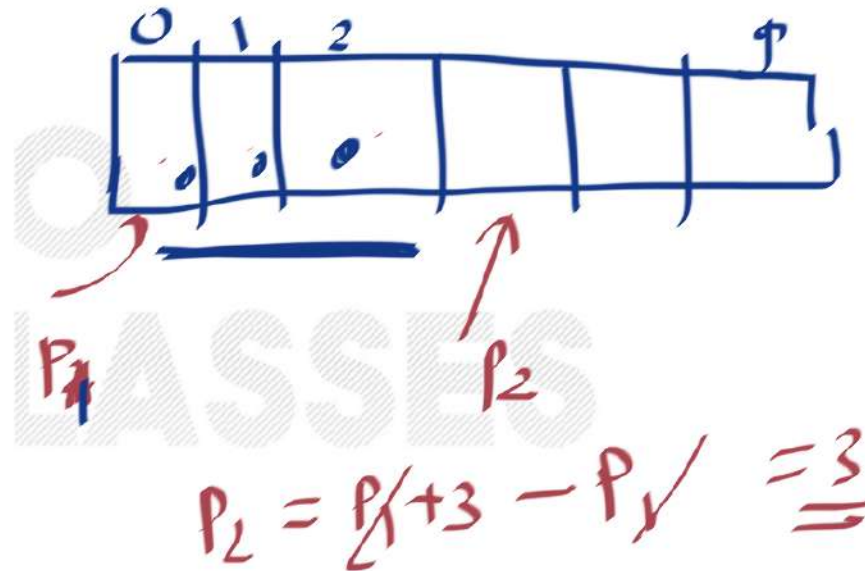


Pointer arithmetic (difference between two pointers)

```
int a[10];  
int *p1 = a;  
int *p2 = p1+3;
```

```
printf("%d", p2-p1);
```



When two pointers are subtracted, both **shall point to elements of the same array object.**

If not then result can be any garbage value or error.

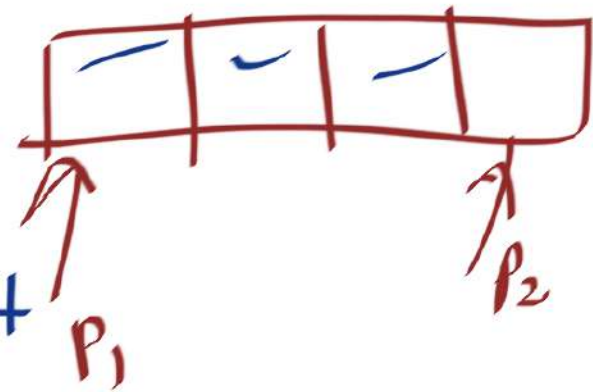
$$p_1 \rightarrow 1000$$

$$p_2 = p_1 + 3$$

$$= 1000 + 3 * \text{sizeof}(*p_1)$$

$$= 1000 + 3 * 4$$

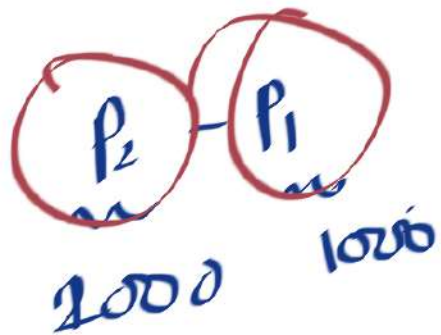
$$p_2 = \underline{\underline{1012}}$$



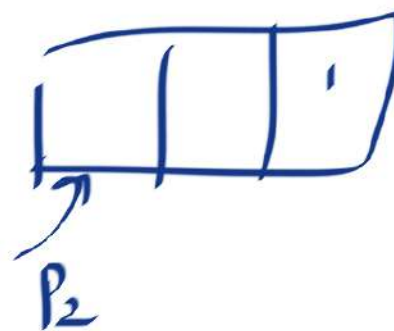
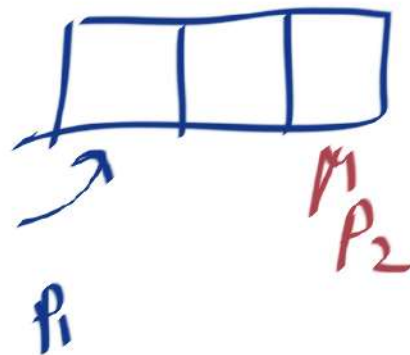
$$\underline{p_2 - p_1 = 3}$$

$$p_2 - p_1 = \frac{1012 - 1000}{\text{sizeof}(*p_1)}$$

$$= \frac{12}{4} = 3$$



$$\frac{2000 - 1000}{4}$$



Subtracting two pointers does not result in a pointer ("address"), it results in integer which is the "distance" between those pointers. This only makes sense if the pointers both point into the same array. Likewise, comparing pointers only makes sense if they are in the same array.

When it doesn't make sense, the result is undefined--that does NOT mean that the program will fail, or crash, or produce an error of any kind. It means ANYTHING may happen, and you have no right to complain.

