IL COMPUTER E IL CODICE BINARIO

PREPARATO DA:

ANDREA PIFERI

INTRODUZIONE

Il codice binario è il linguaggio fondamentale dei computer. Attraverso questo sistema numerico posizionale, i dispositivi elettronici comunicano tra loro e gestiscono informazioni. Vediamo più da vicino come funziona e perché è così cruciale nell'ambito dell'informatica.

Il Codice Binario: Fondamenti e Applicazioni

La Nascita del Codice Binario

Il matematico e filosofo tedesco Gottfried Wilhelm Leibniz concepì per la prima volta il codice binario nel 17° secolo. Voleva creare un sistema numerico semplice e universale basato esclusivamente su due cifre: 0 e 1. Solo con l'avvento dell'informatica e dei calcolatori elettronici, il codice binario avrebbe trovato la sua applicazione principale.

Struttura del Codice Binario

- **Bit:** Ogni cifra binaria è chiamata **bit** (binary digit) e può assumere solo i valori **0** o **1**. Un bit rappresenta un singolo elemento di informazione all'interno di un sistema digitale.
- **Byte:** Un insieme di **8 bit** forma un **byte**, l'unità di misura fondamentale per quantità di memoria o capacità di archiviazione in un computer. Ad esempio, un file di testo di **1 kilobyte (KB)** ha una dimensione approssimativa di **8.000 bit.**

Rappresentazione Numerica

- I numeri binari seguono un sistema posizionale simile a quello decimale, ma con la base **2** invece che **10.**
- Ogni cifra di un numero binario ha un peso corrispondente a una potenza di **2**.
- Ad esempio, il numero binario **1101** corrisponde al numero decimale **13**:
  - (1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 0 + 1 = 13).

Applicazioni del Codice Binario

- **Circuiti Digitali:** I dispositivi elettronici utilizzano il codice binario per rappresentare elettronicamente informazioni attraverso circuiti digitali.
- **Rappresentazione di Numeri e Dati**: Grazie a specifici codici e standard come **ASCII** e **Unicode**, le sequenze di bit rappresentano numeri, caratteri alfanumerici e simboli.
- **Programmazione e Linguaggi di Programmazione**: I linguaggi di programmazione si traducono in istruzioni binarie comprensibili dai computer.

In breve, il codice binario è il linguaggio universale dei computer, permettendo loro di elaborare dati e informazioni in modo efficiente e affidabile

Rappresentazione binaria dei numeri decimali

Numeri Decimali

I numeri decimali sono parte integrante della nostra vita quotidiana. Quando parliamo di misure, prezzi o qualsiasi altra quantità, spesso utilizziamo numeri decimali. Ad esempio, se diciamo “Giorgio è alto un metro e cinquantasei” o “Quel pacco di figurine costa 1 euro e 10 centesimi”, stiamo usando numeri decimali.

Definizione

Un numero decimale è formato da due parti separate da una virgola:

- La parte a sinistra della virgola è detta parte intera (ad esempio, in 12,724 la parte intera è 12).
- La parte a destra della virgola è detta parte decimale (in 12,724 la parte decimale è 724).

Lettura dei Numeri Decimali

Per leggere un numero decimale:

1. Parti dalla parte intera e prosegui con la parte decimale.
2. Usa la virgola come congiunzione “e”.

Ad esempio:

- 12,724 si legge come “dodici e settecentoventiquattro millesimi”.
- 57,13 si legge come “cinquantasette e tredici centesimi”.

Numeri Binari

Il sistema binario è un sistema numerico posizionale in base 2. A differenza del sistema decimale (in base 10), nel sistema binario le sole cifre utilizzate sono 0 e 1.
Questi numeri vengono chiamati numeri binari.
Il sistema binario è alla base dell'informatica e definisce il codice binario.
Ogni cifra binaria è chiamata bit (binary digit).
Ad esempio, il numero binario 1011 si legge come
"uno zero uno uno in base 2".

