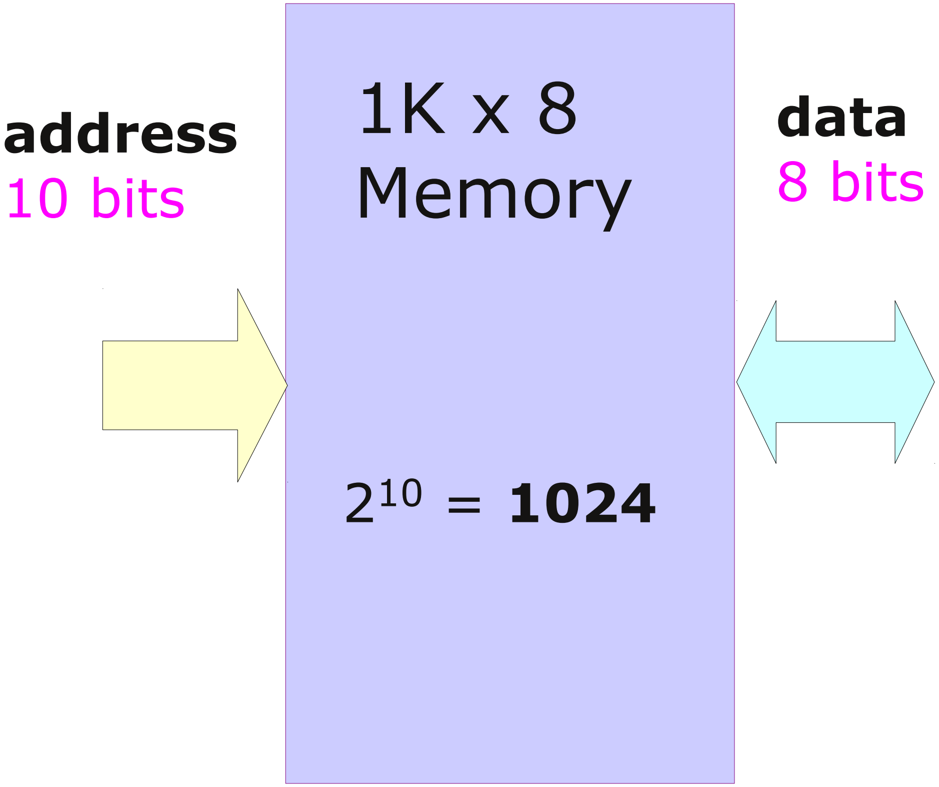
Pointer (1A)

Young Won Lim
7/16/11

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice.

Address and Data in a Memory



address
10 bits

1K x 8
Memory

2^{10} = **1024**

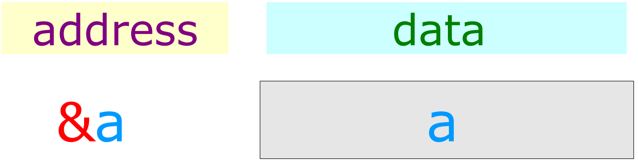
data
8 bits

	address	data
1023	0x3FF	
1022	0x3FE	
1021	0x3FD	
1020	0x3FC	
		• • •
0003	0x003	
0002	0x002	
0001	0x001	
0000	0x000	
DEC address	HEX address	data
	10 bits	8 bits

