

## RAPPRESENTAZIONE DELL'INFORMAZIONE

- Internamente a un elaboratore, ogni informazione è rappresentata tramite *sequenze di bit (cifre binarie)*
- Una sequenza di bit *non dice "cosa" essa rappresenta: l'interpretazione è negli occhi di chi guarda*
- Ad esempio, 01000001 può rappresentare:
  - l'intero 65, il carattere 'A', il boolean 'vero', ...
  - ... il valore di un segnale musicale,
  - ... il colore di un puntino sullo schermo...

numeri naturali

1

## INFORMAZIONI NUMERICHE

- La rappresentazione delle *informazioni numeriche* è di particolare rilevanza
- In questa sede ci limiteremo ai *numeri naturali (interi senza segno)*

**Dominio:  $N = \{ 0,1,2,3, \dots \}$**

numeri naturali

2

## NUMERI NATURALI (interi senza segno)

- **Dominio:**  $N = \{ 0,1,2,3, \dots \}$
- **Rappresentabili con diverse notazioni**
  - ♦ **non posizionali**
    - ♦ ad esempio la notazione romana: I, II, III, IV, V, .... IX, X, XI...
  - ♦ **posizionale**
    - ♦ 1, 2, .. 10, 11, ... 200, ...

numeri naturali

3

## NUMERI NATURALI (interi senza segno)

- **Dominio:** **Non posizionali:** hanno regole proprie, che rendono spesso assai complessa l'esecuzione dei calcoli
- **Rappresentabili con diverse notazioni**
  - ♦ **non posizionali**
    - ♦ ad esempio la notazione romana: I, II, III, IV, V, .... IX, X, XI...
  - ♦ **posizionale**
    - ♦ 1, 2, .. 10, 11, ... 200, ...

**Posizionale:** rappresenta i numeri in modo compatto, e rende semplice l'effettuazione dei calcoli

4

## NOTAZIONE POSIZIONALE

- Concetto di **base di rappresentazione  $B$**
- Rappresentazione del numero come **sequenza di simboli (cifre)** appartenenti a un **alfabeto di  $B$  simboli distinti**
- **ogni simbolo rappresenta un valore compreso fra 0 e  $B-1$**

Esempio di rappresentazione su  $N$  cifre:

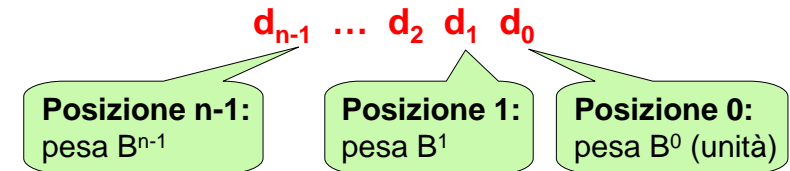
$d_{n-1} \dots d_2 d_1 d_0$

numeri naturali

5

## NOTAZIONE POSIZIONALE

- Il **valore di un numero** espresso in questa notazione è ricavabile
  - ♦ a partire dal valore rappresentato da ogni simbolo
  - ♦ **pesandolo in base alla posizione** che occupa nella sequenza



numeri naturali

6

## NOTAZIONE POSIZIONALE

In formula:

dove 
$$v = \sum_{k=0}^{n-1} d_k B^k$$

♦  $B = \text{base}$

♦ ogni cifra  $d_k$  rappresenta un valore fra 0 e  $B-1$

Esempio (base  $B=4$ ):

1 2 1 3

$d_3 d_2 d_1 d_0$

Valore =  $1 * B^3 + 2 * B^2 + 1 * B^1 + 3 * B^0 = \text{centotre}$

numeri naturali

7

## NOTAZIONE POSIZIONALE

- Quindi, **una sequenza di cifre non è interpretabile** se non si precisa **la base** in cui è espressa
- Esempi:

Stringa	Base	Alfabeto	Calcolo valore	Valore
"12"	quattro	{0,1,2,3}	$4 * 1 + 2$	sei
"12"	otto	{0,1,...,7}	$8 * 1 + 2$	dieci
"12"	dieci	{0,1,...,9}	$10 * 1 + 2$	dodici
"12"	sedici	{0,...,9, A,., F}	$16 * 1 + 2$	diciotto

numeri naturali

8

## NOTAZIONE POSIZIONALE

- Inversamente, ogni numero può essere espresso, *in modo univoco*, **come sequenza di cifre in una qualunque base**
- Esempi:

