

Note su scanf()

Problemi con la funzione scanf() in Windows

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      unsigned char a, b;
7
8      printf ("\n INSERISCI UN NUMERO DECIMALE INTERO (0-255) -a-: ");
9      fflush (stdin); scanf ("%d",&a);
10
11     printf ("\n<DEBUG> VALORE di a: %d\n", a);
12
13     printf ("\n INSERISCI UN NUMERO DECIMALE INTERO (0-255) -b-: ");
14     fflush (stdin); scanf ("%d",&b);
15
16     printf ("\n<DEBUG> VALORE di a: %d\n", a);
17     printf ("\n<DEBUG> VALORE di b: %d\n", b);
18
19     return 0;
20 }
21
```

INSERISCI UN NUMERO DECIMALE INTERO (0-255) -a-: 36

<DEBUG> VALORE di a: 36

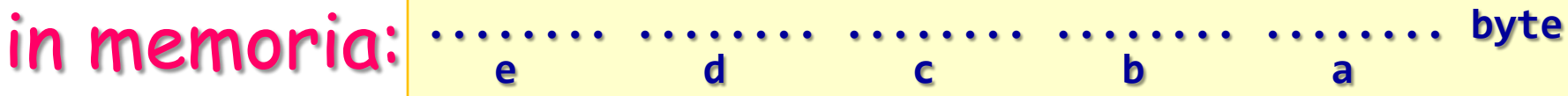
INSERISCI UN NUMERO DECIMALE INTERO (0-255) -b-: 21

<DEBUG> VALORE di a: 0

<DEBUG> VALORE di b: 21

la lettura di b ha modificato il valore di a

Perché?



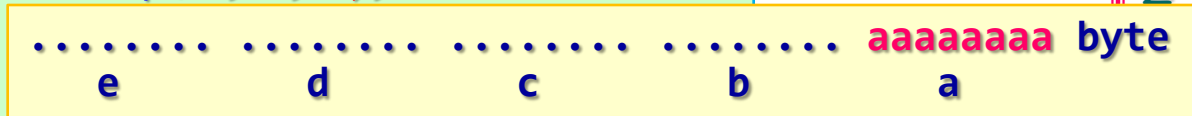
```
#include <stdio.h>
#include <stdlib.h>
int main()
{
  unsigned char a=0, b=0, c=0, d=0, e=0;

  printf ("\n<DEBUG> VALORE a = %d\t&a = %p\n", a,&a);
  printf ("<DEBUG> VALORE b = %d\t&b = %p\n", b,&b);
  printf ("<DEBUG> VALORE c = %d\t&c = %p\n", c,&c);
  printf ("<DEBUG> VALORE d = %d\t&d = %p\n", d,&d);
  printf ("<DEBUG> VALORE e = %d\t&e = %p\n", e,&e);
```

<DEBUG> VALORE a = 0 &a = 0028FEFB
<DEBUG> VALORE b = 0 &b = 0028FEFA
<DEBUG> VALORE c = 0 &c = 0028FEF9
<DEBUG> VALORE d = 0 &d = 0028FEF8
<DEBUG> VALORE e = 0 &e = 0028FEF7

```
    /** 1) LEGGE con %d a **/
    printf ("\n INSERISCI UN NUMERO DECIMALE INTERO (0-255) -a-: ");
    fflush (stdin); scanf ("%d",&a);
    printf ("\n<DEBUG> VALORE a = %d\t&a = %p\n", a,&a);
```

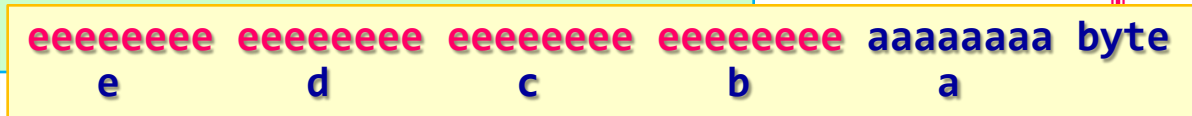
<DEBUG> VALORE a = 36