Tabella Numeri
Binari positivi (3 bit)

Eccoti una tabella che illustra le conversioni in codice binario utilizzando 3 bit per numeri positivi

Numero Decimale	Numero Binario (3 bit)
0	000
+1	001
+2	010
+3	011
+4	100
+5	101
+6	110
+7	111

Tabella Numeri
Binari positivi (4 bit)

Eccoti una tabella che illustra le conversioni in codice binario utilizzando 4 bit per numeri positivi

Numero Decimale	Numero Binario (4 bit)
+ 7	0111
+ 8	1000
+ 9	1001
+10	1010
+11	1011
+12	1100
+13	1101
+14	1110
+15	1111

Tabella Numeri
Binari negativi

Per rappresentare numeri negativi, esistono due metodi comuni:

Modulo e segno: Si guarda la cifra più a sinistra. Se è 0, il valore è positivo; se è 1, il valore è negativo.

Esempio:

Numero Binario		Numero Decimale
0111	=	+7
1111	=	-7

Tabella Numeri
Binari negativi

Complemento a due: Questo metodo è più utilizzato. Ecco come funziona:

1. Si rappresenta il numero positivo in base binaria.
2. Si inverte ogni bit (cambia 0 in 1 e viceversa) e si aggiunge 1 al risultato.
3. Se il primo bit è 1, il numero è negativo; se è 0, il numero è positivo.

Esempio:

15

in complemento a due con **8 bit:**

- 15 = 00001111

Tabella Numeri Binari negativi (3 bit)

Ecco una tabella con i numeri negativi rappresentati in 3 bit utilizzando il metodo del complemento a 2:

Numero Decimale	Numero Binario (3 bit)
0	000
-1	111
-2	110
-3	101
-4	100
-5	011
-6	010
-7	001

Tabella Numeri Binari negativi (4 bit)

Ecco una tabella con i numeri negativi rappresentati in 4 bit utilizzando il metodo del complemento a 2:

Numero Decimale	Numero Binario (4 bit)
- 7	1001
- 8	1000
- 9	1111
-10	1110
-11	1101
-12	1100
-13	1011
-14	1010
-15	1001

Rappresentazione dei numeri reali con virgola fissa

La rappresentazione in codice binario di numeri reali con virgola fissa è un metodo per rappresentare numeri decimali con precisione limitata utilizzando una notazione binaria.
In breve, la notazione a virgola fissa è utile per rappresentare numeri reali in modo approssimato utilizzando una quantità fissa di bit. È importante scegliere attentamente il numero di bit per bilanciare la precisione e la capacità di rappresentazione

Rappresentazione dei numeri reali con virgola fissa

Una possibile codifica dei numeri reali è quella in virgola fissa (fixed point) in cui si stabilisce un numero di cifre a disposizione per la parte intera, un nume-ro a disposizione per la parte frazionaria e un bit per rappresentare il segno. È detta a virgola fissa poiché la posizione della virgola (o punto decimale) è stabilita una volta per tutte, ovvero i posti per rappresentare le cifre dopo la virgola sono prefissati e quindi la virgola viene pensata sempre allo stesso posto.

Rappresentazione dei numeri reali con virgola fissa su 16 bit

SEGNO

7 bit per rappresentare la parte intera

8 bit per rappresentare la parte frazionaria

Quì è idealmente localizzata la virgola

Il numero minimo è:

$N_{min} = -1111111,11111111 = -(2^7-1),(2^8-1)$

Il numero massimo è:

$N_{max} = +1111111,11111111 = +(2^7-1),(2^8-1)$

Come abbiamo visto per gli interi, anche per i numeri reali ci riferiamo sempre a un sottoinsieme. Nella rappresentazione in virgola fissa i numeri sono rappresentati con una certa approssimazione. Per esempio 5,37 (periodico) 5,37373737373... assumerebbe la rappresentazione in virgola fissa +0000005.37373737 che ha un numero di cifre decimali inferiore a quelle reali. Analogamente, il numero irrazionale /=1,414213562... verrebbe troncato a +0000001.41421356.