Variable

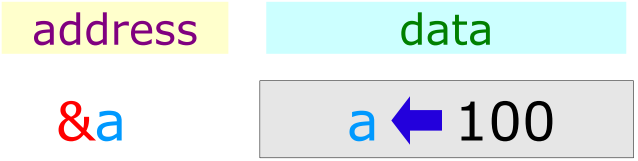
```
int    a;
```

a can hold an *integer*



```
a = 100;
```

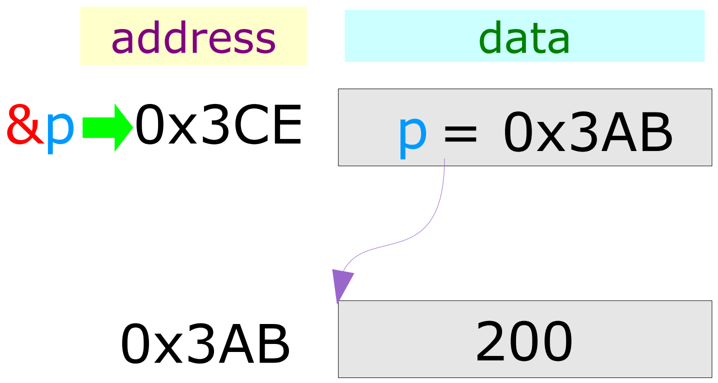
a holds an *integer* 100



Pointer Variable

```
int *     p;
```

p can hold an *address*



$\underbrace{\texttt{int} \quad \texttt{*}}_{\textit{pointer to int}} \ \underbrace{\texttt{p;}}$

p holds an *address* of Int type.

$\underbrace{\texttt{int}}_{\textit{int}} \ \underbrace{\texttt{*} \quad \texttt{p;}}$

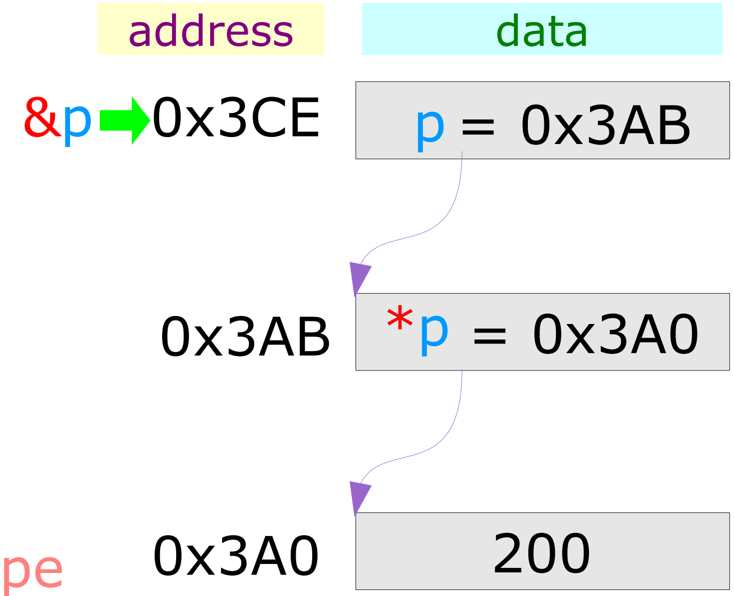
*p holds an *integer*

&p ➡ 0x3CE
p ➡ 0x3AB
*p ➡ 200

Pointer to Pointer Variable

```
int **      p;
```

p can hold an *address*



$\underbrace{\texttt{int **}}_{\textit{pointer to pointer to int}}$ `p;`

p holds an *address* of Pointer to Int type

$\underbrace{\texttt{int *}}_{\textit{pointer to int}}$ `*p;`

*p holds an *address* of Int type

$\underbrace{\texttt{int}}_{\textit{int}}$ `**p;`

**p holds an *integer*

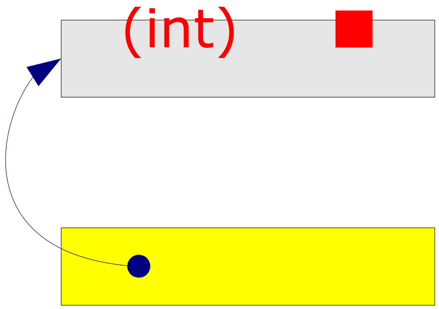
&p → 0x3CE

p → 0x3AB

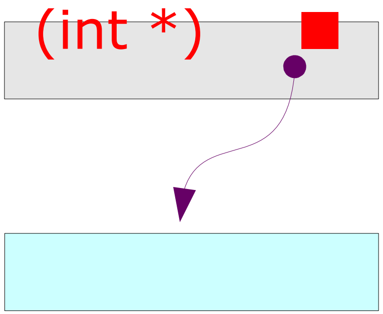
*p → 0x3A0

**p → 200

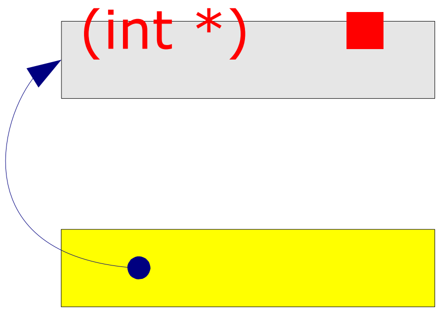
Interpretation of Pointer (1)



