

Programmazione I

Lezione 14

Mini-test

<https://forms.gle/W5Wrex2jtgyKmopB9>

Torniamo alla rappresentazione dei dati

Overview:

- Codifica tipi int; l'idea della rappresentazione in complemento a 2
- Tipi float e memoria; lo standard IEEE 754
- Tipi rune; ASCII, UNICODE e UTF8
- Tipi string

Approfondimento: base 16 (esadecimale)

Base 10	Base 2	Base 16
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Torniamo alla rappresentazione dei dati:

- Esercizio preparatorio:
contare le cifre di un numero intero, e stamparne la somma
- Esercizio preparatorio:
dalla base 10 alla base 2
dalla base 2 alla base 10
dalla base 2 alla base 16
- Stampa formattata in go e “format verbs”:

%b base 2

%d base 10

%x base 16, with lower-case letters for a-f

%X base 16, with upper-case letters for A-F

Rappresentazione complemento a 2

	<div>Bit di segno</div>		
valore di partenza	0100 1100	76	1011 0100 -76
(Puro flip di ogni bit)	(1011 0011)		(0100 1011)
complemento a 2 (flip + 0000 0001)	1011 0100	-76	0100 1100 76
	0100 1100 +		
	1011 0100 =		
	0000 0000		

Recap: il sistema dei tipi

- Tipi “base” in GO
 - `bool`
 - `string`
 - `int int8 int16 int32 int64`
 - `uint uint8 uint16 uint32 uint64 uintptr`
 - `byte // alias for uint8`
 - `rune // alias for int32, represents a Unicode code point`
 - `float32 float64`
 - `complex64 complex128`
- `int` e `uint` sono “implementation dependent” (32 o 64 bit)
- Bit, byte, word, allineamento dei dati in memoria (libreoffice)