Share Edit Follow

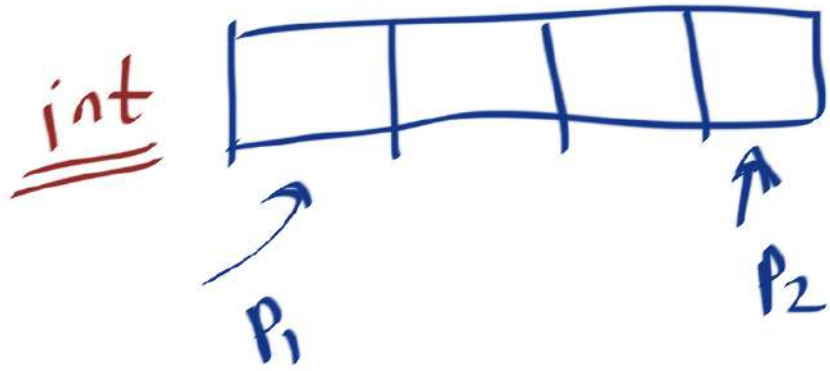
answered Nov 8, 2017 at 20:37



[Lee Daniel Crocker](#)

12.7k ● 1 ● 28 ● 50

Add a comment

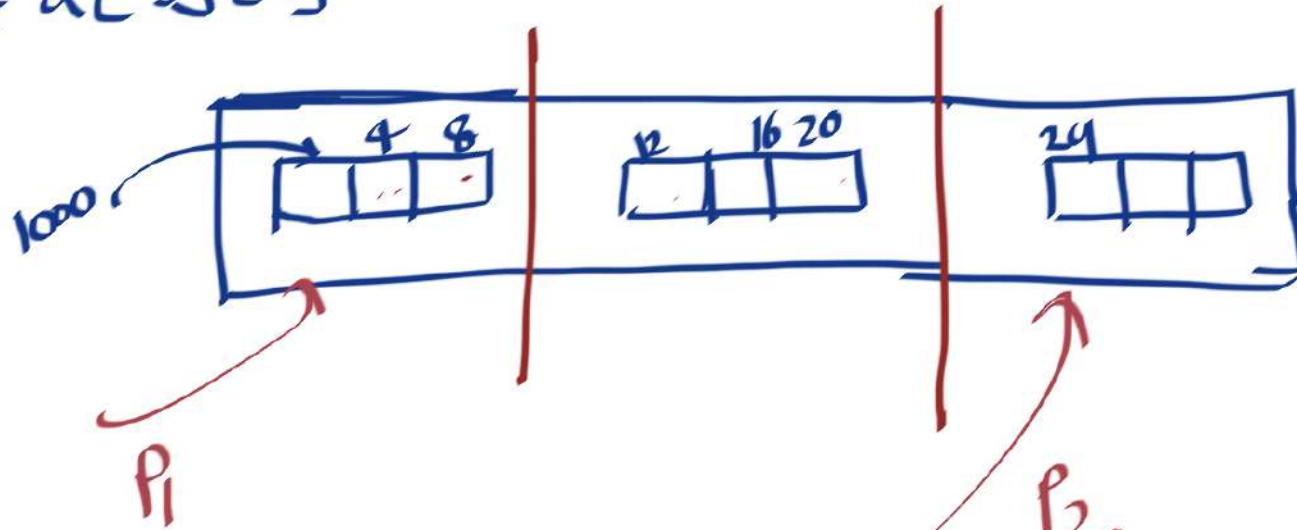


$\text{sizeof}(p_1) \rightarrow 8$
 $\text{sizeof}(*p_1) = 4$

$$p_2 - p_1 = 3$$

$$p_1 - p_2 = -3$$

int a[3][3]



$$\left[\begin{array}{l} \text{int } (*P_1)[3] \\ P_1 = a \end{array} \right]$$

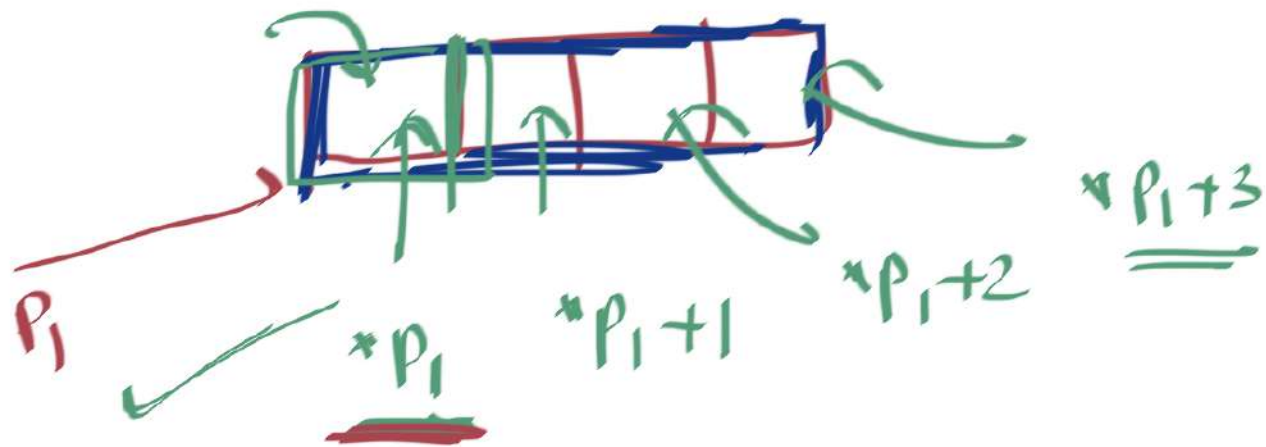
$$\left[\begin{array}{l} \text{int } (*P_2)[3] \\ P_2 = a + 2 \end{array} \right]$$