Quindi occorre prestare attenzione, perché espressioni come 1/3+1/3+1/3=1, perfettamente valide in matematica, non sono generalmente valide se eseguite su un computer; infatti 1/3 verrebbe approssimato in +0000000.33333333 e quindi la somma delle tre frazioni dell'espressione darebbe per risultato +0000000.99999999, che è diverso da 1. Il sistema a virgola fissa, essendo basato su un numero di posti fisso per la rappresentazione delle cifre dopo la virgola, può comportare la perdita più o meno grave di numerose cifre significative e ha, quindi, campi di applicazione specifici.

Rappresentazione dei numeri reali con virgola mobile

Per risolvere gli inconvenienti della rappresentazione in virgola fissa è stata studiata la rappresentazione in virgola mobile (floating point), diventata poi quella più usata dai costruttori di computer.
L'idea è quella di far variare dinamicamente (il termine inglese floating significa "mobile", "fluttuante") la posizione della virgola. In questo caso il numero viene espresso secondo la notazione scientifica, cioè scisso in due parti:

Rappresentazione dei numeri reali con virgola mobile

> la mantissa, che rappresenta un numero compreso tra 0,100000... e 0,99999...;
> l'esponente, che indica la potenza di 10 per cui occorre moltiplicare la mantissa al fine di ottenere il numero che si intende rappresentare.

Consideriamo il numero 8,41. Per esprimerlo in notazione scientifica dobbiamo procedere con un'operazione di normalizzazione, che consiste nel portare il punto decimale immediatamente a sinistra della prima cifra diversa da zero della mantissa: il punto quindi si muove verso l'inizio della prima parte del numero, moltiplicandolo per un opportuno esponente.

Rappresentazione dei numeri reali con virgola mobile

Il numero 8,41 normalizzato diventa 0.841 x 10, che, a sua volta, può essere espresso come 0.841 x 10' = 0.841E+ 1 dove la lettera E sta per "10 elevato all'esponente" (+1).

Generalizzando, il criterio per rappresentare un numero in forma normalizzata è il seguente: ogni numero può essere rappresentato come prodotto di un numero decimale e di una potenza di 10, in modo tale che la parte intera del numero decimale risulti sempre uguale a 0 e la prima cifra decimale risulti sempre diversa da 0.

Rappresentazione dei numeri reali con virgola mobile

Per ottenere questo risultato, l'esponente di 10 è calcolato secondo la seguente regola: se il numero dato è in valore assoluto maggiore di 1, l'esponente è positivo e uguale al numero delle cifre intere; se il numero dato è in valore assoluto minore di 1, l'esponente è negativo e uguale al numero degli zeri dopo la virgola. Vediamo qualche esempio:

812 = 0.812E+ 3
-812 =-0.812E+ 3
-0.007 =-0.7E- 2

Osserviamo che il numero 0 non può obbedire a questa regola e occorre quindi stabilire la sua forma normalizzata, per esempio: 0.0.

Rappresentazione dei numeri reali con virgola mobile

Oggi giorno non sono più i singoli progettisti o produttori di sistemi a stabilire il codice da usare per la rappresentazione floating point dei numeri all'interno di un sistema di calcolo.
La soluzione di questo problema ha ormai raggiunto un livello di maturità tale da consentire a organismi internazionali di definire degli standard a cui tutti i progettisti e costruttori si devono adeguare.
In particolare, lo standard più diffuso è quello stabilito dall'IEEE (Institute of Electrical and Electronics Engineers) e identificato dalla sigla 754.

