



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

Programmazione Cl.B

Andrea Piroddi

Dipartimento di Informatica, Scienza e Ingegneria

Elaborato 5 - *base_conversion_rc*

Conversione da base 10 a base b.

Si procede nel modo seguente:

dividere il numero da convertire per la base b fino a quando l'ultimo quoziente è minore della base stessa (b), dopodiché *il numero convertito si ottiene* prendendo l'ultimo quoziente e tutti i resti delle divisioni, procedendo dall'ultimo resto al primo e scrivendoli da sinistra verso destra.

Esempio: Convertire il numero 12 da Base 10 a Base 2

$$12 : 2 = 6 \text{ con resto}=0$$

$$6 : 2 = 3 \text{ con resto } =0$$

$$3 : 2 = 1 \text{ con resto } =1$$

$$1 : 2 = 0 \text{ con resto } =1$$

quindi: $(1100)_2$



Elaborato 5 - *base_conversion_rc*

La gestione dei caratteri in C

In C i caratteri sono gestiti mediante variabili di tipo char (interi a 8 bit)

Ad ogni carattere corrisponde un codice numerico ASCII

(American Standard Code for Information Interchange)

Per le basi maggiori di 10, le cifre da 10 a 15 devono essere rappresentate utilizzando le lettere maiuscole dell'alfabeto secondo la seguente codifica:

10 = A, 11 = B, 12 = C,
13 = D, 14 = E, 15 = F

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 Space		64	40	100	@ @		96	60	140	` `	
1	1	001	SOH (start of heading)	33	21	041	! !		65	41	101	A A		97	61	141	a a	
2	2	002	STX (start of text)	34	22	042	" "		66	42	102	B B		98	62	142	b b	
3	3	003	ETX (end of text)	35	23	043	# #		67	43	103	C C		99	63	143	c c	
4	4	004	EOT (end of transmission)	36	24	044	$ \$		68	44	104	D D		100	64	144	d d	
5	5	005	ENQ (enquiry)	37	25	045	% %		69	45	105	E E		101	65	145	e e	
6	6	006	ACK (acknowledge)	38	26	046	& &		70	46	106	F F		102	66	146	f f	
7	7	007	BEL (bell)	39	27	047	' '		71	47	107	G G		103	67	147	g g	
8	8	010	BS (backspace)	40	28	050	((72	48	110	H H		104	68	150	h h	
9	9	011	TAB (horizontal tab)	41	29	051))		73	49	111	I I		105	69	151	i i	
10	A	012	LF (NL line feed, new line)	42	2A	052	* *		74	4A	112	J J		106	6A	152	j j	
11	B	013	VT (vertical tab)	43	2B	053	+ +		75	4B	113	K K		107	6B	153	k k	
12	C	014	FF (NP form feed, new page)	44	2C	054	, ,		76	4C	114	L L		108	6C	154	l l	
13	D	015	CR (carriage return)	45	2D	055	- -		77	4D	115	M M		109	6D	155	m m	
14	E	016	SO (shift out)	46	2E	056	. .		78	4E	116	N N		110	6E	156	n n	
15	F	017	SI (shift in)	47	2F	057	/ /		79	4F	117	O O		111	6F	157	o o	
16	10	020	DLE (data link escape)	48	30	060	0 0		80	50	120	P P		112	70	160	p p	
17	11	021	DC1 (device control 1)	49	31	061	1 1		81	51	121	Q Q		113	71	161	q q	
18	12	022	DC2 (device control 2)	50	32	062	2 2		82	52	122	R R		114	72	162	r r	
19	13	023	DC3 (device control 3)	51	33	063	3 3		83	53	123	S S		115	73	163	s s	
20	14	024	DC4 (device control 4)	52	34	064	4 4		84	54	124	T T		116	74	164	t t	
21	15	025	NAK (negative acknowledge)	53	35	065	5 5		85	55	125	U U		117	75	165	u u	
22	16	026	SYN (synchronous idle)	54	36	066	6 6		86	56	126	V V		118	76	166	v v	
23	17	027	ETB (end of trans. block)	55	37	067	7 7		87	57	127	W W		119	77	167	w w	



Elaborato 5 - *base_conversion_rc*

Esempio:

Convertire il numero 122 da Base 10 a Base 12

$122 : 12 = 10$ con resto=2

$10 : 12 = 0$ con resto =10

Essendo il resto $\geq 10 \rightarrow \text{printf}(\text{'\%c'}, x+55) \rightarrow A$

Risultato $(A2)_{12}$



Elaborato 5 - *base_conversion_rc*

Esempio:

Convertire il numero 123456 da Base 10 a Base 12

$123456:12 = 10288$ con resto = 0

$10288:12 = 857$ con resto = 4

$857:12 = 71$ con resto = 5

$71:12 = 5$ con resto 11

$5:12 = 0$ con resto 5

Essendo il resto di $71:12 \geq 10 \rightarrow \text{printf}(\text{'\%c'}, 11+55) \rightarrow B$

