

# tipo degli array e aritmetica dei puntatori

5.3 e 5.4 del testo

che tipo ha un array ?

`char A[100];` // `char *` o `char []`

e che valore ha ?

`cout<< A<< &A[0];` // stampa 2 indirizzi uguali

# A è una costante

provate a compilare `A=A+1; //` da errore

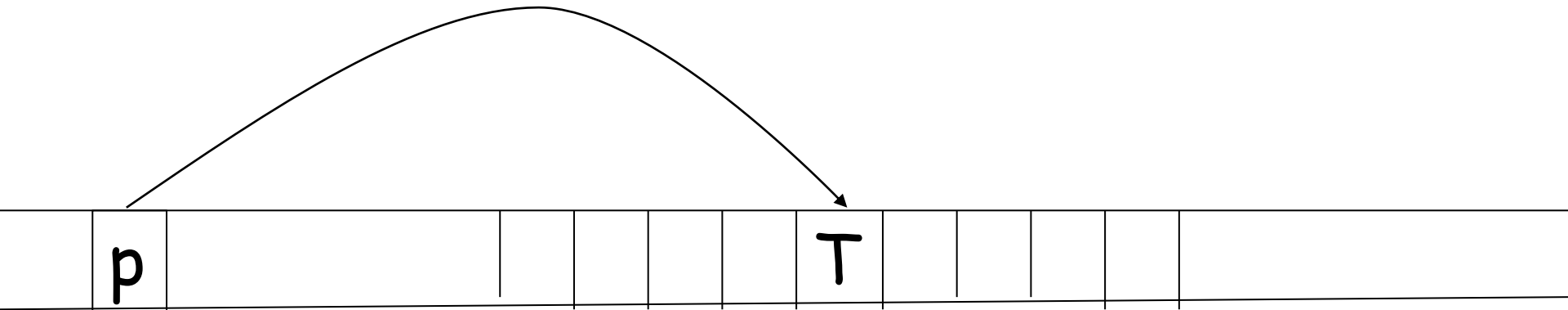
## PERCHE'?

se cambiassi A perderei l'accesso all'array  
e questo **non può essere GIUSTO**

ma  $A+1$  è espressione valida che indica il  
puntatore all'elemento di indice 1  
dell'array

$A+k$  punta all'elemento di indice  $k$

come il C++ vede un puntatore di tipo  $T^*$



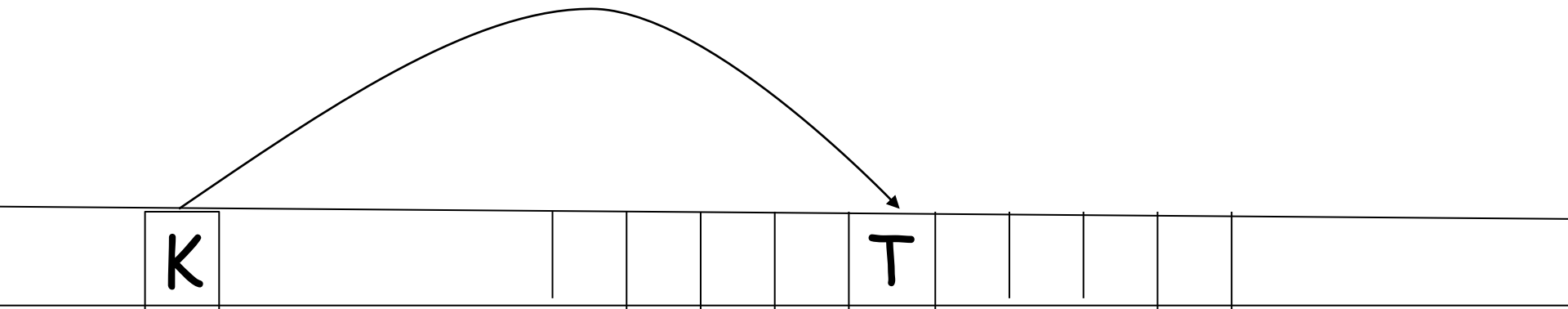
p punta ad un elemento di un array di elementi di tipo T, quindi  $p+n$  sposta il puntatore di n elementi a destra e  $p-n$  lo sposta di n elementi a sinistra