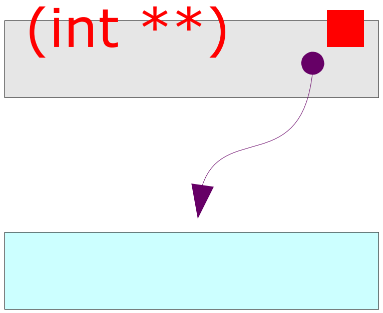
This type must be (int *)



This type must be (int)



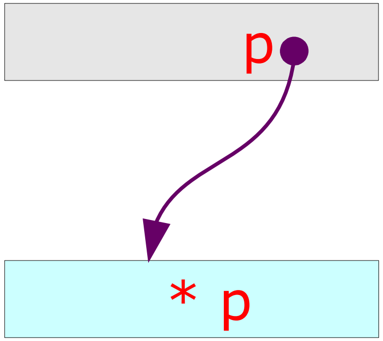
This type must be (int **)



This type must be (int *)

Interpretation of Pointer (2)

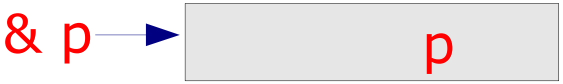
Following a pointer :
*Dereferencing operator **



p

* p

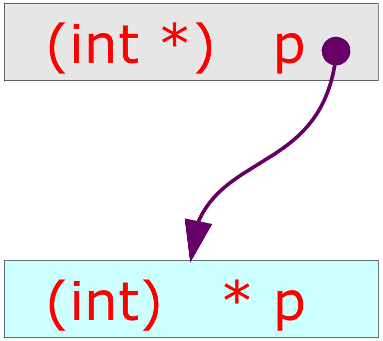
The address of a variable :
Address of operator &



& p

p

If p is a pointer to integer type



(int *) p

(int) * p

*If *p is an integer type*

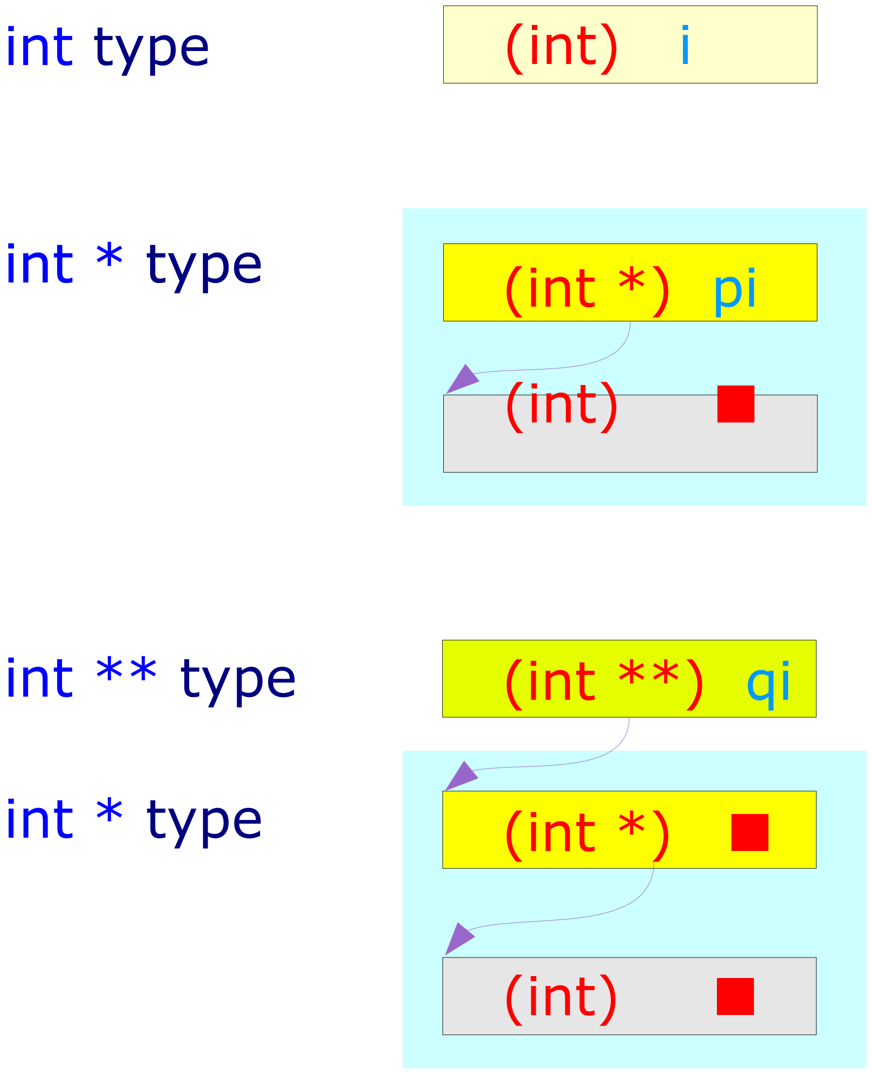
Integer Pointer Examples (1)

```
int      i;
int *    pi;
int **   qi;
```

