Pointers (1A)

Young Won Lim
11/1/17

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

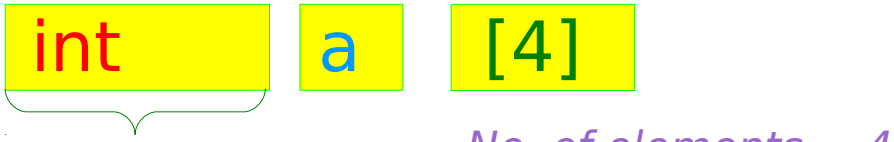
This document was produced by using OpenOffice.

Young Won Lim
11/1/17

Array of Pointers (1)

```
int     a [4];

int *   b [4];
```

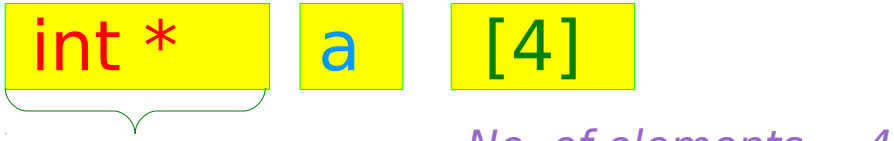
Array name a holds the starting *address*



No. of elements = 4

Type of each element

Array name b holds the starting *address*



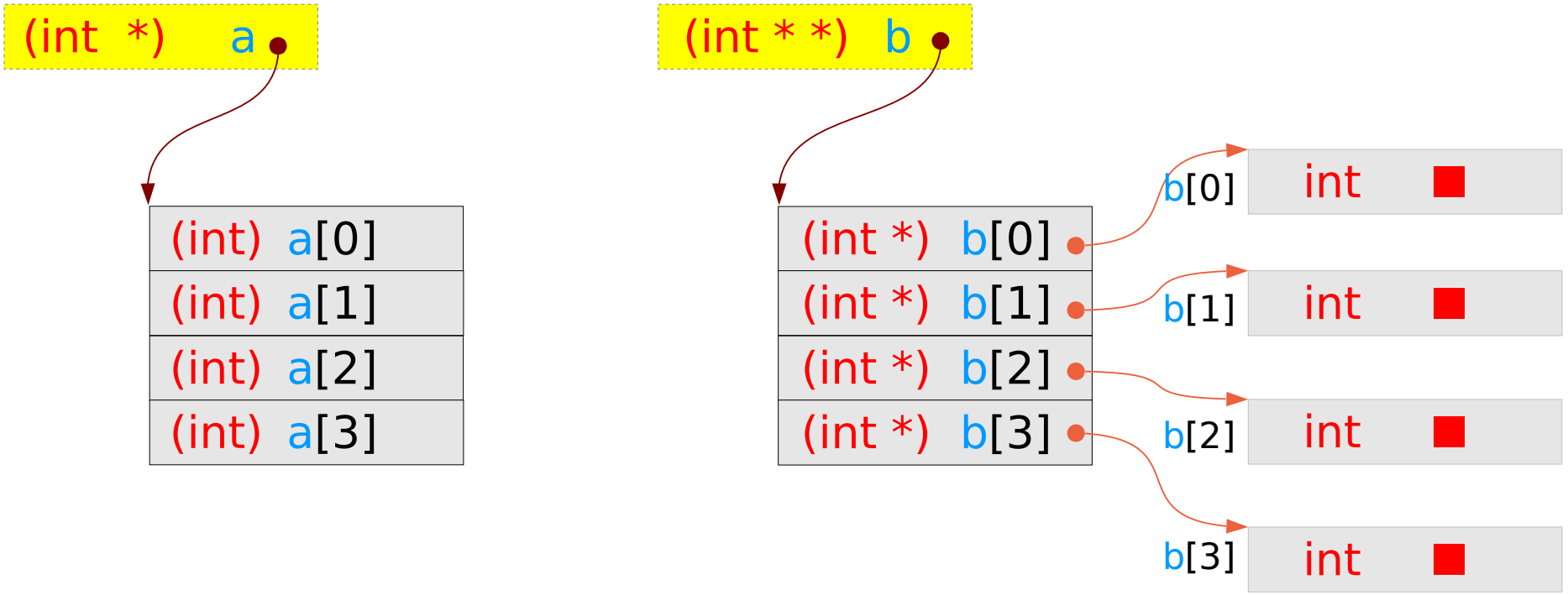
No. of elements = 4

Type of each element

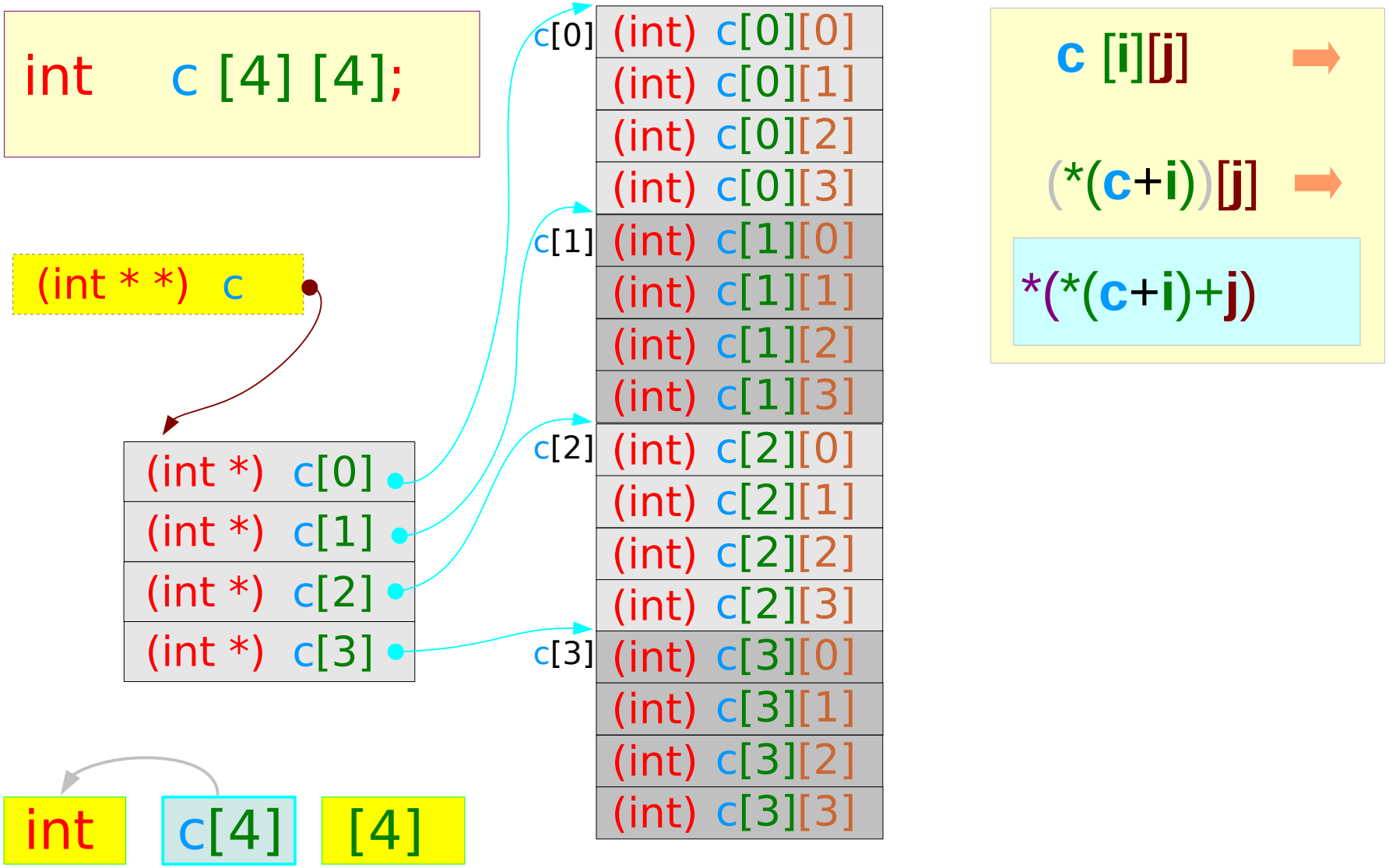
Array of Pointers (2)