$$P_2 - P_1 = \textcircled{2}$$

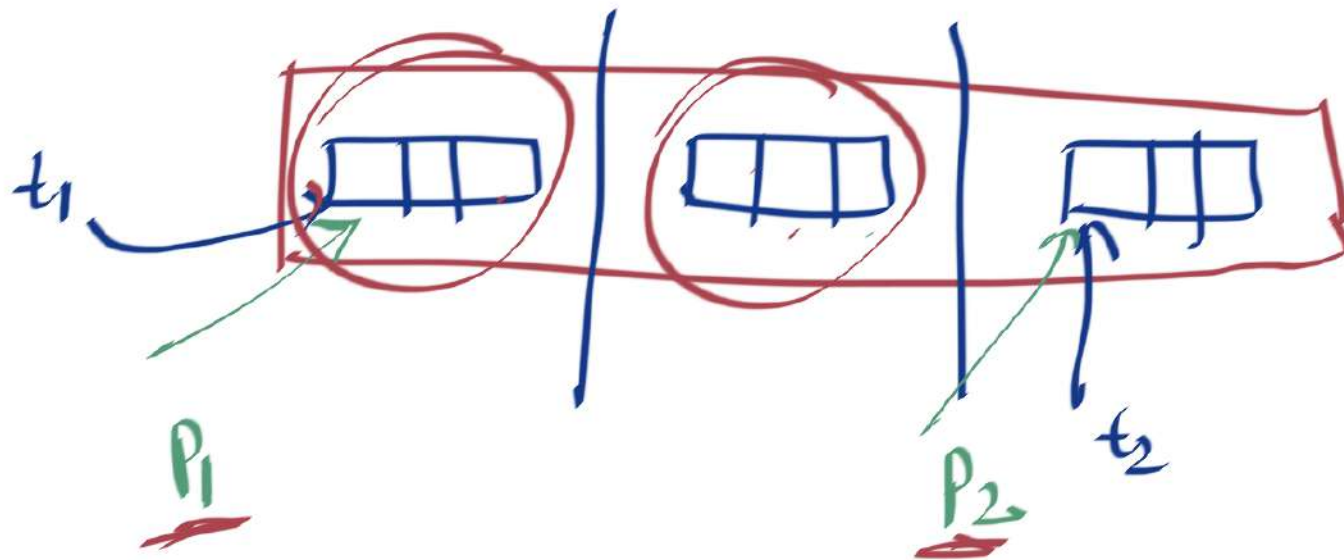
$$\frac{1024 - 1000}{\text{sizeof } (*P_1)} = \frac{24}{12} = \textcircled{2}$$

