CLASS RETTANGOLO {}

CLASS QUADRATO EXTENDS RETTANGOLO {}

PUBLIC CLASS PROVA3
  PUBLIC STATIC VOID {};
    QUADRATO q = (QUADRATO) NEW RETTANGOLO(); }
                ↳ dà errore in esecuzione

Rettangolo r = new quadrato()

QUADRATO q = (quadrato) r; → in compilazione e in esecuz. non ho errori

Rettangolo r = new quadrato() → VA BENE!

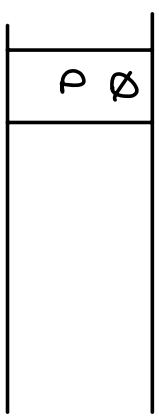
— ○ — ○ — ○ — ○

↳ questo puntatore non punta a niente, ha dentro zero

```
int *p = NULL;
printf ("%d", *p);
```

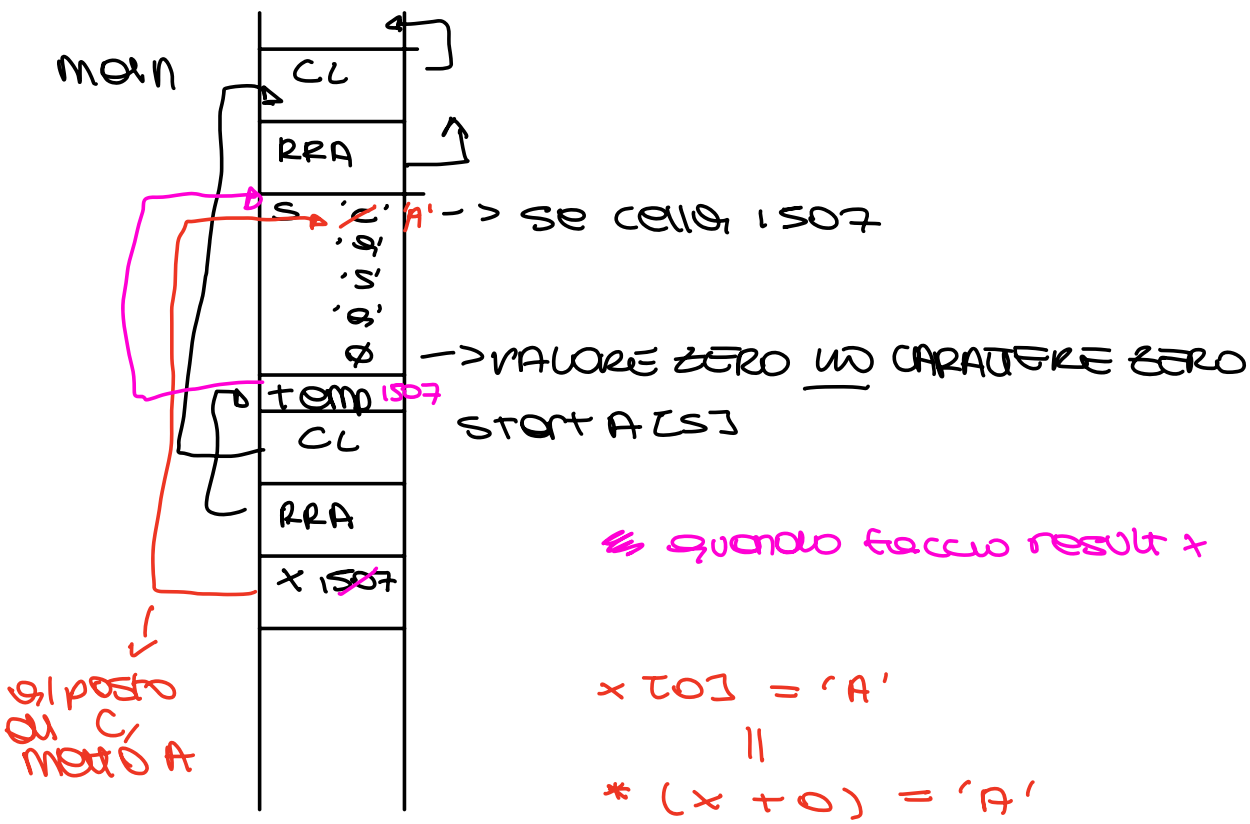
↓ cerca di accedere alla memoria zero

↓ indirizzo zero però non valido



```
char * starta (char * x){
    x[0] = 'A';
    return x;}

int main (void){
  char s[5] = 'casa';
  printf(starta(s));}
```



start A[s]

≠ quando faccio result x

x[0] = 'A'

||

*(x+0) = 'A'

||

*x = 'A'

Alla fine stampa "Aasa"

```
char * starta (char * x){
    x[10] = 'A';   <- VARIA RISPETTO A PRIMA
    return x; }

int main (void){
  char s[5] = 'casa';
  printf(starta(s)); }
```

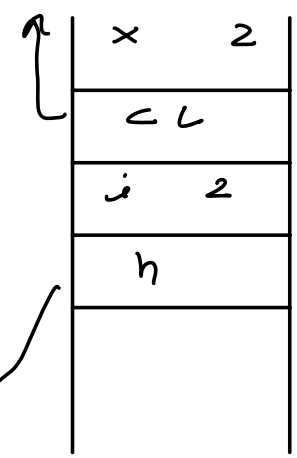
`x[10] = 'A'` -> scrive A nella memoria però non al posto di 'c'

=> È sbagliato ma non dà errore
<> non restituisce il risultato che vorrei

— o — o — o — o — o —

```
void foo (int i){
  int * h = malloc(i * sizeof(int));
  h[i - 1] = 100;
return;
```

NO RRA PERCHÉ VOID



puntatore a interi allocato nello heap con la malloc

```
main:  int x = 2;
       foo(x);
```

h[i-1] = 100 ⇒ h[2-1] = 100

⇓

h[1] = 100

→ da vedere quando ho la malloc!

genera un memory leak → quando esco da foo sento h non posso più accedere ⇐ non dealloco la malloc

— o — o — o — o —

```
int * foo (int * x){
    int h = * x;
    int * y =& h;
    return y; }

main:  int i =5;
       int * p = foo(&i);
```