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

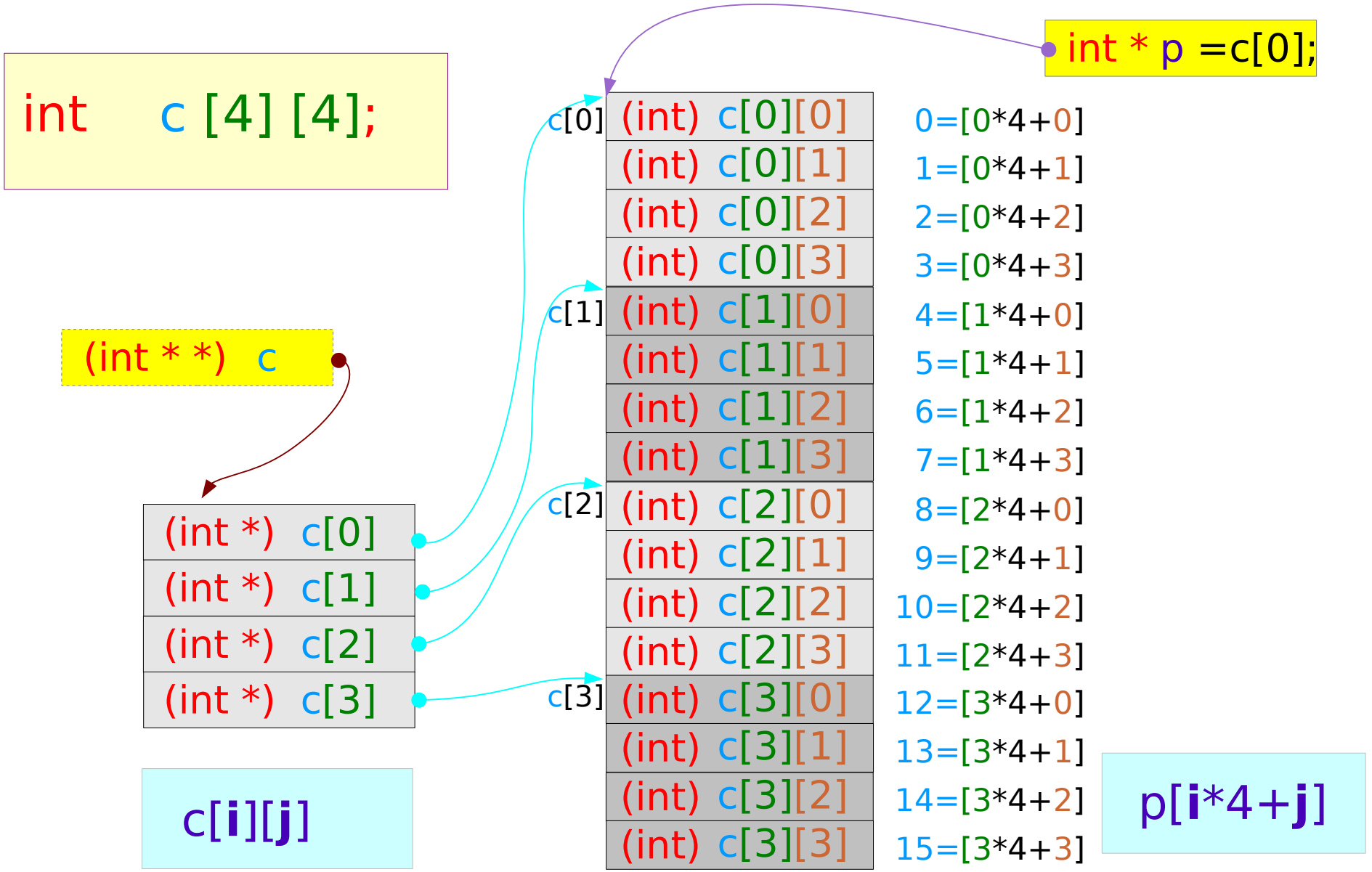
```
int *     b [4];
```



A 2-D Array via a double pointer

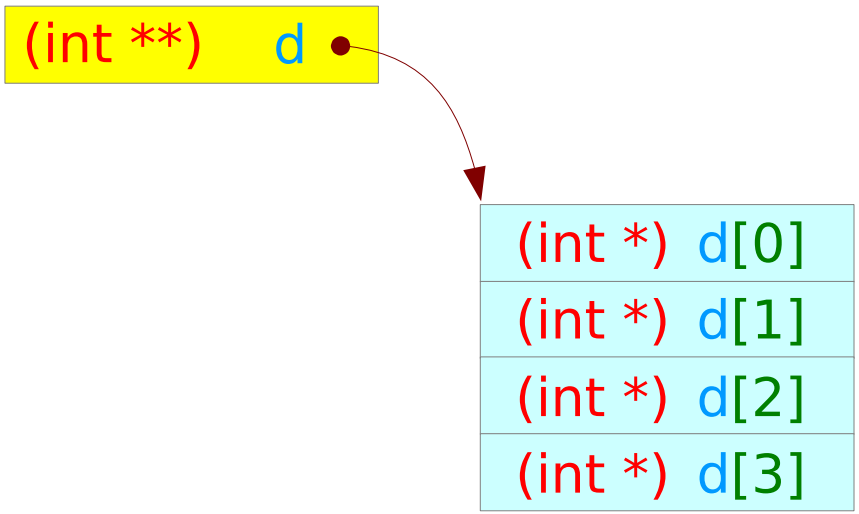


A 2-D array via a single pointer