p_1 is pointer to an array of size 3
 $*p_1$ \rightarrow array of size 3

p is pointer to an integer
 $*p$ \rightarrow int



$*p_1$ is an array



$\text{int } *t_1$

$\text{int } *t_2$

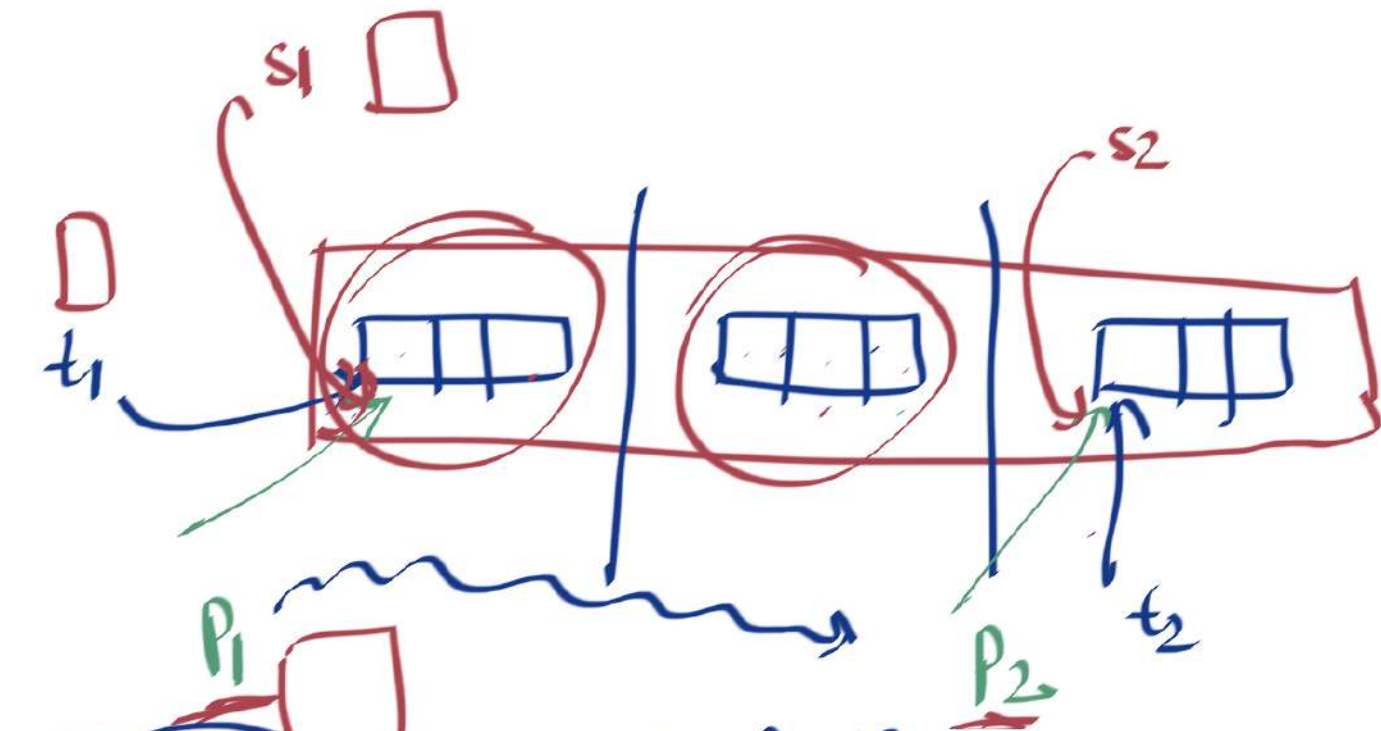
$\text{int } (*p_1)[3]$

$\text{int } (*p_2)[3]$

$$\underline{p_2 - p_1} = \underline{2} \text{ elements} \quad t_2 - t_1 = \underline{6}$$

$p_2 + 1$ → array of size 3

int



$$(\hat{p}_1 + 2) = p_2 \Rightarrow p_2 - p_1 = 2$$

$$s_2 - s_1 = 6/2 = 3$$

$$\underline{\underline{t_1 = p_1}}$$

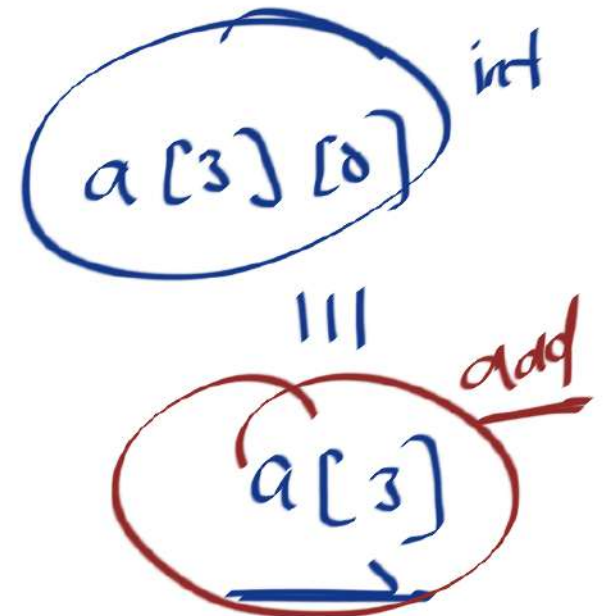
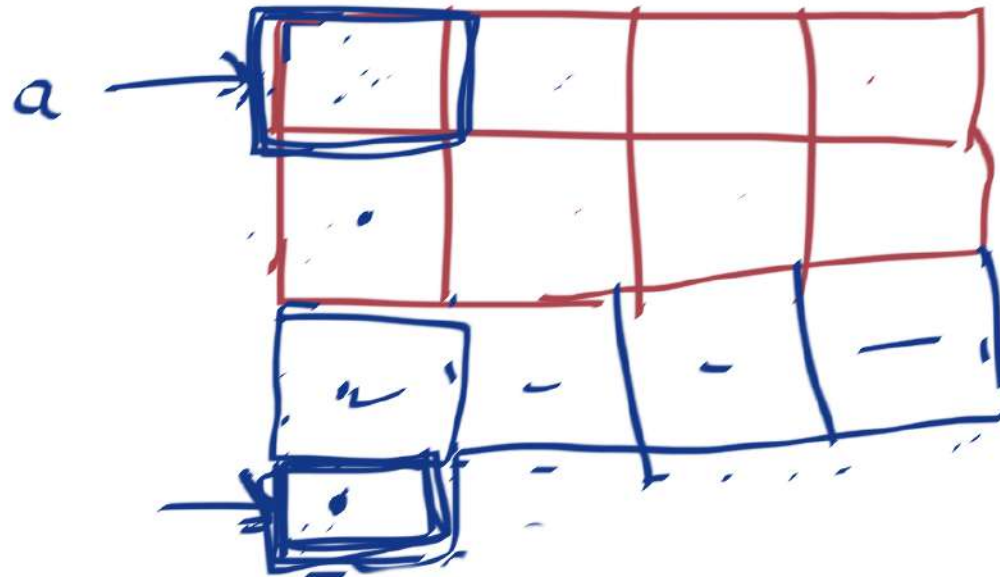
$$\frac{p_2 - p_1}{3} = 6/3 = 2$$

int *t1
int *t2

int (*p1)[3]
int (*p2)[3]