Rappresentazione dei numeri reali con virgola mobile

Questo standard prevede due formati di rappresentazione, uno in precisione normale codificato su 32 bit e uno in doppia precisione codificato su 64 bit. Inoltre lo standard prevede che:

› la mantissa venga rappresentata in modulo e segno;

› l'esponente (o caratteristica) venga rappresentato in modo polarizzato.

Abbiamo appena detto che lo standard IEEE 754 prevede 32 bit per rappresentare un valore in virgola mobile a precisione singola. I 32 bit a disposizione saranno così organizzati:

In particolare:

> 1 bit per il segno;

> 8 bit di esponente polarizzato biased 127 (detto anche in eccesso 127);

> 23 bit di parte frazionaria (mantissa).

Rappresentazione dei numeri reali con virgola mobile

All'interno di questi 32 bit, quindi, sarà inserita una successione di cifre binarie il cui valore sarà quello del numero reale. Rappresentiamo in singola precisione il numero reale decimale $43{,}687500_{(10)}$.

I passi da seguire sono i seguenti:

> Calcolare il segno del numero. Per il segno è necessario inserire 1 se il numero è negativo e 0 se è positivo. Nel nostro caso, essendo il numero positivo, il valore da inserire nel bit riservato al segno è 0.

> Trasformare il numero senza segno in forma binaria. Utilizzando il metodo delle divisioni successive, convertiamo la parte intera 43 in binario.

Rappresentazione dei numeri reali con virgola mobile

Dividendo	Divisore	Quoziente	Resto
43	2	21	1
21	2	10	1
10	2	5	0
5	2	2	1
2	2	1	0
1	2	0	1

Quindi : $43_{(10)}$ = $101011_{(2)}$.

Rappresentazione dei numeri reali con virgola mobile

Procediamo, ora, con la conversione della parte decimale 0,6875 servendoci del metodo delle moltiplicazioni successive:

0,6875 x 2 = 1,375 parte intera 1
0,375 x 2 = 0,75 parte intera 0
0,75 x 2 = 1,5 parte intera 1
0,5x2=1,0 parte intera 1

Quindi la parte decimale

$0{,}6875_{(10)} = 1011_{(2)}$.

Complessivamente

$43{,}687500_{(10)} = 101011.1011_{(2)}$.

Rappresentazione dei numeri reali con virgola mobile

> Normalizzare il numero binario ottenuto.
Spostiamo la virgola verso sinistra lasciando solo un 1 a sinistra.

Otteniamo $1.010111011 \cdot 2^5$.

> Completare la mantissa sino a coprire tutti i bit a disposizione.
Stiamo rappresentando un numero reale in precisione singola per cui abbiamo a disposizione 23 bit per la mantissa.
Dobbiamo inserire in coda al numero ottenuto (cioè alla sua destra) tanti zeri sino a completare i 23 bit.

Otteniamo, quindi

1.01011101100000000000000.

In particolare:

> 1 bit per il segno;

> 8 bit di esponente polarizzato biased 127 (detto anche in eccesso 127);

