

Informatica 1

Corso di Laurea Triennale in Matematica

Gianluca Rossi

`gianluca.rossi@uniroma2.it`

Dipartimento di Matematica
Università di Roma "Tor Vergata"

7: Introduzione ai Puntatori



Array, valori, indirizzi...

```
int n = 10;  
int a[4] = { 1, 2, 3, 4 };
```

- Il valore della variabile che individua un array (p.e. *a*) è l'indirizzo di memoria (*puntatore*) che contiene il valore del primo elemento del vettore (*a[0]*)

$n = 10$	9A1 9A2
...	...
$a[0] = 1$	F1A F1B
$a[1] = 2$	F1C F1D
$a[2] = 3$	F1E F1F
$a[3] = 4$	F20 F21
...	...
$a = F1A$	F6A F6B F6C



Esempio

```
#include<stdio.h>
main(){
    int a[4] = {1, 2, 3, 4};
    printf("il vettore a e' memorizzato in %p\n", a);
}
```



The screenshot shows a terminal window titled "Terminale — bash — 85x15". The user is logged in as gianluca@iggymac.local. The terminal shows the following sequence of commands and output:

```
gianluca@iggymac.local[03/31 21:09:28 - 07]
$>
gianluca@iggymac.local[03/31 21:09:28 - 07]
$>
gianluca@iggymac.local[03/31 21:09:28 - 07]
$>
gianluca@iggymac.local[03/31 21:09:28 - 07]
$>
gianluca@iggymac.local[03/31 21:09:28 - 07]
$> gcc 01-init.c
gianluca@iggymac.local[03/31 21:09:31 - 07]
$> ./a.out
il vettore a e' memorizzato in posizione 0xbffff9e0
gianluca@iggymac.local[03/31 21:09:33 - 07]
$> |
```

Operatore &

Data una variabile x

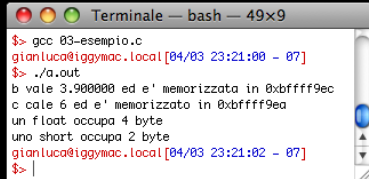
$\&x;$

indica l'indirizzo in cui è memorizzata la variabile x ;

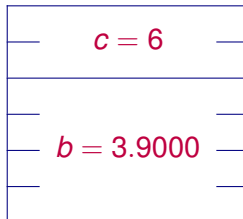


Esempio

```
#include<stdio.h>
main(){
    float b = 3.9;
    short c = 6;
    printf("b vale %f ed e' memorizzata in %p\n", b, &b);
    printf("c vale %d ed e' memorizzato in %p\n", c, &c);
    printf("un float occupa %d byte\n", sizeof(float));
    printf("uno short occupa %d byte\n", sizeof(short));
}
```



```
Terminale — bash — 49x9
$> gcc 03-esempio.c
gianluca@iggyamac.local[04/03 23:21:00 - 07]
$> ./a.out
b vale 3.900000 ed e' memorizzata in 0xbfff9ec
c vale 6 ed e' memorizzato in 0xbfff9ea
un float occupa 4 byte
uno short occupa 2 byte
gianluca@iggyamac.local[04/03 23:21:02 - 07]
$> |
```



bfff9ea = &c
bfff9eb
bfff9ec = &b
bfff9ed
bfff9ee
bfff9ef



Tipo *

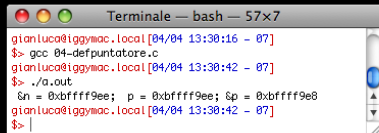
```
tipo *p;
```

definisce la variabile *p* come un puntatore a *tipo*. Ovvero, *p* conterrà indirizzi di memoria che si riferiscono a contenuti di tipo *tipo*.



Esempio

```
#include<stdio.h>
main(){
    short n, *p;
    n = 11;
    p = &n;
    printf("  &n = %p;   p = %p;  &p = %p\n", &n, p, &p);
}
```



```
Terminale — bash — 57x7
gianluca@iggyamac.local[04/04 13:30:16 - 07]
$> gcc 04-defpuntatore.c
gianluca@iggyamac.local[04/04 13:30:42 - 07]
$> ./a.out
&n = 0xbffff9ee;  p = 0xbffff9ee;  &p = 0xbffff9e8
$>
```

0xbffff9ee

...

11

bffff9e8 = &p
bffff9e9
bffff9ea
bffff9eb

...
bffff9ee = &n
bffff9ef



Definizione di puntatori

Operatore *

Data una variabile *p* di tipo puntatore a *tipo*

tipo *p;

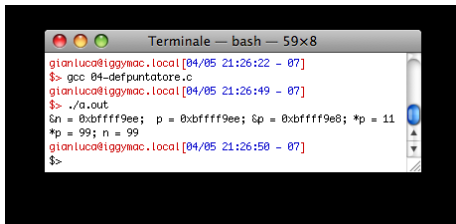
l'operatore * si riferisce al **valore** contenuto nell'indirizzo contenuto in *p*.

*p;

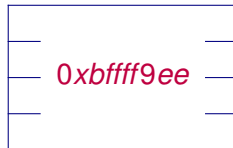


Esempio

```
#include<stdio.h>
main(){
    short n, *p;
    n = 11;
    p = &n;
    printf("&n = %p;  p = %p;  &p = %p;  *p = %d\n", &n, p, &p, *p);
    *p = 99;
    printf("*p = %d;  n = %d\n", *p, n);
}
```



```
Terminale — bash — 59x8
gianluca@iggyamac.local[04/05 21:26:22 - 07]
$> gcc 04-defpuntatore.c
gianluca@iggyamac.local[04/05 21:26:49 - 07]
$> ./a.out
&n = 0xbffff9ee;  p = 0xbffff9ee;  &p = 0xbffff9e8;  *p = 11
*p = 99;  n = 99
gianluca@iggyamac.local[04/05 21:26:50 - 07]
$>
```



`bffff9e8 = &p`
`bffff9e9`
`bffff9ea`
`bffff9eb`

...

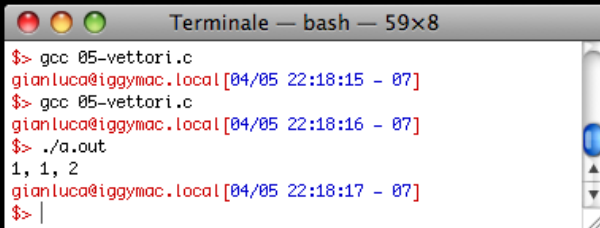


...
`bffff9ee = &n`
`bffff9ef`



Puntatori e vettori

```
#include<stdio.h>
main(){
    short v[] = {1,2,3};
    short *p1, *p2, *p3;
    p1 = v; /*l'indirizzo del primo elemento di v*/
    p2 = &(v[0]);
    p3 = &(v[1]);
    printf("%d, %d, %d\n", *p1, *p2, *p3);
}
```



```
Terminale — bash — 59x8
$> gcc 05-vettori.c
gianluca@iggyMac.local[04/05 22:18:15 - 07]
$> gcc 05-vettori.c
gianluca@iggyMac.local[04/05 22:18:16 - 07]
$> ./a.out
1, 1, 2
gianluca@iggyMac.local[04/05 22:18:17 - 07]
$> |
```