```
    /** 2) LEGGE con %d e (in memoria dista da a 5 byte) **/
    printf ("\n INSERISCI UN NUMERO DECIMALE INTERO (0-255) -e-: ");
    fflush (stdin); scanf ("%d",&e);
    printf ("\n<DEBUG> VALORE a = %d\t&a = %p\n", a,&a);
    printf ("<DEBUG> VALORE e = %d\t&e = %p\n", e,&e);
```

<DEBUG> VALORE a = 36
<DEBUG> VALORE e = 21

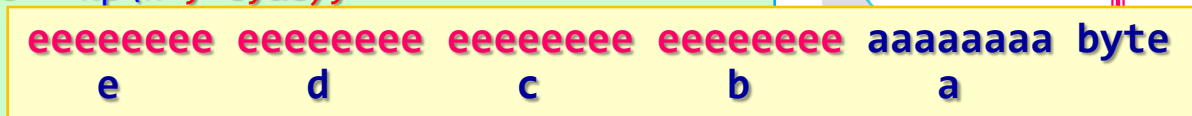
...
a non è stato modificato



```
...
/** 2) LEGGE con %d e (in memoria dista da a 5 byte) */
printf ("\n INSERISCI UN NUMERO DECIMALE INTERO (0-255) -e-: ");
fflush (stdin); scanf("%d",&e);
printf ("\n<DEBUG> VALORE a = %d\t&a = %p\n", a,&a);
printf ("<DEBUG> VALORE e = %d\t&e = %p\n", e,&e);
```

<DEBUG> VALORE a = 36
<DEBUG> VALORE e = 21

a non è stato modificato



```
/** 3) LEGGE con %d b (in memoria dista da a 1 byte) */
printf ("\n INSERISCI UN NUMERO DECIMALE INTERO (0-255) -b-: ");
fflush (stdin); scanf("%d",&b);
printf ("\n<DEBUG> VALORE a = %d\t&a = %p\n", a,&a);
printf ("<DEBUG> VALORE b = %d\t&b = %p\n", b,&b);
printf ("<DEBUG> VALORE e = %d\t&e = %p\n", e,&e);
```

<DEBUG> VALORE a = 0
<DEBUG> VALORE b = 36
<DEBUG> VALORE e = 21

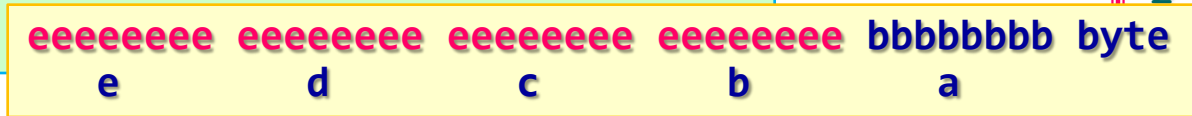
a è stato modificato



```
/** 4) LEGGE con %d di nuovo e (in memoria dista da b 4 byte) */
printf ("\n INSERISCI UN NUMERO DECIMALE INTERO (0-255) -e-: ");
fflush (stdin); scanf("%d",&e);
printf ("\n<DEBUG> VALORE a = %d\t&a = %p\n", a,&a);
printf ("<DEBUG> VALORE b = %d\t&b = %p\n", b,&b);
printf ("<DEBUG> VALORE e = %d\t&e = %p\n", e,&e);
```

<DEBUG> VALORE a = 0
<DEBUG> VALORE b = 0
<DEBUG> VALORE e = 15

b è stato modificato



%d in scanf(): legge sempre 4 byte (int)

in Windows/Linux

%d in scanf(): legge sempre 4 byte (int)

%hd in scanf(): legge sempre 2 byte (short)

... provare

in Linux

%hhd in scanf(): legge sempre 1 byte (char)

(non funziona in Windows con MinGW)

compilazione in Linux

```
$ gcc -o main.exe main.c
```

```
unsigned char a=0, b=0, c=0, d=0, e=0;
```

```
fflush (stdin); scanf("%d",&a);
```

```
main.c: In function 'main':  
main.c:51:27: warning: format '%d' expects argument of type 'int *',  
but argument 2 has type 'unsigned char *' [-Wformat=]  
    fflush (stdin); scanf("%d",&a);  
                        ^
```

```
fflush (stdin); scanf("%hd",&b);
```

```
main.c:76:27: warning: format '%hd' expects argument of type 'short int *',  
but argument 2 has type 'unsigned char *' [-Wformat=]  
    fflush (stdin); scanf("%hd",&b);  
                        ^
```

```
fflush (stdin); scanf("%hhd",&c);
```

OK!