Numero	Base	Alfabeto	Rappresentazione
venti	due	{0,1}	"10100"
venti	otto	{0,1,...,7}	"24"
venti	dieci	{0,1,...,9}	"20"
venti	sedici	{0,..,9, A,, F}	"14"

numeri naturali

9

## CONCLUSIONE

Quindi:

- *i numeri sono concetti*, che esistono in quanto tali
- la loro *rappresentazione* può invece variare a seconda delle convenzioni adottate

**Non bisogna confondere un numero con una sua rappresentazione!**

numeri naturali

10

## NUMERI E LORO RAPPRESENTAZIONE

- Internamente, un elaboratore adotta per i numeri interi (non negativi) una *rappresentazione binaria* (base  $B=2$ )
- Esternamente, le costanti numeriche che scriviamo nei programmi e i valori che stampiamo a video / leggiamo da tastiera sono invece *sequenze di caratteri ASCII*

Il passaggio dall'una all'altra forma richiede dunque un processo di **conversione**.

numeri naturali

11

## Tabella ASCII

0 NUL	1 SOH	2 STX	3 ETX	4 EOT	5 ENQ	6 ACK	7 BEL
8 BS	9 HT	10 NL	11 VT	12 NP	13 CR	14 SO	15 SI
16 DLE	17 DC1	18 DC2	19 DC3	20 DC4	21 NAK	22 SYN	23 ETB
24 CAN	25 EM	26 SUB	27 ESC	28 FS	29 GS	30 RS	31 US
32 SP	33 !	34 "	35 #	36 \$	37 %	38 &	39 '
40 (	41 )	42 *	43 +	44 ,	45 -	46 .	47 /
48 0	49 1	50 2	51 3	52 4	53 5	54 6	55 7
56 8	57 9	58 :	59 ;	60 <	61 =	62 >	63 ?
64 @	65 A	66 B	67 C	68 D	69 E	70 F	71 G
72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O
80 P	81 Q	82 R	83 S	84 T	85 U	86 V	87 W
88 X	89 Y	90 Z	91 [	92 \	93 ]	94 ^	95 _
96 `	97 a	98 b	99 c	100 d	101 e	102 f	103 g
104 h	105 i	106 j	107 k	108 l	109 m	110 n	111 o
112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w
120 x	121 y	122 z	123 {	124	125 }	126 ~	127 DEL

000-037: caratteri di controllo, non ortografici  
Viene mantenuto l'ordinamento alfabetico

numeri naturali

12

## ESEMPIO (interno / esterno)

- Numero: *centoventicinque*
- Rappresentazione interna binaria (16 bit):

00000000 01111101

- Rappresentazione esterna in base 10:

occorre produrre la *sequenza di caratteri ASCII* '1', '2', '5'

00110001 00110010 00110101

numeri naturali

vedi tabella ASCII

## ESEMPIO (esterno / interno)

- Rappresentazione esterna in base 10:

È data la *sequenza di caratteri ASCII* '3', '1', '2', '5', '4'

vedi tabella ASCII

00110011 00110001 00110010 00110101 00110100

- Rappresentazione interna binaria (16 bit):

01111010 00010110

- Numero:  
*trentunomiladuecentocinquantaquattro*

numeri naturali

14

## CONVERSIONE STRINGA / NUMERO (da "esterno" a "interno")

Si applica la definizione:

$$v = \sum_{k=0}^{n-1} d_k B^k$$

le cifre  $d_k$  sono note, il valore  $v$  va calcolato

$$= d_0 + B * ( d_1 + B * ( d_2 + B * ( d_3 + ... )))$$

Ciò richiede la valutazione di un polinomio  
→ *Metodo di Horner*

numeri naturali

15

## IMPLEMENTARE GLI ALGORITMI

### *Conversione da stringa a numero*

- **una funzione**: `stringa2numero`
- **in ingresso**:
  - base (base)
  - stringa di simboli, (stringa)
- **in uscita**:
  - il valore del numero (intero senza segno)

numeri naturali

16

## IMPLEMENTARE GLI ALGORITMI

```
function numero=stringa2numero(stringa,base)

numero=0;

<finché ci sono lettere nella stringa>
    <converti in numero la prima cifra
    della stringa>
    <aggiorna il valore di numero>
    <togli la prima cifra dalla stringa>
```