i holds an *integers*

pi holds an *address*
of Int type

qi holds an *address*
of Pointer to Int type



int type

(int) i

int * type

(int *) pi

(int) ■

int ** type

(int **) qi

int * type

(int *) ■

(int) ■

Integer Pointer Examples (2)

```
int      i;
int *    pi;
int **   qi;
```

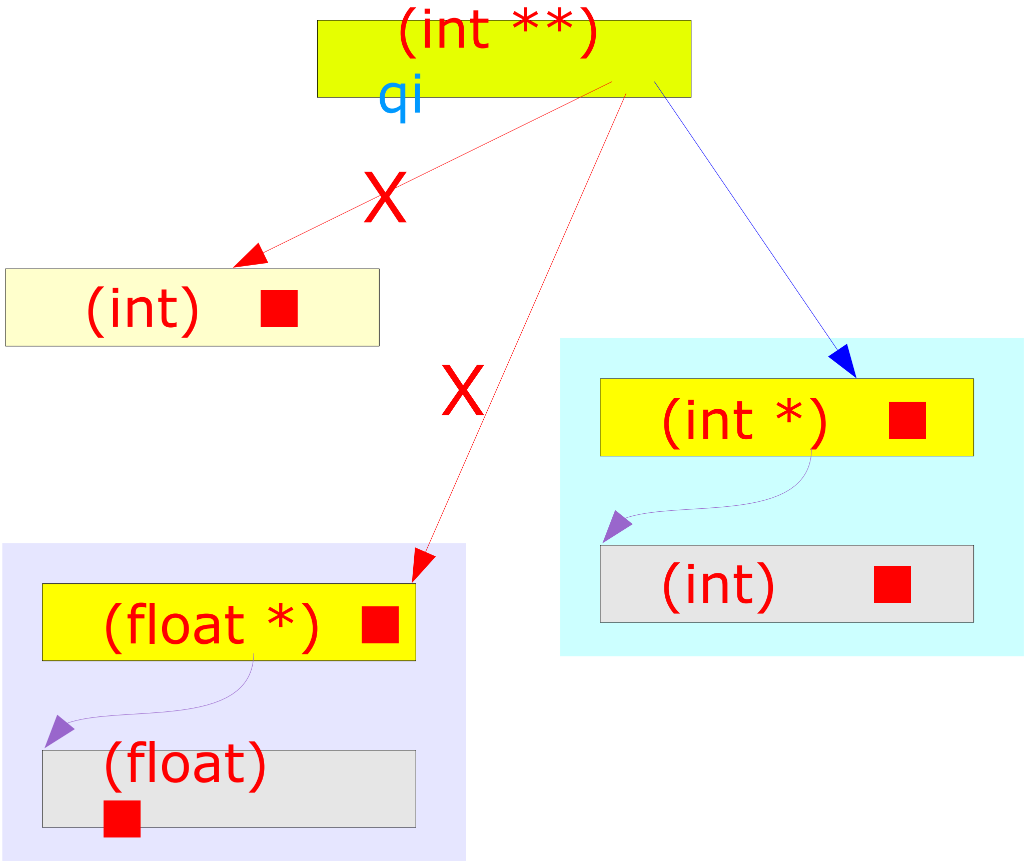
i holds an *integers*

pi holds an *address*

of Int type

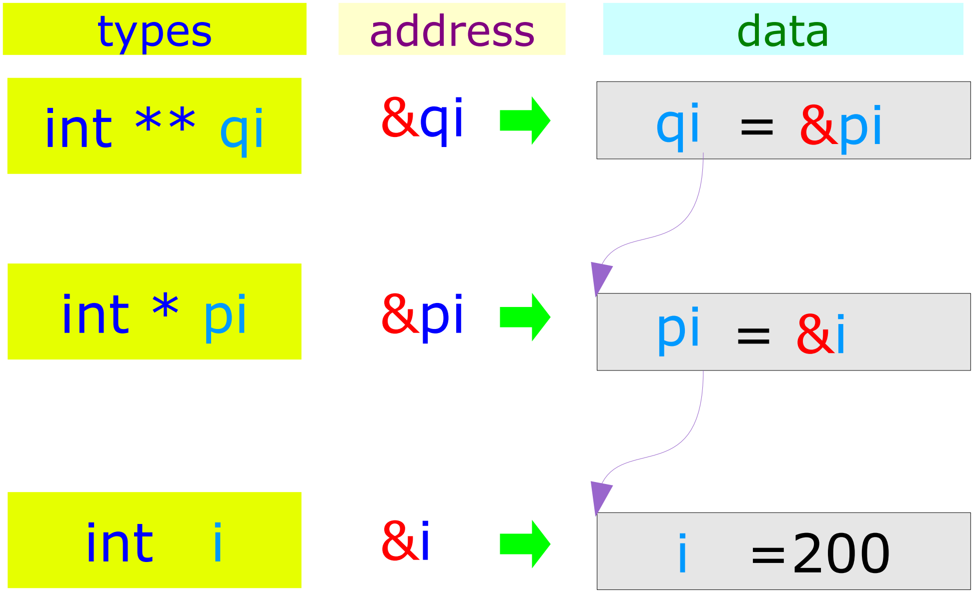
qi holds an *address*

of Pointer to Int type



Integer Pointer Examples (3)

```
int       i = 200;
int *     pi = &i;
int **    qi = &pi;
```



i holds an *integers*

pi holds an *address*
of Int type

qi holds an *address*
of Pointer to Int type

*qi = pi

*pi = i

**qi = *pi = i

Array of Pointers (1)

```
int       a [4];

int *     b [4];
```

Array name a holds the starting *address*

int   a   [4]

No. of elements = 4

Type of each element

Array name b holds the starting *address*

int *   a   [4]