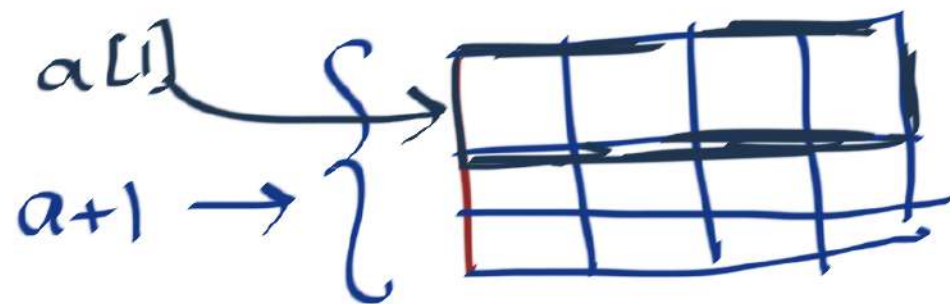
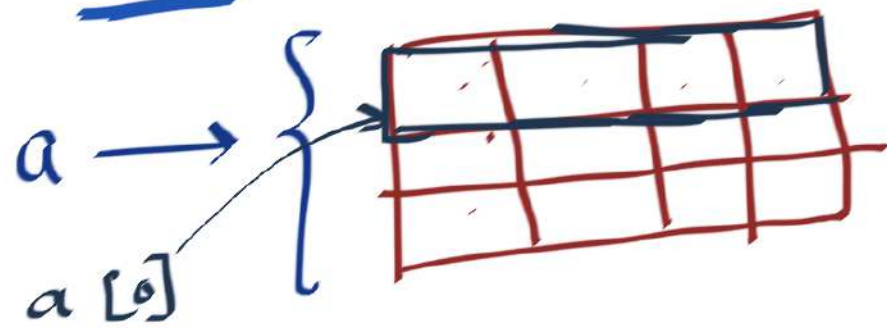
int (*s1)[2]
int (*s2)[2]

int a[2][4];



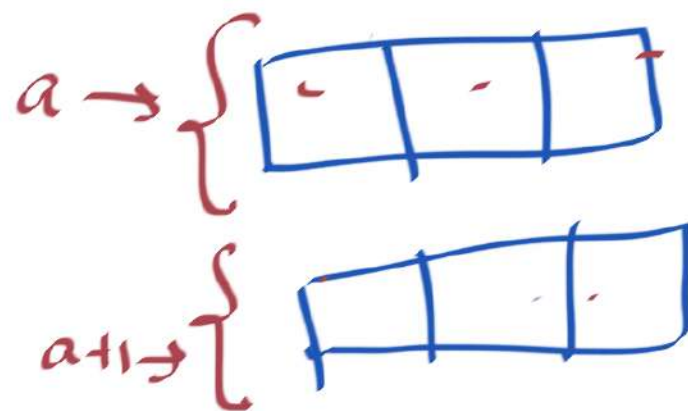
a[3] — a[0] \equiv (12)
int pointer int pointer

$a[2][3][4]$

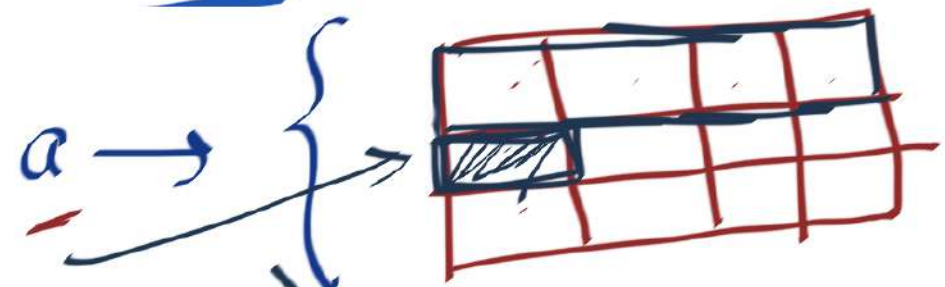


$$\underbrace{a[1]} - \underbrace{a[0]} = \underline{\underline{3}} \quad \frac{12}{4} = \underline{\underline{3}}$$