## IMPLEMENTARE GLI ALGORITMI

```
function numero=stringa2numero(stringa,base)

numero=0;

while length(stringa)>0
    <converti in numero la prima cifra
    della stringa>
    <aggiorna il valore di numero>
    <togli la prima cifra dalla stringa>
end
```

## IMPLEMENTARE GLI ALGORITMI

```
function numero=stringa2numero(stringa,base)

numero=0;

while length(stringa)>0
    cifra=valorecifra(stringa(1));
    <aggiorna il valore di numero>
    <togli la prima cifra dalla stringa>
end
```

## IMPLEMENTARE GLI ALGORITMI

```
function numero=stringa2numero(stringa,base)

numero=0;

while length(stringa)>0
    cifra=valorecifra(stringa(1));
    numero=base*numero+cifra;
    <togli la prima cifra dalla stringa>
end
```

## IMPLEMENTARE GLI ALGORITMI

```
function numero=stringa2numero(stringa,base)

numero=0;

while length(stringa)>0
    cifra=valorecifra(stringa(1));
    numero=base*numero+cifra;
    stringa(1)=[ ];
end
```

numeri naturali

21

## IMPLEMENTARE GLI ALGORITMI

```
function numero=stringa2numero(stringa,base)

numero=0;

while length(stringa)>0
    cifra=valorecifra(stringa(1));
    numero=base*numero+cifra;
    stringa(1)=[ ];
end
```

valorecifra ??

numeri naturali

22

## IMPLEMENTARE GLI ALGORITMI

```
function cifra=valorecifra(carattere)
```

*Come fare per calcolare il valore rappresentato da carattere?*

- Il carattere è rappresentato internamente da un numero, secondo la codifica ASCII
- È garantito che i caratteri da '0' a '9' sono in sequenza:
  - se '0' è rappresentato internamente dal numero  $\alpha$
  - '1' deve essere rappresentato dal numero  $\alpha + 1$ ,
  - '2' deve essere rappresentato dal numero  $\alpha + 2$ ,
  - ...
  - '9' deve essere rappresentato dal numero  $\alpha + 9$

numeri naturali

23

## IMPLEMENTARE GLI ALGORITMI

```
function cifra=valorecifra(carattere)
```

*Come fare per calcolare il valore rappresentato da carattere?*

La differenza **carattere - '0'** dà come risultato proprio il numero corrispondente al valore rappresentato dal carattere

- Il carattere è rappresentato internamente da un numero, secondo la codifica ASCII
- È garantito che i caratteri da '0' a '9' sono in sequenza:
  - se '0' è rappresentato internamente dal numero  $\alpha$
  - '1' deve essere rappresentato dal numero  $\alpha + 1$ ,
  - '2' deve essere rappresentato dal numero  $\alpha + 2$ ,
  - ...
  - '9' deve essere rappresentato dal numero  $\alpha + 9$

numeri naturali

24

## IMPLEMENTARE GLI ALGORITMI

```
function cifra=valorecifra(carattere)
if carattere>='0' & carattere<='9'
    cifra=carattere-'0';
end
```

## CONVERSIONE NUMERO / STRINGA (da "interno" a "esterno")

- Problema: **dato un numero, determinare la sua rappresentazione in una base data**
- Soluzione (notazione posizionale): **manipolare la formula** per dedurre un algoritmo

$$v = \sum_{k=0}^{n-1} d_k B^k$$

*v è noto, le cifre  $d_k$  vanno calcolate*

$$= d_0 + B * ( d_1 + B * ( d_2 + B * ( d_3 + ... )))$$

## CONVERSIONE NUMERO / STRINGA (da "interno" a "esterno")

- Per trovare le cifre bisogna *calcolarle una per una*, ossia...
- ... bisogna trovare un modo per *isolarne una dalle altre*

$$v = d_0 + B * (...)$$

### Osservazione:

**$d_0$  è la sola cifra non moltiplicata per  $B$**

### Conseguenza:

**$d_0$  è ricavabile come  $v$  modulo  $B$**

## CONVERSIONE NUMERO / STRINGA

### **Algoritmo delle divisioni successive**

- si divide  $v$  per  $B$ 
  - *il resto costituisce la cifra meno significativa ( $d_0$ )*
  - *il quoziente serve a iterare il procedimento*
- se tale quoziente è zero, l'algoritmo termina;
- se non lo è, lo si assume come nuovo valore  $v'$ , e si itera il procedimento con il valore  $v'$ .