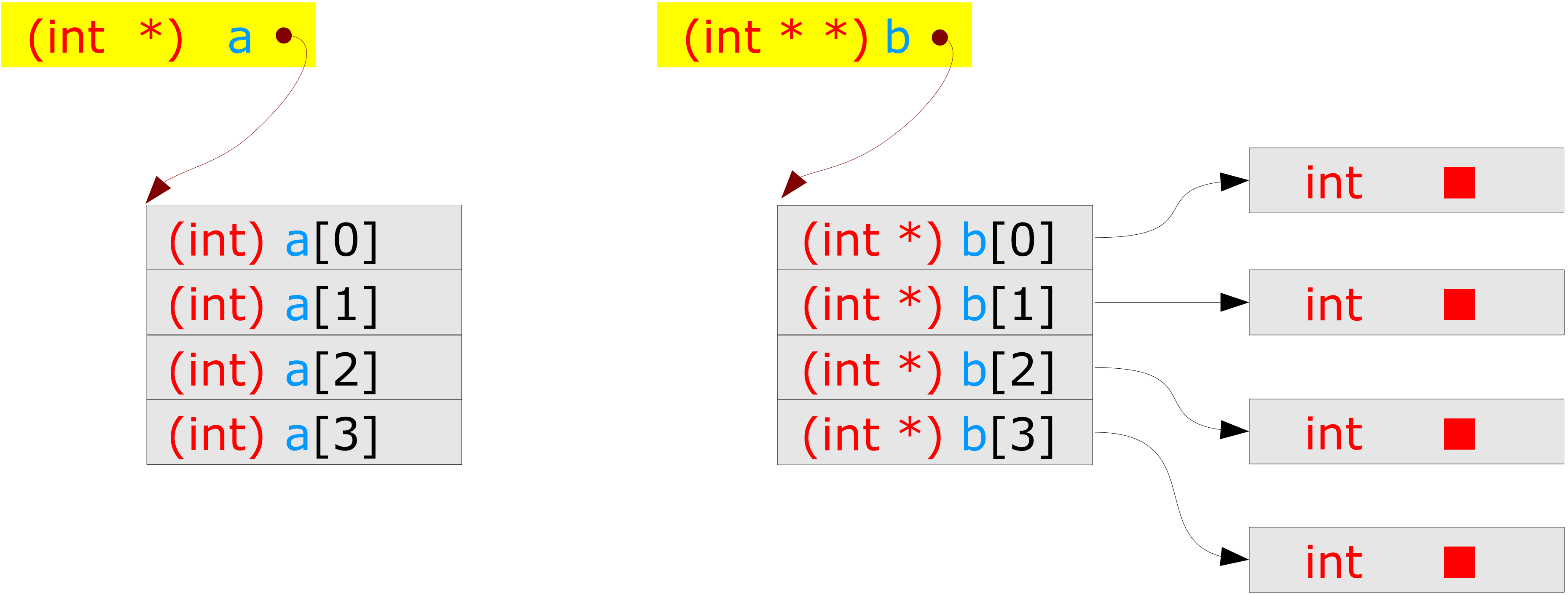
No. of elements = 4

Type of each element

Array of Pointers (2)

```
int        a [4];
```

```
int *      b [4];
```



(int *) a

(int) a[0]

(int) a[1]

(int) a[2]

(int) a[3]

(int * *) b

(int *) b[0]

(int *) b[1]

(int *) b[2]

(int *) b[3]

int 🟥

int 🟥

int 🟥

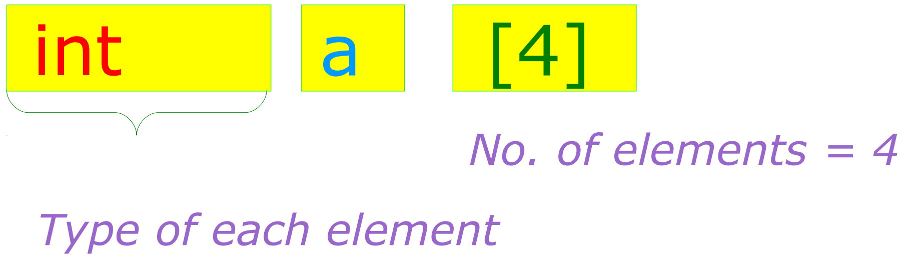
int 🟥

2-D Array (1)

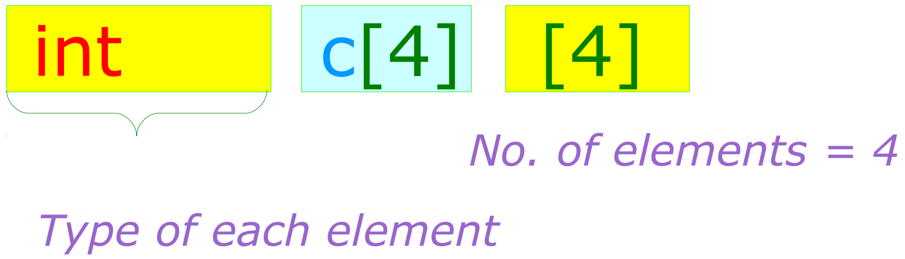
```
int        a [4];

int        c [4] [4];
```