$a[2][3]$



$a[2][3][4]$



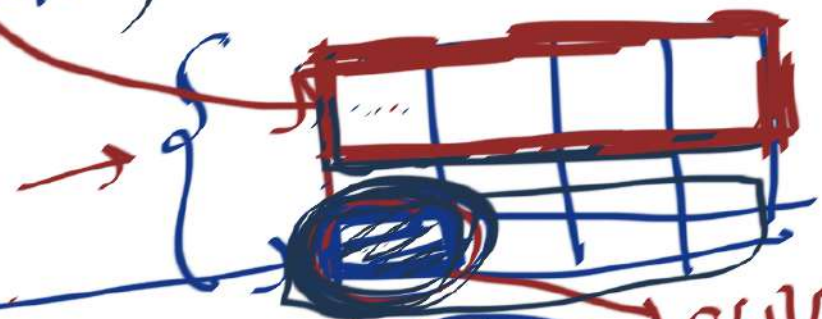
$a[1][2]$

$$p(\downarrow (a+1) + 2) \rightarrow$$



$(a[1] + 2)$

$a+1$



$(a[1] + 2)$
 $a[1]$
 $(a+1)$

$a[1][2]$

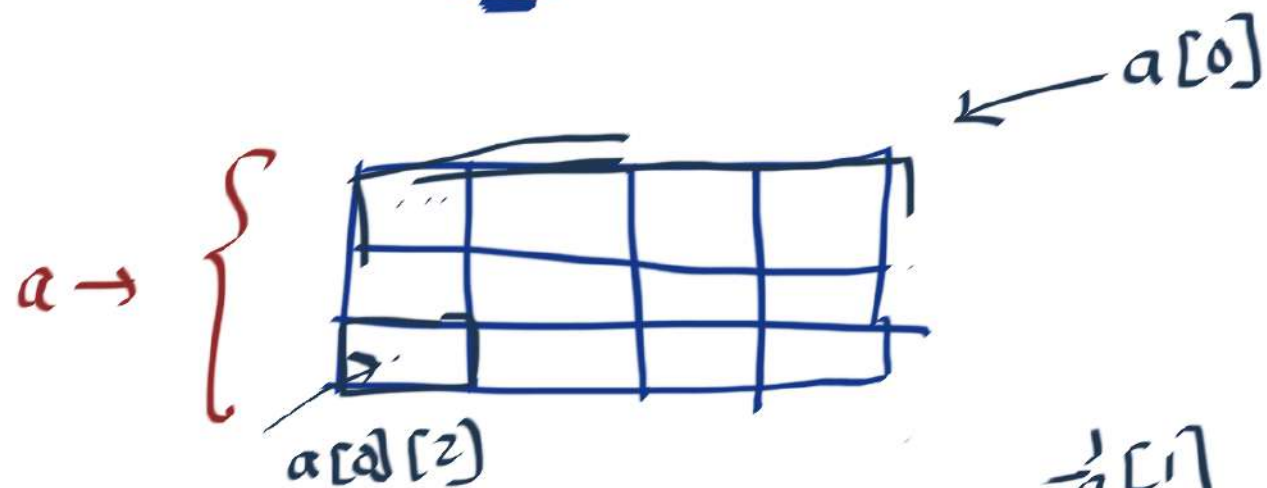
$a[1][2]$

$a[0][1][0]$

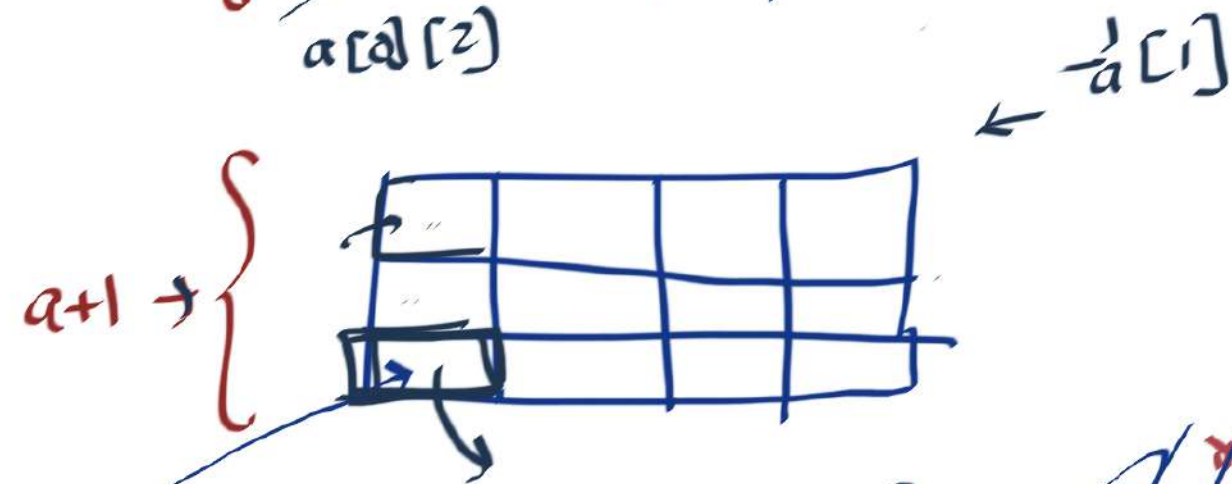
$a[1][2][0]$

16

int a[2][3][4]



