x`, `e`, `f`, `g`
- I primi 3 sono per gli interi, gli altri per i float/double
- `d`, `u`, `x`: con segno, senza segno, in notazione esadecimale
- `e`, `f`, `g`: in notazione in virgola mobile, in notazione decimale, o l'una o l'altra a seconda delle circostanze

- Tra il % e uno dei caratteri di cui sopra, ci possono essere:
 - un numero che indichi la lunghezza del tutto
 - * se è più piccola si mettono degli spazi a sinistra
 - * serve se si vogliono allineare numeri diversi
 - un numero che indichi la precisione del numero (va preceduto da un punto)
 - * se si parla di un intero, è il minimo numero di cifre da scrivere
 - se sono di meno, se mettono degli 0 a sinistra
 - * se si tratta di un float/double, dice quanti numeri (esattamente) scrivere dopo la virgola
 - un modificatore di lunghezza
 - * per distinguere i long (l) e i short (h)
 - * nella printf, non ha importanza distinguere tra float e double
 - * per la scanf s

```
#include <stdio.h>

int main()
{
    unsigned u = 4;
    int i = -4;
    long l = 450;
    float f = 1e+40; /* troppo grande per un float */
    double d = 1e+40; /* ok per un double */
    printf("%.2u\n%5.2u\n", u, u);
    printf("%.2d\n%5.2d\n", i, i);
    printf("%.2ld\n%5.2ld\n", l, l);
    printf("%3.2f\n%3.2e\n", f, f);
    printf("%3.2f\n%3.2e\n", d, d);
    /* le 2 linee seguenti hanno lo stesso effetto delle 2 precedenti */
    printf("%3.2lf\n%3.2le\n", f, f);
    printf("%3.2lf\n%3.2le\n", d, d);
}
```

```
#include <stdio.h>

int main()
{
    unsigned y;
    double x, prod;
    printf("Immettere il moltiplicando: ");
    scanf("%lf", &x); /* per forza lf; se x fosse stato float, allora per forza f; la
                       printf non fa distinzioni */
    printf("Immettere il moltiplicatore: ");
    scanf("%u", &y);
    printf("%.3lf*%u = ", x, y);
    for (prod = 0; y > 0; y--)
        prod += x;
    printf("%.3le\n", prod);
}
```

- Scrivere un programma C che prende in input un intero x e restituisce la somma delle cifre (decimali) di x
 - Esempi: $23 \longrightarrow 5$; $2453 \longrightarrow 14$; $327652317 \longrightarrow 36$
- Riscrivere il programma di cui sopra facendo sì che prenda in input, oltre ad x , anche un intero b , e che poi consideri le cifre di x rappresentato in base b
- Scrivere un programma iterativo che dato in input un naturale x calcoli la radice digitale di x . La radice digitale di un numero si ottiene sommando le sue cifre (decimali) fino a che non si riducono ad un numero di una sola cifra (decimale)
 - Esempio. La radice digitale di 4366554 è 6.
- Riscrivere il programma di cui sopra facendo sì che prenda in input, oltre ad x , anche un intero b , e che poi consideri le cifre di x rappresentato in base b

- Un numero naturale è divisibile per 3 se la somma delle sue cifre decimali è divisibile per 3. Scrivere un programma iterativo per decidere se un naturale dato è divisibile per 3. È consentito l'uso dell'operatore modulo % *solo* per dividere per 10.
- Dato un naturale n , siano $si(n)$ e $sp(n)$ rispettivamente la somma delle cifre di ordine pari e dispari in n . (dove la 0-esima cifra è quella di minor peso). n è divisibile per 11 se il valore assoluto della differenza tra $sp(n)$ e $si(n)$ cioè $abs(sp(n)-si(n))$ è divisibile per 11. Scrivere un programma iterativo per decidere se un numero naturale dato è divisibile per 11. È consentito l'uso dell'operatore modulo % *solo* per dividere per 10.


```
#include <stdio.h>

int main()
{
    unsigned x, sum = 0, b;
    printf("Immettere il numero e la base: ");
    scanf("%u%u", &x, &b);
    printf("La somma delle cifre di %u considerato in base %u e': ", x, b);
    while (x != 0) {
        sum += x%b;
        x /= b;
    }
    printf("%u\n", sum);
}
```

```
#include <stdio.h>

int main()
{
    unsigned x, sum;
    printf("Immettere il numero: ");
    scanf("%u", &x);
    printf("La radice digitale di %u e': ", x);
    while (x > 9) {
        unsigned y = x;
        sum = 0;
        while (y != 0) {
            sum += y%10;
            y /= 10;
        }
        x = sum;
    }
    printf("%u\n", x);
}
```

- Scrivere un programma che legge un numero non negativo x da tastiera e stampa su schermo a quanti giorni, ore, minuti e secondi corrispondono x secondi
- Scrivere un programma che legge da tastiera 4 numeri non negativi d , h , m , s e dice quanti secondi ci sono in d giorni, h ore, m minuti e s secondi
- Scrivere un programma che stampi a video quanti bit e quanti bytes occorrono per i tipi predefiniti del C: int, unsigned, long, float, char...
- Scrivere dei programmi (iterativi) che chiedano da tastiera 2 interi x e y e poi stampino:
 1. il prodotto $x \times y$, ma *senza usare* l'operatore $*$
 2. la potenza x^y , ma *senza usare* la funzione `pow`
 3. il resto e il quoziente di x/y , ma *senza usare* né l'operatore $/$, né l'operatore $\%$

- Riscrivere i programmi 1 e 2 in maniera che uno tra x e y sia reale (nel caso 2, deve essere per forza la base); stampare il risultato sia in notazione decimale, sia in notazione scientifica (virgola mobile), sia limitando il numero di cifre decimali dopo la virgola a 2 (questo sia per la notazione decimale che per quella scientifica)
- Scrivere un programma C che prende in input un intero x e restituisce la somma delle cifre (decimali) di x
- Riscrivere il programma di cui sopra facendo sì che prenda in input, oltre ad x , anche un intero b , e che poi consideri le cifre di x rappresentato in base b

- Scrivere un programma iterativo che dato in input un naturale x calcoli la radice digitale di x . La radice digitale di un numero si ottiene sommando le sue cifre (decimali) fino a che non si riducono ad un numero di una sola cifra (decimale)
- Riscrivere il programma di cui sopra facendo sì che prenda in input, oltre ad x , anche un intero b , e che poi consideri le cifre di x rappresentato in base b
- Un numero naturale è divisibile per 3 se la somma delle sue cifre decimali è divisibile per 3. Scrivere un programma iterativo per decidere se un naturale dato è divisibile per 3. È consentito l'uso dell'operatore modulo % *solo* per dividere per 10.

- Dato un naturale n , siano $si(n)$ e $sp(n)$ rispettivamente la somma delle cifre di ordine pari e dispari in n . (dove la 0-esima cifra è quella di minor peso). n è divisibile per 11 se il valore assoluto della differenza tra $sp(n)$ e $si(n)$ cioè $abs(sp(n)-si(n))$ è divisibile per 11. Scrivere un programma iterativo per decidere se un numero naturale dato è divisibile per 11. È consentito l'uso dell'operatore modulo % *solo* per dividere per 10.