

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



- Il puntatore

tipo *p;

è un indirizzo, quindi un valore numerico;

- In quanto valore numerico può essere soggetto ad operazioni aritmetiche (addizione e sottrazione);
- Il risultato numerico di una operazione aritmetica su p dipende da **sizeof(tipo)**
- In particolare, il valore numerico di $p + c$ è $p + c \cdot \text{sizeof}(\text{tipo})$;



Aritmetica dei puntatori

```
#include <stdio.h>
main(){
    short v[] = {1,2,3};
    short *p;
    p = v;
    printf("%p %p %p\n", &(v[0]), &(v[1]), &(v[2]));
    printf("%p %p %p\n", p, p+1, p+2);
    printf("%d %d %d\n", *p, *(p+1), *(p+2));
}
```

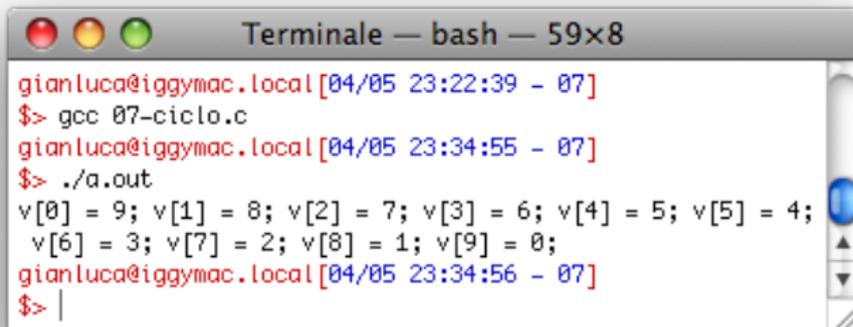


```
Terminale — bash — 59x8
$> gcc 06-aritmetica.c
gianluca@iggymac.local [04/05 23:12:23 - 07]
$> ./a.out
0xbffff9e6 0xbffff9e8 0xbffff9ea
0xbffff9e6 0xbffff9e8 0xbffff9ea
1 2 3
gianluca@iggymac.local [04/05 23:12:25 - 07]
$>
```



Aritmetica dei puntatori

```
#include<stdio.h>
main(){
    int v[] = {9, 8, 7, 6, 5, 4, 3, 2, 1, 0};
    int *p, i;
    p = v;
    for(i = 0; i < sizeof(v)/sizeof(int); i++)
        printf("v[%d] = %d; ", i, *(p+i));
    printf("\n");
}
```



```
Terminale — bash — 59x8
gianluca@iggyMac.local [04/05 23:22:39 - 07]
$> gcc 07-ciclo.c
gianluca@iggyMac.local [04/05 23:34:55 - 07]
$> ./a.out
v[0] = 9; v[1] = 8; v[2] = 7; v[3] = 6; v[4] = 5; v[5] = 4;
v[6] = 3; v[7] = 2; v[8] = 1; v[9] = 0;
gianluca@iggyMac.local [04/05 23:34:56 - 07]
$> |
```



```
01: #include<stdlib.h>
02: #include<time.h>
03: #define N 1000
04: main(){
05:     int v[N];
06:     int i;
07:     int seed = time(0);
08:     srand(seed);
09:     for(i = 0; i < N; i++)
10:         v[i] = rand() % 10;
11: }
```

Numero di operazioni nel ciclo for

- Test ($i < N$);
- Incremento ($i++$);
- Somma + Moltiplicazione ($v[i]$, indirizzo in $v + i \text{ sizeof (int)}$);
- Chiamata a funzione + Modulo + Assegnazione ($= \text{rand}() \% 10$);

Numero totale di operazioni nel for

$$7 \cdot N$$

```
01: #include<stdlib.h>
02: #include<time.h>
03: #define N 1000
04: main(){
05:     int v[N];
06:     int *p,i;
07:     int seed = time(0);
08:     srand(seed);
09:     for(i = 0, p = v; i < N; i++)
10:         *(p+i) = rand() % 10;
11: }
```

Numero di operazioni nel ciclo for

- Test ($i < N$);
- Incremento ($i++$);
- Somma + Moltiplicazione ($p+i$, indirizzo in $p + i \text{ sizeof (int)}$);
- Chiamata a funzione + Modulo + Assegnazione ($= \text{rand}() \% 10$);

Numero totale di operazioni nel for

$$7 \cdot N$$

```
01: #include <stdlib.h>
02: #include <time.h>
03: #define N 1000
04: main(){
05:     int v[N];
06:     int *p, i;
07:     int seed = time(0);
08:     srand(seed);
09:     for(p = v; p < v + N; p++)
10:         *p = rand() % 10;
11: }
```

Numero di operazioni nel ciclo for

- Test + Somma ($p < v + N$);
- Incremento ($p++$);
- Chiamata a funzione + Modulo + Assegnazione (= **rand()** % 10);

Numero totale di operazioni nel for

$$6 \cdot N$$



```
01: #include<stdlib.h>
02: #include<time.h>
03: #define N 1000
04: main(){
05:     int v[N];
06:     int *p,i, *lp;
07:     int seed = time(0);
08:     srand(seed);
09:     lp = v + N;
10:     for(p = v; p < lp; p++)
11:         *p = rand() % 10;
12: }
```

Numero di operazioni nel ciclo for

- Test ($p < lp$);
- Incremento ($p++$);
- Chiamata a funzione + Modulo + Assegnazione ($= \text{rand}() \% 10$);

Numero totale di operazioni nel for

$$5 \cdot N$$



```
int v[9] = {0}, *p, u[2];  
p = v; /* legale */  
u = p; /* illegale */
```

Non si può assegnare un valore diverso (puntatore) ad una variabile dichiarata come vettore. Infatti, il valore assegnato al momento della definizione risulta un valore costante non modificabile.