$a[1][2][0]$

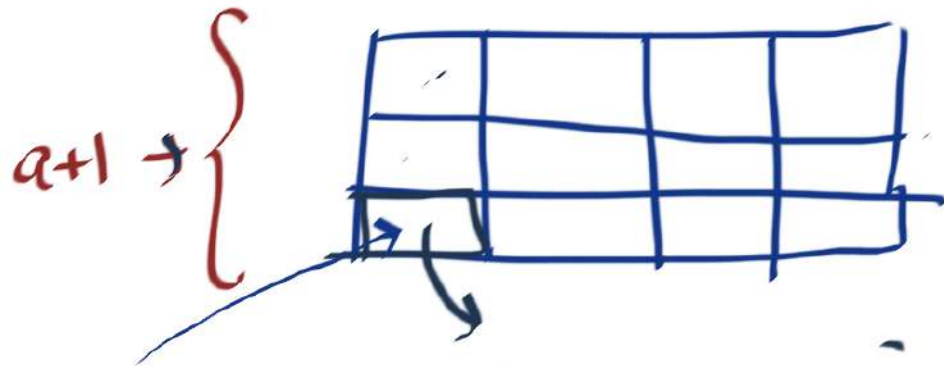
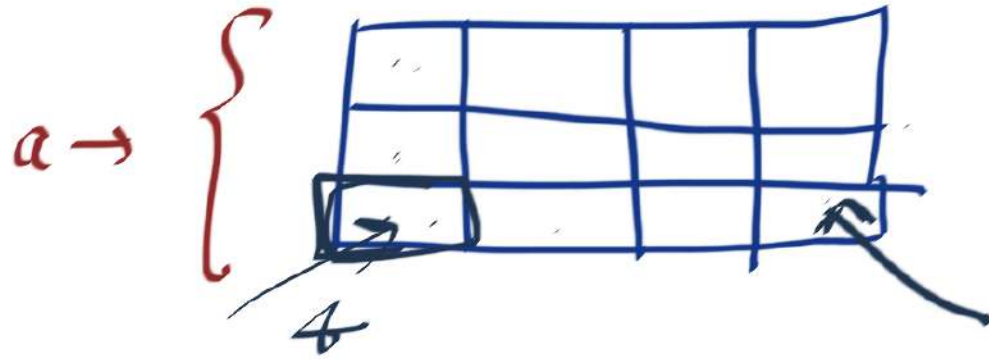


= $a[1][2]$

$a[1][2]$ $\equiv \cancel{768} * (a[1][2][0] + 0)$

int a[2][3][4]

a[0]

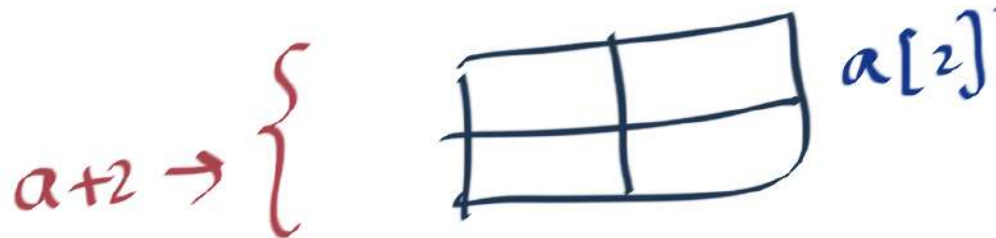
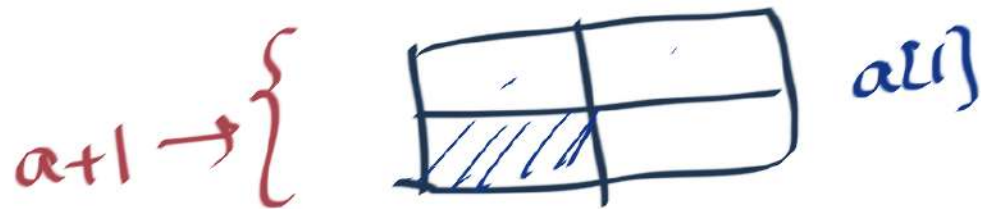
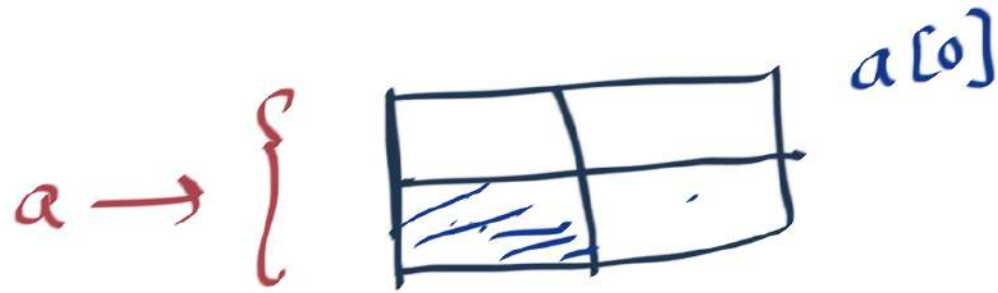


a[0][2] ~~[0]~~

4 * (a[0][2] + 3)