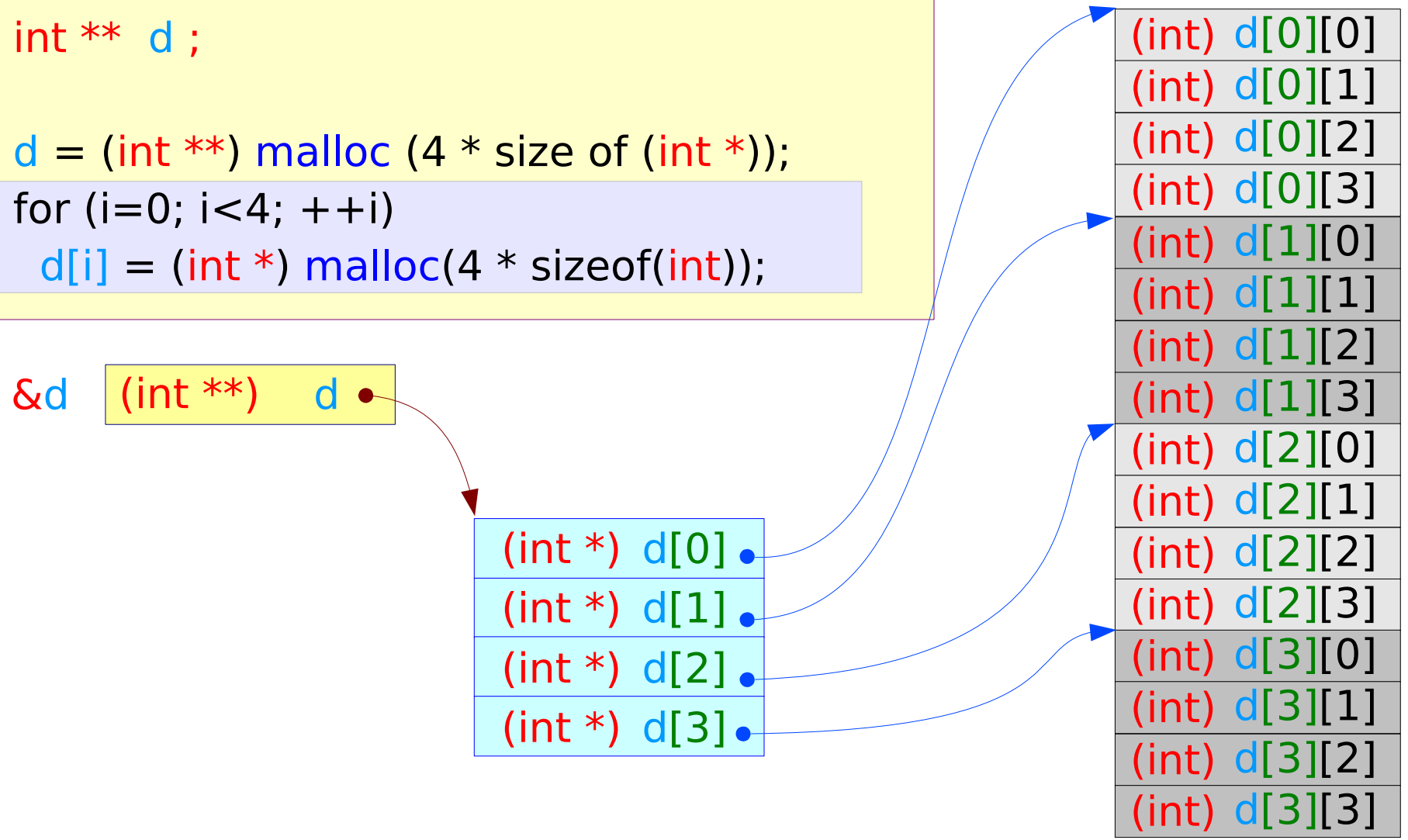
2-D Array Dynamic Memory Allocation (1)

```
int **  d ;

d = (int **) malloc (4 * size of (int *));
for (i=0; i<4; ++i)
  d[i] = (int *) malloc(4 * sizeof(int));
```



