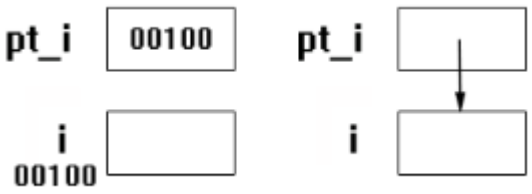
PUNTATORI

I puntatori sono uno speciale tipo di dato.

Essi puntano alla singola cella di memoria; contengono dunque l'indirizzo della cella stessa.



Per poter utilizzare le variabili di tipo puntatore è necessario specificare qual è il tipo di dato a cui punterà. Per farlo è sufficiente aggiungere un * al tipo.

int* p;

ad esempio, indica che p è puntatore ad intero; conterrà quindi l'indirizzo di una variabile di tipo int.

N.B. un puntatore deve puntare a una variabile già esistente. La semplice dichiarazione del puntatore non alloca memoria per la variabile puntata.

OPERATORI

Nell'utilizzare i puntatori, sono fondamentali due operatori: & e *.

-& consente di risalire all'indirizzo di una variabile; permettendo di assegnarlo ad un puntatore:

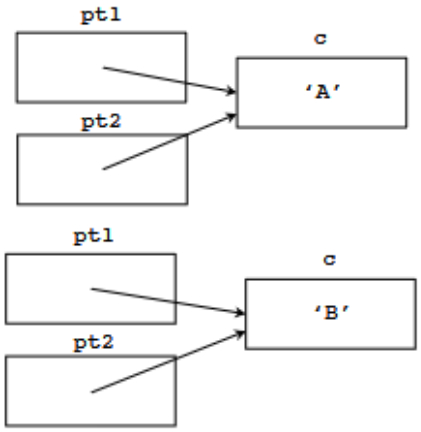
```
int a;       //allochiamo memoria per la variabile a
int* p = &a; //ne mettiamo l'indirizzo in p
```

-* consente di risalire al contenuto della variabile puntata

```
int a;       //allochiamo memoria per la variabile a
int* p = &a; //ne mettiamo l'indirizzo in p
*p = 10;     //*p equivale al contenuto di a, dunque ora a=10
```

CONDIVISIONE DI MEMORIA

Un'area di memoria a cui fanno riferimento due o più puntatori è detta condivisa. In questo caso, le modifiche effettuate tramite un puntatore sono visibili tramite l'altro (e viceversa).



ARITMETICA DEI PUNTATORI

Alle variabili di tipo puntatore è possibile sommare o sottrarre valori interi. Così facendo scorriamo gli indirizzi coerentemente al tipo del puntatore. Ovviamente sommando 2 a un puntatore a char ci sposteremo di due dimensioni di char ecc… Tale operazione è automatica:

```
int n;
int* p = &n;
int* p2 = p+2;  //ci spostiamo di 2*sizeof(int) indirizzi

char c;
char* q = &c;
char* q2 = q+2;  //ci spostiamo di 2*sizeof(char) indirizzi
```