Risultato $(5B540)_{12}$



Elaborato 5 – *base_conversion_rc*

```
void base_conversion_rc(unsigned int x, unsigned int base) {  
    if(base >= 2 && base <= 16) {  
        if(x != 0) {  
            base_conversion_rc(x/base, base);  
            ...  
        }  
    }  
}
```

*prendiamo l'ultimo
quoziente (**minore della
base stessa**) e tutti i
resti delle divisioni,
procedendo dall'ultimo
resto al primo e li
scriviamo da sinistra
verso destra*



Elaborato 5 - base_conversion_it

- Implementazione ricorsiva: richiede poche righe di codice ed è estremamente efficiente in termini di tempo di calcolo.
- Implementazione iterativa: molto più complessa da sviluppare ed inefficiente in termini di tempo di calcolo (a causa dei vincoli imposti). Lo sforzo implementativo è notevolmente semplificato se si riescono ad isolare due precisi sotto-problemi e si affrontano in modo indipendente l'uno dall'altro. Difficilmente si può migliorare l'efficienza computazionale.
 - * **Sotto-problema 1:** determinare di quante cifre è composto il numero nella conversione in base b .
 - * **Sotto-problema 2:** stampare una singola cifra (indicizzata) della rappresentazione in base b .



Elaborato 5 - base_conversion_it

- Vogliamo convertire un numero intero $n > 0$ dalla rappresentazione in base 10 alla rappresentazione in base b . Di quante cifre sarà composta la rappresentazione in base b di n ?

- Avendo a disposizione k cifre, in base b possiamo rappresentare i numeri positivi

$$0, 1, \dots, b^k - 1$$

- Per poter rappresentare il numero n , dobbiamo trovare un k sufficientemente grande, tale per cui

$$\begin{aligned} b^k - 1 &\geq n && \implies \\ \log_b(b^k) &\geq \log_b(n + 1) && \implies \\ k &\geq \log_b(n + 1) \end{aligned}$$

Quindi qualsiasi $k \geq \log_b(n + 1)$ ci permette di rappresentare in base b il numero n .

- Scegliamo il *più piccolo intero* k tra quelli che soddisfano la disuguaglianza: questo è il numero esatto di cifre della rappresentazione di n in base b .

$$k = \lceil \log_b(n + 1) \rceil$$

- La formula per determinare il numero di cifre nella conversione non funziona per il numero 0, che deve essere gestito come caso speciale.
- Come convertiamo un logaritmo da base a a base b ?

$$\log_b(n) = \log_a(n) \log_a(b)$$



Elaborato 5 - base_conversion_it – Sotto Problema 1

* **Sotto-problema 1:** determinare di quante cifre è composto il numero nella conversione in base b .

- Scegliamo il *più piccolo intero* k tra quelli che soddisfano la disequazione: questo è il numero esatto di cifre della rappresentazione di n in base b .

$$k = \lceil \log_b(n + 1) \rceil$$

- Come convertiamo un logaritmo da base a a base b ?

$$\log_b(n) = \log_a(n) \log_a(b)$$

```
static int num_of_digits(unsigned int x,  
unsigned int base) {  
    ...  
}
```

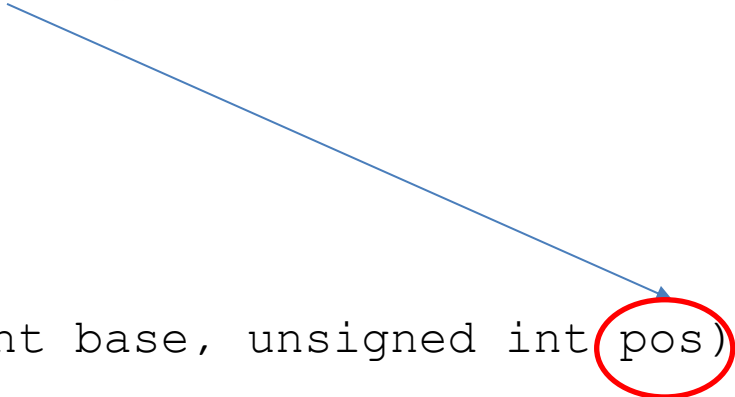
La funzione **ceil()** calcola l'intero più vicino maggiore dell'argomento passato. La funzione è definita nel file di header **<math.h>**.



Elaborato 5 - base_conversion_it - Sotto Problema 2

- * **Sotto-problema 2:** stampare una singola cifra (indicizzata) della rappresentazione in base b .

```
static void print_digit(unsigned int x, unsigned int base, unsigned int pos) {  
    unsigned int q=x,r=0;  
    int i;  
    for(i = 0; i <= pos; i++) {  
        ...  
    }  
    if(...) printf(...);  
    ...  
}
```



Elaborato 5 - base_conversion_it

```
void base_conversion_it(unsigned int x, unsigned int base) {  
    if(base >=2 && base <= 16) {  
        int i;  
        ...  
    }  
}
```