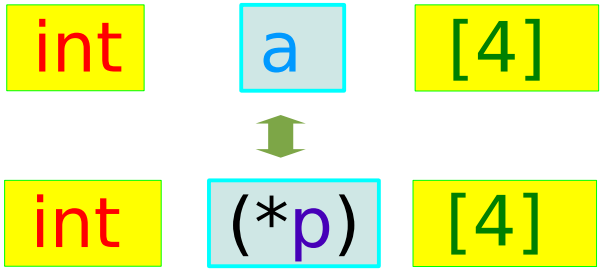
2-D Array Dynamic Memory Allocation (2)

```
int **  d ;

d = (int **) malloc (4 * size of (int *));
for (i=0; i<4; ++i)
  d[i] = (int *) malloc(4 * sizeof(int));
```

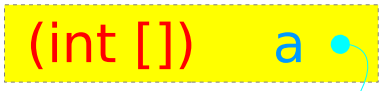


Pointer to array (1)

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



pointer to the array of 4 elements



(int) a[0]
(int) a[1]
(int) a[2]
(int) a[3]

```
int  m ;          an integer variable
int  *n ;         a pointer variable

int func (int  a,  int  b) ;      a prototype
int (* fp) (int  a,  int  b) ;    a function's type

int * fp (int  a,  int  b) ;      function pointer
```