## tipo e dimensione degli oggetti puntati

`int K[5][10];` tipo di K = `int (*) [10]` = `int [] [10]`  
dimensione =  $10 \times 4$

`char K[4][6][8];` tipo = `char (*) [6][8]` =  
`char [] [6][8]` dimensione =  $6 \times 8$

`double K[3][5][7][9];` tipo = `double (*) [5][7][9]`  
dimensione =  $5 \times 7 \times 9 \times 8$



e il valore degli array K?

```
cout<<K ;
```

produce la stampa dell'indirizzo (L-valore)  
del primo elemento dell'array

# Tipo e valore di array

double F[3][5][7][9]; tipo = double (\*)[5][7][9]  
e valore = &F[0][0][0][0]

tipo di \*F e suo R-valore ?  
double (\*)[7][9] e &F[0][0][0][0], \*F=F[0]

tipo di \*\*F e suo R-valore ?  
double (\*)[9] e &F[0][0][0][0], \*\*F=F[0][0]

tipo di \*\*\*F e suo R-valore ?  
double \* e &F[0][0][0][0], \*\*\*F=F[0][0][0]

tipo di \*\*\*\*F e suo R-valore ?

double e valore di F[0][0][0][0]

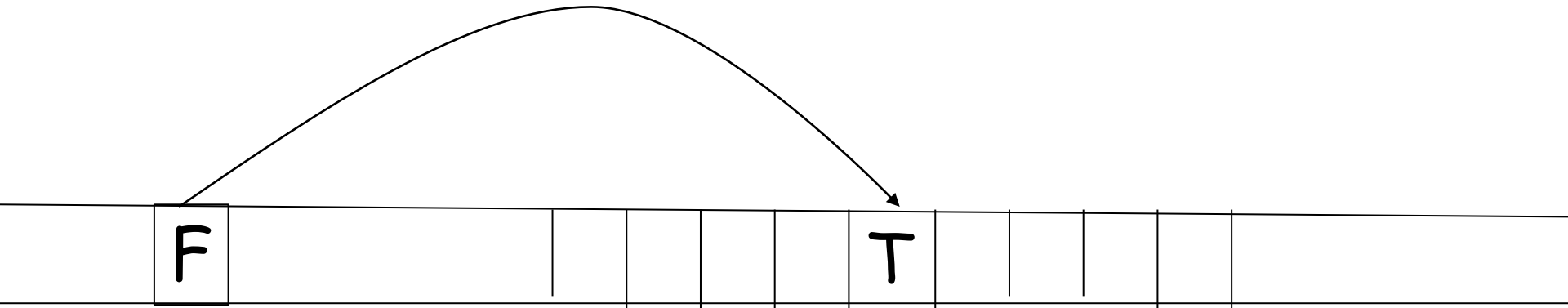
F, \*F, \*\*F, \*\*\*F sono tutti puntatori, ad oggetti di dimensioni diverse e quindi a loro si applica l'aritmetica dei puntatori **con effetti diversi**

che valore ha F+2 e (\*F)+2 e (\*\*F)+2 e (\*\*\*F)+2 ?

**basta sapere il tipo di ciascun puntatore**



partiamo da F, ha tipo, double (\*) [5][7][9]  
T ha dimensione =  $(5*7*9)*8 = 315*8$