## Un esempio

- **Esempio:** Convertire in forma binaria  $N_{10}=44$

Divisione	Quoziente	Resto
44 : 2	22	0
22 : 2	11	0
11 : 2	5	1
5 : 2	2	1
2 : 2	1	0
1 : 2	0	1

Si legge dal basso:  $(44)_{10} = (101100)_2$

numeri naturali

29

## CONVERSIONE NUMERO / STRINGA

### Esempi

Numero	Base	Calcolo valore	Stringa
<i>quattordici</i>	4	14 / 4 = 3 con resto 2 3 / 4 = 0 con resto 3	“32”
<i>undici</i>	2	11 / 2 = 5 con resto 1 5 / 2 = 2 con resto 1 2 / 2 = 1 con resto 0 1 / 2 = 0 con resto 1	“1011”
<i>sessantatre</i>	10	63 / 10 = 6 con resto 3 6 / 10 = 0 con resto 6	“63”
<i>sessantatre</i>	16	63 / 16 = 3 con resto 15 3 / 16 = 0 con resto 3	“3F”

numeri naturali

30

## IMPLEMENTARE GLI ALGORITMI

### Conversione da numero a stringa

- una funzione: `numero2stringa`
- in ingresso:
  - base b
  - numero n
- in uscita:
  - stringa di simboli,

numeri naturali

31

## IMPLEMENTARE GLI ALGORITMI

```
function stringa=numero2stringa(numero,base)
```

```
stringa='';
```

```
<Finché numero>0>
```

```
<Calcola il resto di numero/base>
```

```
<Converti in carattere il resto>
```

```
<aggiungi il carattere in testa alla stringa>
```

```
<sostituisci a numero il quoziente>
```

numeri naturali

32



## IMPLEMENTARE GLI ALGORITMI

```
function stringa=numero2stringa(numero,base)

stringa='';

while numero>0
    <Calcola il resto di numero/base>
    <Converti in carattere il resto>
    <aggiungi il carattere in testa alla
    stringa>
    <sostituisci a numero il quoziente>
end
```

numeri naturali

33

## IMPLEMENTARE GLI ALGORITMI

```
function stringa=numero2stringa(numero,base)

stringa='';

while numero>0
    resto=rem(numero,base);
    <Converti in carattere il resto>
    <aggiungi il carattere in testa alla
    stringa>
    <sostituisci a numero il quoziente>
end
```

numeri naturali

34

## IMPLEMENTARE GLI ALGORITMI

```
function stringa=numero2stringa(numero,base)

stringa='';

while numero>0
    resto=rem(numero,base);
    carattere=convertiincarattere(resto);
    <aggiungi il carattere in testa alla
    stringa>
    <sostituisci a numero il quoziente>
end
```

numeri naturali

35

## IMPLEMENTARE GLI ALGORITMI

```
function stringa=numero2stringa(numero,base)

stringa='';

while numero>0
    resto=rem(numero,base);
    carattere=convertiincarattere(resto);
    stringa=[carattere stringa];
    <sostituisci a numero il quoziente>
end
```

numeri naturali

36

## IMPLEMENTARE GLI ALGORITMI

```
function stringa=numero2stringa(numero,base)

stringa='';

while numero>0
    resto=rem(numero,base);
    carattere=convertiincarattere(resto);
    stringa=[carattere stringa];
    numero=floor(numero/base);
end
```

numeri naturali

37

## IMPLEMENTARE GLI ALGORITMI

```
function stringa=numero2stringa(numero,base)

stringa='';

while numero>0
    resto=rem(numero,base);
    carattere=convertiincarattere(resto);
    stringa=[carattere stringa];
    numero=floor(numero/base);
end
```

occorre scrivere  
una funzione  
convertiincarattere

numeri naturali

38

## IMPLEMENTARE GLI ALGORITMI

```
function carattere=convertiincarattere(resto)
if resto>=0 & resto <=9
    carattere='0'+resto;
end
```

La somma `resto+'0'`  
produce proprio il carattere ASCII  
corrispondente al valore resto

numeri naturali

39