Pointer to array (2)

```
int (*p) [4] ;
```

```
int a [4]
```

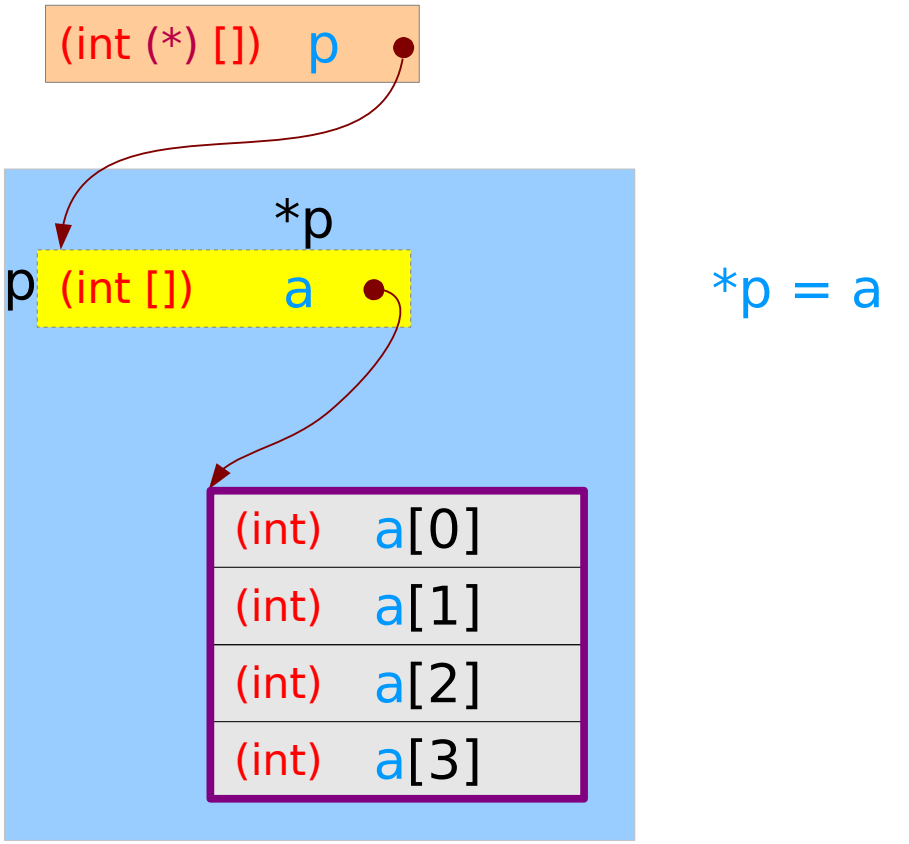
(*p) = a

&(*p) = &a

p = &a

sizeof(p)= 4 bytes

sizeof(*p)= 16 bytes



*p = a

an array with 4 integer elements

Pointer to array (3)