4 a[0][2][¹¹3]

$a[3][2][2]$



$$\begin{array}{c} a[0][1] = a[1][1] \\ \downarrow \\ = \underline{-4} \end{array}$$

int a[3][2][2]

$$\overset{1008}{a[0][1]} - \overset{1024}{a[1][1]} = \overset{-16}{\frac{-16}{4}} = -4$$

a → 1000

$$\Rightarrow *(* (a + 0) + 1)$$

$$= 1000 + 1 * \text{sizeof}(\text{ })$$

whatever it points

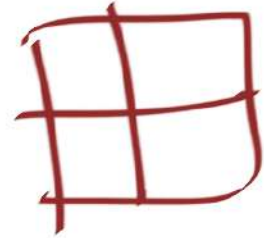
$$= \underline{\underline{1008}}$$

1000

type is 1D array of size 2

1000 type int array

a[i][i], 1024 ↪ int pointer



((a+1) + 1)



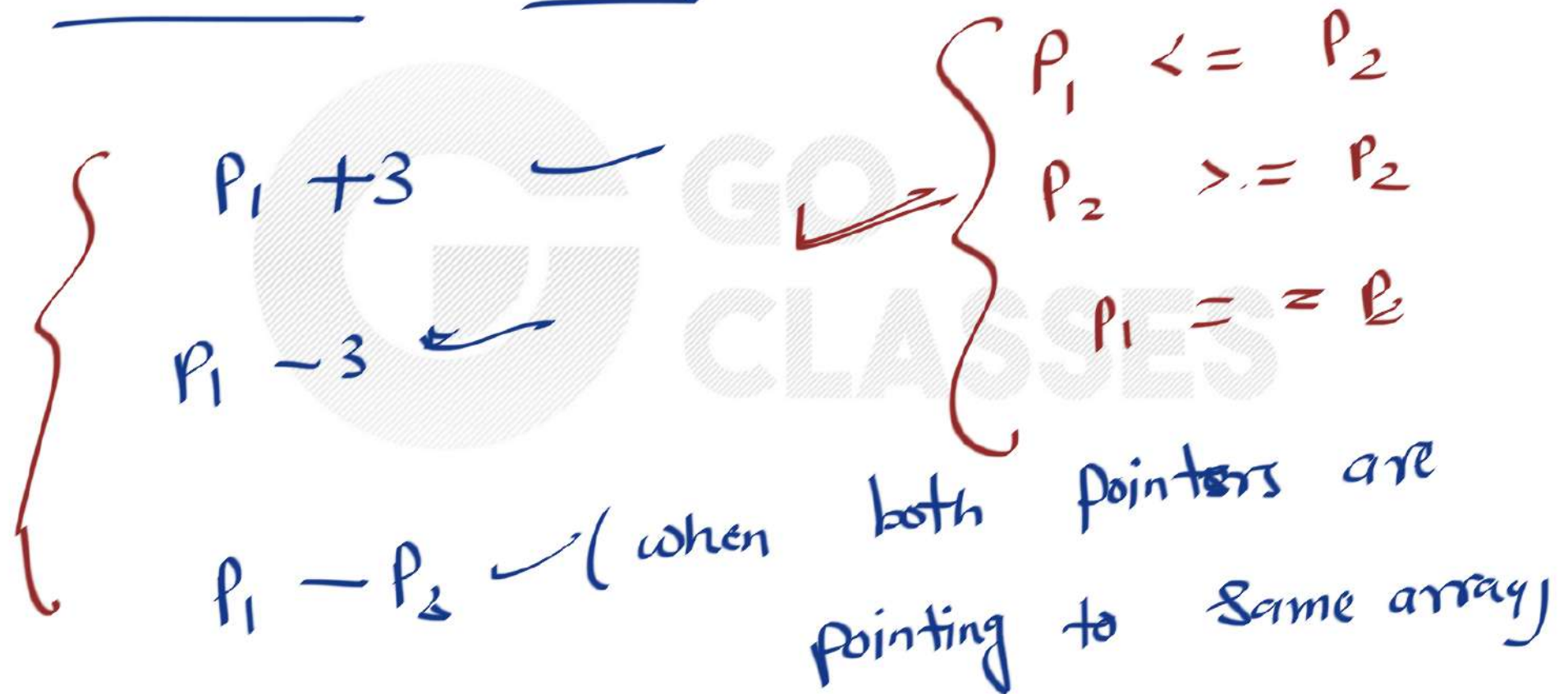
1000+16

1016+8

= 1024

* 1016 type 1D array

Valid arithmetic on pointers



Invalid arithmetic on pointers

$p1 + p2$

$p1 * p2$

$p1 \% p2$

$p1 / p2$