ARRAY & PUNTATORI (ARITMETICA DEI PUNTATORI)

```
int x[100];

typedef struct {
    int g, m, a
} data;

data d[100];

typedef char string[500];

string x[10];

x[0]
x[1]
⋮
x[9]
```

```
int    *x;
data   *p;
  :
int    **y;
```

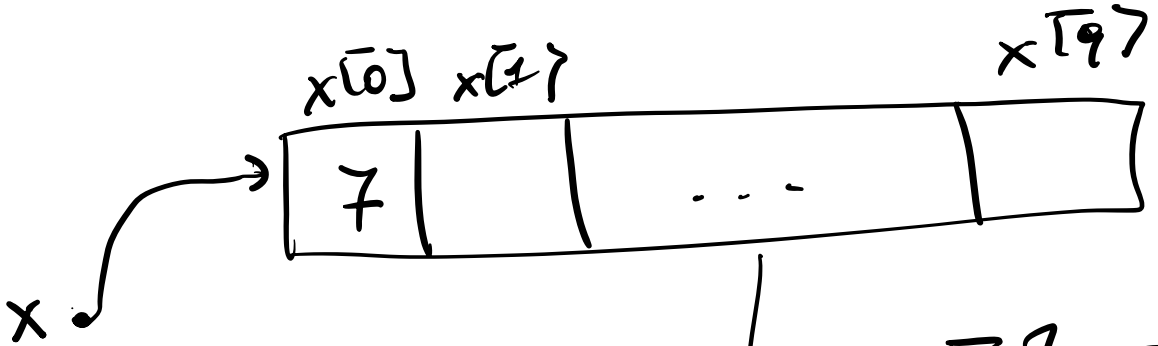
```
x = NULL;
int  i;
x = &i;

*x = 15;
```

```
x = (int *) malloc(sizeof(int))
        void *
  :
free (x);
```

int x[10];

x → puntatore a un intero



x[0]=7;
x[1]=16;

*x = 7;
*(x+1)=16;

ARITMETICA DEI PUNTATORI

puntatore + intero

= p + x * byte necessari per un oggetto a cui p punta

typedef char string[500];

string x[10];

*(x+7)

((x+7))='a';

x[7]

x[7][0]='a';

```
int   f (int  x[]) {
        printf ("%d", x[0]);

}


int   f (int  *x ) {
        printf ("%d", x[0])

}
int   main () {
        int   a [50];
        int
        f(a);

        int  *p;    //a(sizeof(int)
```

→ p = (int *) malloc(100*

p[0] = 5;

p[1] = 12;

...

f(p);

```
void fill_zero (int *x, int n) {
    for (int i=0; i<n; i++)
        x[i]=0;
}

int main () {
    int a[50];
    fill_zero (a, 50);
```

```
int strlen(char *s) {
    for (int i=0;0; i++)
        if (s[i]==0) {
            return i
        }
    }
}
```

'\0'

```
if (!s[i]) {
    return i
}
```