



Fondamenti di Informatica L-A

Esercitazione del 18/10/07

Ing. Giorgio Villani



Ripasso Teoria

■ Caratteri

- ☐ `char x;`
- ☐ `x = 'a';`
- ☐ `printf("Il carattere e' %c", x);`
- ☐ `putchar(x);`
- ☐ `scanf("%c", &x);`
- ☐ `x = getchar();`



Ripasso Teoria

■ Interi

□ Pre-incremento:

- la variabile viene incrementata prima di essere utilizzata.
- Es: `int i = 1; printf("%d", ++i);` → viene scritto '2'

□ Post-incremento

- La variabile viene incrementata dopo essere stata utilizzata.
- Es: `int i = 1; printf("%d", i++);` → viene scritto '1'

□ NOTA: in entrambi i casi, dopo `printf`, `i` vale 2.



Ripasso Teoria

- `if (espressione) {istruzioni}`
`else {istruzioni}`

```
if (a >= b){  
    printf("a e' maggiore o uguale a b");  
    // altre istruzioni  
}  
else{  
    printf("b e' maggiore di a");  
    // altre istruzioni  
}
```

- È possibile omettere le parentesi `{ }` se `if` o `else` sono seguiti da una sola istruzione.



Ripasso Teoria

■ Operatori relazionali e di uguaglianza

- `a < b`, `a <= b`, ...
- `a == b` /* nota la differenza tra `'='` (assegnamento) e `'=='` (confronto) */
- `a != b`

■ Operatori logici

- `&&` → AND logico
- `||` → OR logico



Ripasso Teoria

■ Cast

- `float f1, f2; int a, b;`
- /* `•` = operatore diverso da `%` */
- `f1 = a • b;` → float
- `a = f1 • f2;` → troncamento
- `a = b • f1;` → troncamento
- `f1 = a • f2;` → float

Esercizio 1

- Scrivere un programma che dato in input un carattere minuscolo (es. 'a'), fornisca in output il carattere maiuscolo (es. 'A').
(utilizzare le diverse alternative)



Tabella ASCII standard

Byte	Cod.	Char	Byte	Cod.	Char	Byte	Cod.	Char	Byte	Cod.	Char
00000000	0	Null	00100000	32	SpC	01000000	64	@	01100000	96	`
00000001	1	Start of heading	00100001	33	!	01000001	65	A	01100001	97	a
00000010	2	Start of text	00100010	34	"	01000010	66	B	01100010	98	b
00000011	3	End of text	00100011	35	#	01000011	67	C	01100011	99	c
00000100	4	End of transmit	00100100	36	\$	01000100	68	D	01100100	100	d
00000101	5	Enquiry	00100101	37	%	01000101	69	E	01100101	101	e
00000110	6	Acknowledge	00100110	38	&	01000110	70	F	01100110	102	f
00000111	7	Audible bell	00100111	39	'	01000111	71	G	01100111	103	g
00001000	8	Backspace	00101000	40	(01001000	72	H	01101000	104	h
00001001	9	Horizontal tab	00101001	41)	01001001	73	I	01101001	105	i
00001010	10	Line feed	00101010	42	*	01001010	74	J	01101010	106	j
00001011	11	Vertical tab	00101011	43	+	01001011	75	K	01101011	107	k
00001100	12	Form Feed	00101100	44	,	01001100	76	L	01101100	108	l
00001101	13	Carriage return	00101101	45	-	01001101	77	M	01101101	109	m
00001110	14	Shift out	00101110	46	.	01001110	78	N	01101110	110	n
00001111	15	Shift in	00101111	47	/	01001111	79	O	01101111	111	o
00010000	16	Data link escape	00110000	48	0	01010000	80	P	01110000	112	p
00010001	17	Device control 1	00110001	49	1	01010001	81	Q	01110001	113	q
00010010	18	Device control 2	00110010	50	2	01010010	82	R	01110010	114	r
00010011	19	Device control 3	00110011	51	3	01010011	83	S	01110011	115	s
00010100	20	Device control 4	00110100	52	4	01010100	84	T	01110100	116	t
00010101	21	Neg. acknowledge	00110101	53	5	01010101	85	U	01110101	117	u
00010110	22	Synchronous idle	00110110	54	6	01010110	86	V	01110110	118	v
00010111	23	End trans. block	00110111	55	7	01010111	87	W	01110111	119	w
00011000	24	Cancel	00111000	56	8	01011000	88	X	01111000	120	x
00011001	25	End of medium	00111001	57	9	01011001	89	Y	01111001	121	y
00011010	26	Substitution	00111010	58	:	01011010	90	Z	01111010	122	z
00011011	27	Escape	00111011	59	;	01011011	91	[01111011	123	{
00011100	28	File separator	00111100	60	<	01011100	92	\	01111100	124	
00011101	29	Group separator	00111101	61	=	01011101	93]	01111101	125	}
00011110	30	Record Separator	00111110	62	>	01011110	94	^	01111110	126	~
00011111	31	Unit separator	00111111	63	?	01011111	95	_	01111111	127	Del

$$97-65=32$$

$$a - A = 32$$



Soluzione

```
/*
  Programma che legge un carattere minuscolo da input e stampa in
  output il carattere maiuscolo
  INPUT : char x
  OUTPUT: char X
*/

#include <stdio.h>

void main(){

    char x;          // dichiarazione variabili

    printf("Inserire un carattere minuscolo\n");
    scanf("%c", &x);    // lettura valore in input

    printf("Il carattere maiuscolo e': %c\n", x-32);

}
```