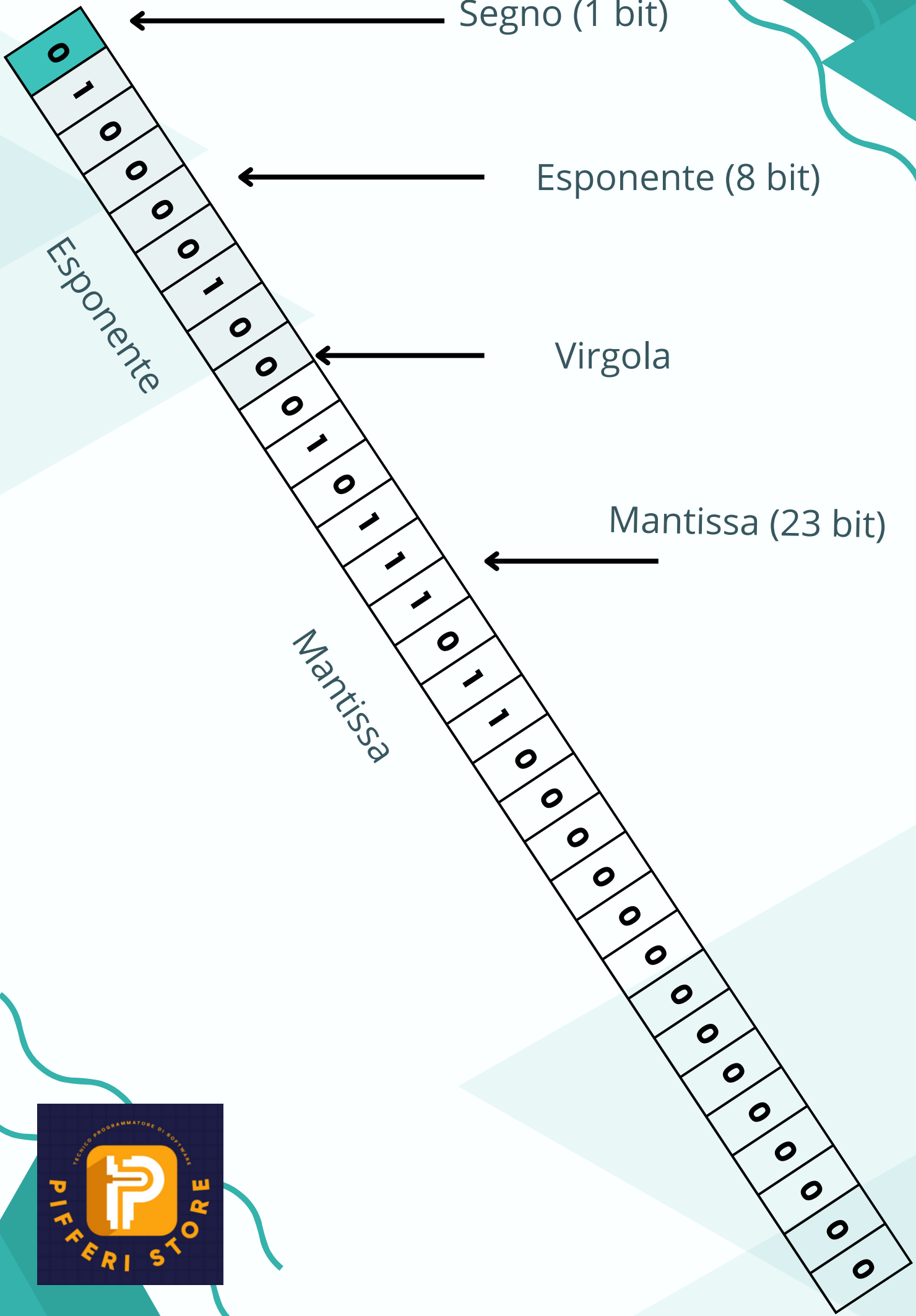
> 23 bit di parte frazionaria (mantissa).

Rappresentazione dei numeri reali con virgola mobile

> Convertire l'esponente. L'esponente che abbiamo ottenuto durante la fase di normalizzazione è pari a 5. Secondo lo standard IEEE 754, l'esponente deve essere polarizzato nel senso che al valore dell'esponente originario oc-corre sommare un certo valore fisso chiamato bias. Nel caso di numeri in precisione singola, il bias è pari a 127.

Abbiamo quindi :

5 + 127 = 13200) = 10000100(2).


Segno (1 bit)
Esponente (8 bit)
Virgola
Mantissa (23 bit)
Esponente
Mantissa
0 1 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0

GRAZIE MILLE

Spero ora di aver fatto chiarezza su come si legge e calcola un Codice Binario

Non esitare a contattarmi per migliorare la tua visibilità sul web
+39 333 38 83 156 tecsvil.soft@libero.it