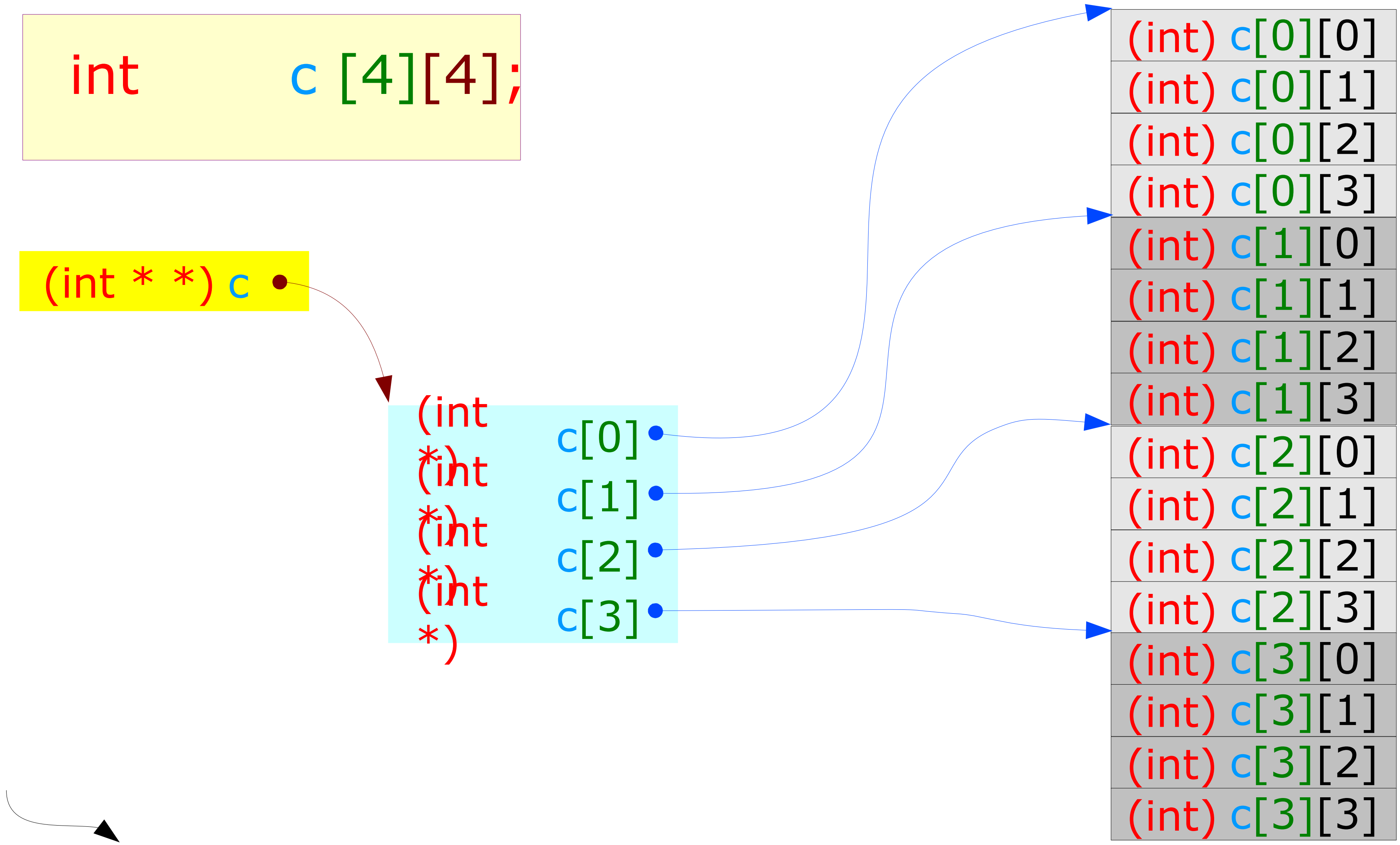
Array name a holds the starting *address*



c[0], c[1], c[2], c[3] holds the starting *address*



2-D Array (2)



int c [4][4];

(int * *) c

(int *) c[0]

(int *) c[1]

(int *) c[2]

(int *) c[3]

(int) c[0][0]

(int) c[0][1]

(int) c[0][2]

(int) c[0][3]

(int) c[1][0]

(int) c[1][1]

(int) c[1][2]

(int) c[1][3]

(int) c[2][0]

(int) c[2][1]

(int) c[2][2]

(int) c[2][3]

(int) c[3][0]

(int) c[3][1]

(int) c[3][2]

(int) c[3][3]

2-d Array

References

[1] Essential C, Nick Parlante
[2] Efficient C Programming, Mark A. Weiss
[3] C A Reference Manual, Samuel P. Harbison & Guy L. Steele Jr.
[4] C Language Express, I. K. Chun