Soluzione alternativa

```
/*
  Programma che legge un carattere minuscolo da input e stampa in
  output il carattere maiuscolo
  INPUT : char x
  OUTPUT: char X
*/

#include <stdio.h>

void main(){

    char x;          // dichiarazione variabili

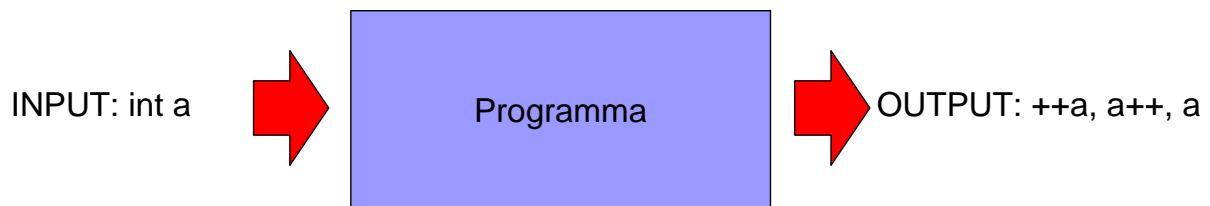
    printf("Inserire un carattere minuscolo\n");
    x = getchar();

    printf("Il carattere maiuscolo e': ");
    putchar(x-32);

}
```

Esercizio 2

- Scrivere un programma che legge in input un intero (x) e scrive in output il valore con pre-incremento (++x), post-incremento (x++), e il valore finale (x).



Soluzione

```
/* Programma che legge un intero da input e fornisce il valore con
pre-incremento, post-incremento, e il valore finale

INPUT : intero a
OUTPUT: ++a, a++, a */

#include <stdio.h>

void main(){

    int a;          // dichiarazione variabili

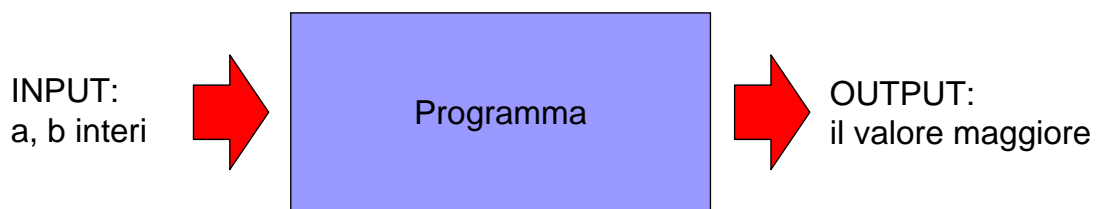
    printf("Inserire valore intero\n");
    scanf("%d", &a);          // lettura valore in input          → Es. a = 1

    printf("Pre-incremento: %d\n", ++a); //incrmento prima di printf          → 2
    printf("Post-incremento: %d\n", a++); //incrmento dopo printf          → 2
    printf("Valore finale: %d\n", a);    //visualizzo incremento dopo printf → 3

}
```

Esercizio 3

- Scrivere un programma che legga in input due numeri interi e scriva in output il valore maggiore. Se i valori inseriti sono uguali il programma scrive “valori uguali”.



Soluzione

```
/*
INPUT : due interi a, b
OUTPUT: il maggiore tra a e b oppure dice che a e b sono uguali
*/

#include <stdio.h>

void main(){

    int a,b;           // dichiarazione variabili

    printf("Inserire il primo intero\n");
    scanf("%d", &a);    // lettura valore in input
    printf("Inserire il secondo intero\n");
    scanf("%d", &b);    // lettura valore in input

    if(a>b){
        printf("Il primo numero e' maggiore del secondo\n");
    }
    else { if(a==b){           //ATTENZIONE a non scrivere if(a=b)
        printf("I numeri inseriti sono uguali\n");
    }
    else{
        printf("Il secondo numero e' maggiore del primo\n");
    }
}
```

Esercizio 4

- Scrivere un programma che legge due interi a, b e due float f1, f2 da input e fornisce in output (float f, int i) $f=a+b$, $i=f1+f2$, $f=a+f1$, $i=a+f$. Commentare i risultati.



Soluzione

```
/* Programma che legge due interi e due float da input e fornisce in
   output esempi di cast */
```

```
#include <stdio.h>
```

```
void main(){
```

```
    int a, b, i;           // dichiarazione variabili
    float f1, f2, f;
```

```
    printf("Inserire valore intero\n");
    scanf("%d", &a);        //a = 1
    printf("Inserire valore intero\n");
    scanf("%d", &b);        //b = 2
    printf("Inserire float\n");
    scanf("%f", &f1);       //f1 = 3.6
    printf("Inserire float\n");
    scanf("%f", &f2);       //f2 = 4.9
```

```
// segue...
```




```
f = a+b; //sommo due interi -> float
printf("Somma dei due interi -> float: %f\n", f); // f = 3.000000

i = f1+f2; //sommo due float -> int
printf("Somma dei due float -> intero: %d\n", i); // i = 8
//NOTA troncamento

f = a+f1; //sommo int e float -> int
printf("Somma del primo intero e del primo float -> float: %f\n", f);
// f = 4.600000

i = a+f1; //sommo int e float -> float
printf("Somma del primo intero e del primo float -> intero: %d\n",
i);
// i = 4
//NOTA troncamento
```