

Esempi aritmetica dei puntatori

(NB: la dereferenziazione tramite * ha precedenza sugli operatori di somma e differenza)

-int A[5][10][20]

ha tipo `int(*)[10][20]` e dimensione $(10*20)*4$ byte

`*A-3` ha tipo `int(*)[20]`, dimensione $20*4$ byte e valore: $A-3*20*4$

-double X[4][10][3][8]

ha tipo `double(*)[10][3][8]` e dimensione $(10*3*8)*8$ byte

`*(X+4)-2` ha tipo `double(*)[8]`, dimensione $8*8$ byte e valore: $X+4*(3*8)*8-2*8*8$

`X[1]` (equivale a scrivere `*(X+1)`) ha tipo `double(*)[3][8]`, dimensione $3*8*8$ byte e valore: $X+1*(10*3*8)*8$

`(X-2)[6][1]` (equivale a scrivere `*((X-2)+6)+1`) ha tipo `double(*)[8]` e valore:

$X-2*(10*3*8)*8+6*(10*3*8)*8+1*(3*8)*8$

-char C[4][3]

ha tipo `char(*)[3]` e dimensione $3*1$ byte

`**(C-3)` ha tipo `char` e valore: $C-3*3*1$

-char X[10][5][10]

ha tipo `char(*)[5][10]` e dimensione $5*10*1$ byte

`*(X-4)+2` ha tipo `char(*)`, dimensione 1 byte e valore: $X-4*10*1+2*1$

`X[-1]` (equivale a scrivere `*(X-1)`) ha tipo `char(*)[10]`, dimensione $10*1$ byte e valore: $X-1*(5*10)*1$

-int A[4][3][6]

ha tipo `int(*)[3][6]` e dimensione $3*6*4$ byte

`*(A+3)-10` ha tipo `int(*)[6]`, dimensione $6*4$ byte e valore: $A+3*(3*6)*4-10*6*4$

`*(A+2)-10` ha tipo `int(*)`, dimensione 4 byte e valore: $A+2*6*4-10*4$

-double B[10][11][12]

ha tipo `double(*)[11][12]` e dimensione $11*12*8$ byte

`*(B-5)-10` ha tipo `double(*)[12]`, dimensione $12*8$ byte e valore: $B-5*12*8-10*12*8$

`*(B[20]-3)` (equivale a scrivere `*(B+20)-3`), ha tipo `double(*)`, dimensione 8 byte e valore: $X+20*(11*12)*8-3*12*8$