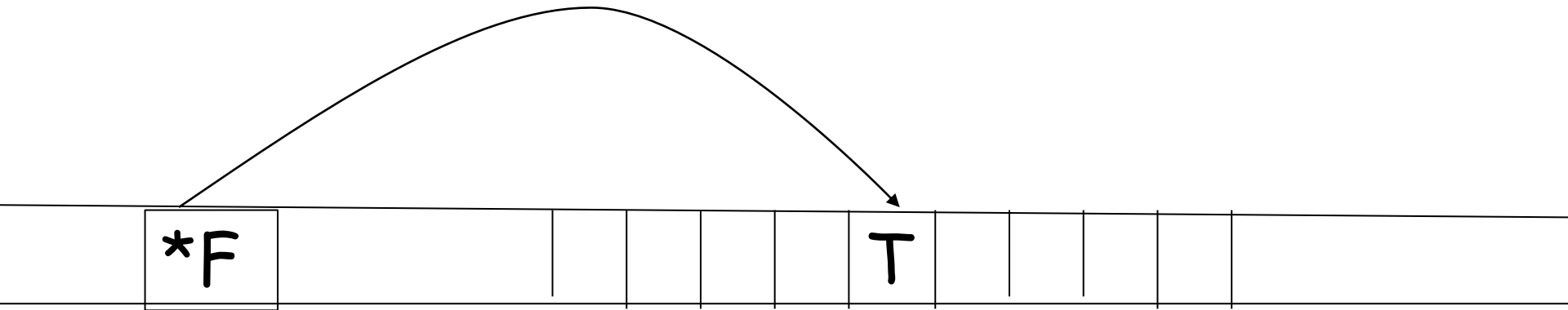


$F+1 = F+315*8$   
 $F+2 = F+2*315*8$   
 $F-5 = F-5*315*8$   
eccetera

} tutti valori di tipo  
double (\*) [5][7][9]

\*F ha tipo `double (*) [7][9]`

T ha dimensione =  $(7*9)*8=63*8$

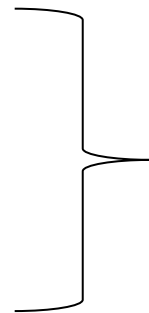


$$*F + 1 = F + 63 * 8$$

$$*F + 2 = F + 2 * 63 * 8$$

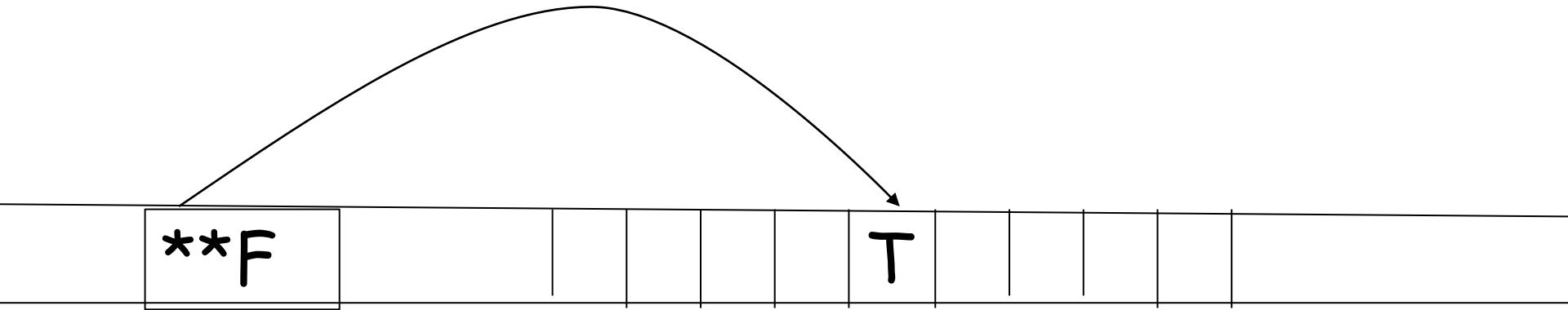
$$*F - 5 = F - 5 * 63 * 8$$

eccetera



tutti valori di tipo  
`double (*) [7][9]`

**\*\*F** ha tipo `double (*) [9]`  
T ha dimensione =  $9 \times 8 = 72$



$**F + 1 = F + 72$   
 $**F + 2 = F + 2 \times 72$   
 $**F - 5 = F - 5 \times 72$   
eccetera

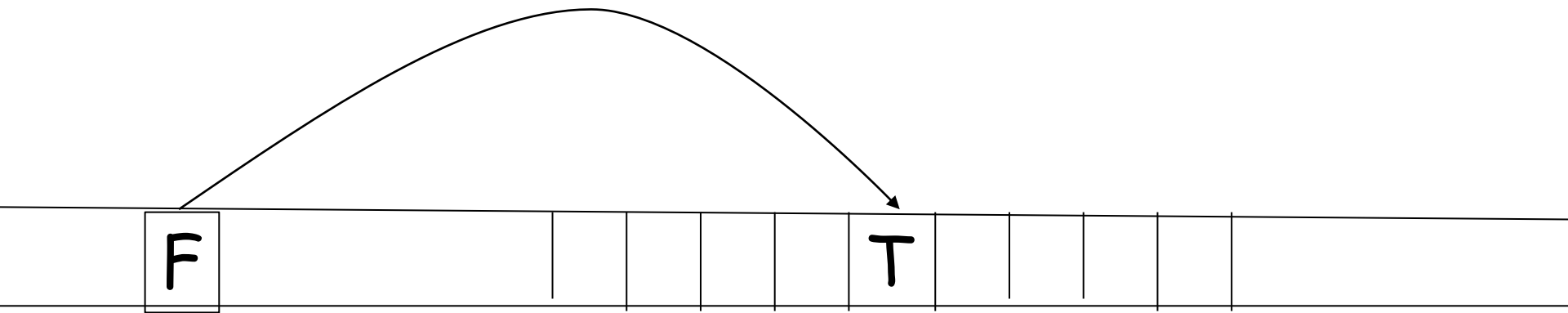
} tutti valori di tipo  
`double (*) [9]`

e subscripting ?

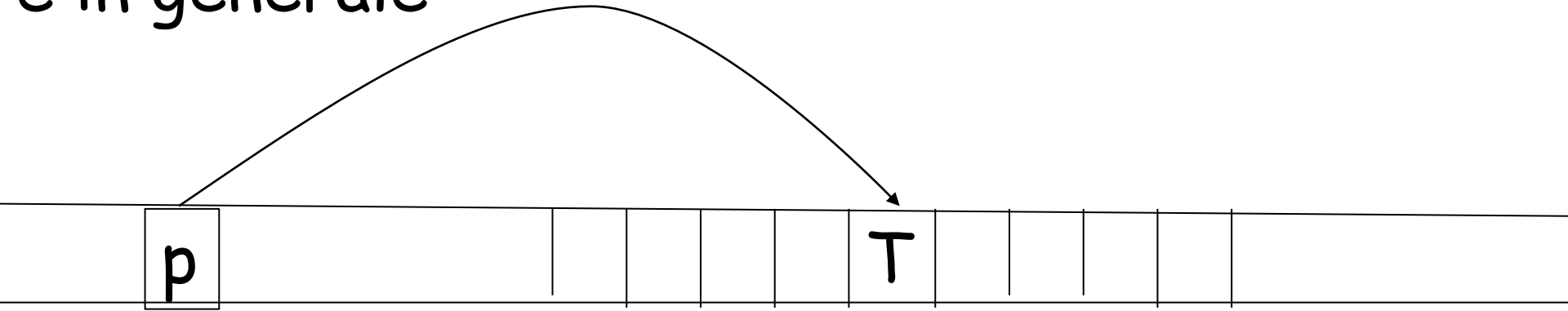
double F[3][5][7][9]; **tipo = double (\*)[5][7][9]**

$F[3] = *(F+3)$        $F[-2] = *(F-2)$

T ha dimensione =  $(5*7*9)*8=315*8$



e in generale



$$p[k] = *(p+k)$$

attenzione: nel  $[k]$  c'è sempre la  
dereferenziazione \*

CAPIRE :

```
float K[3][5][7][10];
```

```
K[1]= *(K+1)
```

```
K[3][2]=*(* (K+3)+2)
```

```
K[2][1][4][1]= *(*(*(* (K+2)+1)+4)+1)
```

## esercizio

float K[3][5][7][10];

indicando con K il valore di K, che valore ha K[2][2]?

$$K[2] = K + 2 * (5 * 7 * 10) * 4 = K + 2800 = L1$$

$$K[2][2] = L1 + 2 * (7 * 10) * 4 = L1 + 560$$

$$K[3][5][10] = ?$$

float K[3][5][7][10];

K[-1][-2] = ?

$K[-1] = K - (5 * 7 * 10) * 4 = L1$

$K[-1][-2] = L1 - 2 * (7 * 10) * 4$

$K[-1][-2][5] = K - (5 * 7 * 10) * 4 - 2 * (7 * 10) * 4 + 5 * 10 * 4$



# CAPIRE

double F[3][5][7][9];

allora \*\*\*F ha tipo double \* e

\*\*\*(F+4) ha anch'esso tipo double\*

la dereferenziazione cambia il tipo,

ma l'aritmetica cambia il valore non il tipo !!!

## tipo degli array e aritmetica dei puntatori

`int A[3][3][2][5][10][20]`

`A : int(*) [3][2][5][10][20]`

`A+3 : tipo? valore ?`

`A[3] : int (*) [2][5][10][20]`

`*(A+3): tipo? valore ?`