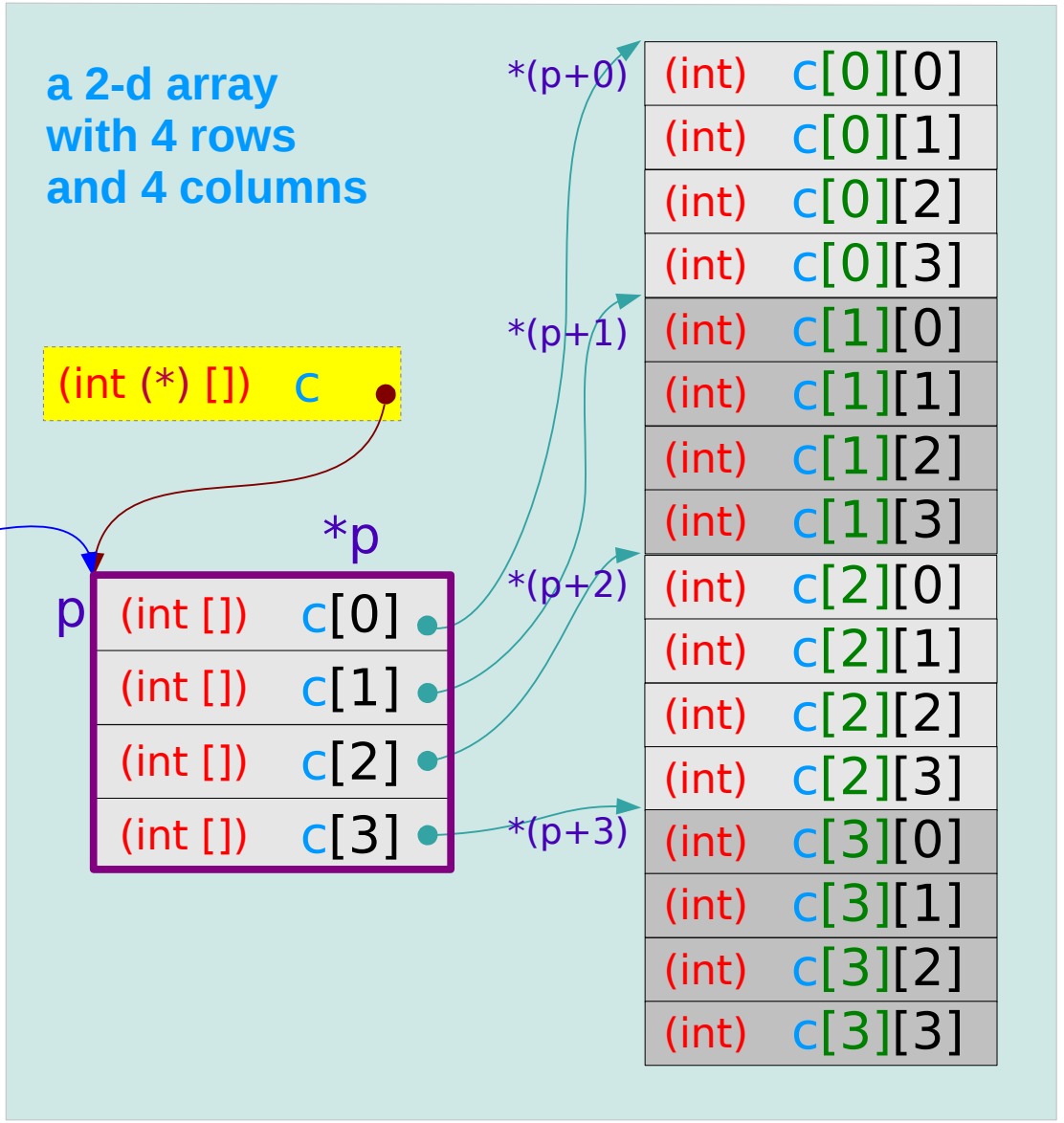
```
int (*p) [4] ;
```

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

&p (int (*) []) p

p = c

(*p) [i][j];



Pointer to array (4)

```
int c [4][4];
int (*p) [4];

p = c;

func(p, ... );
```

```
void func(int (*x)[4], ... )
{

    x[r][c] =

}
```

```
void func(int x[ ][4], ... )
{

    x[r][c] =

}
```

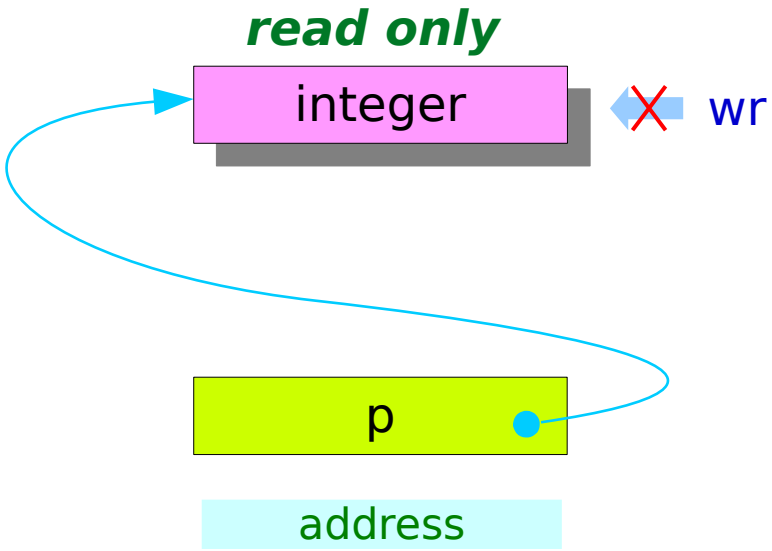
const type, const pointer type (1)

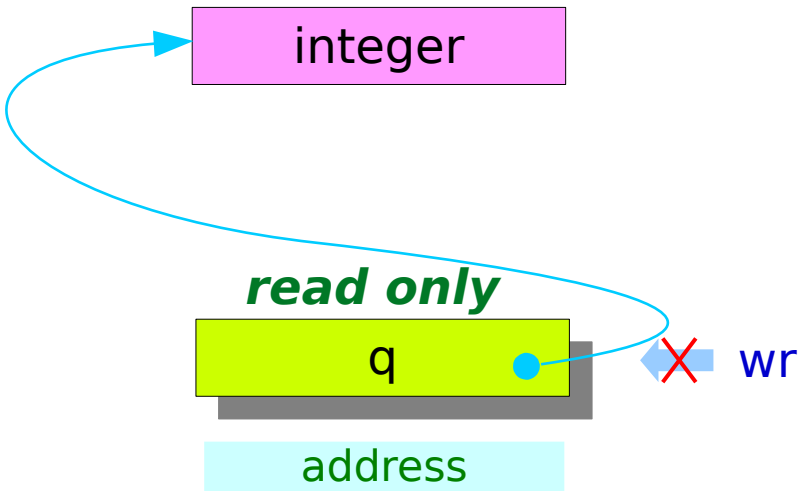
```
const int * p;

int * const q ;

const int * const r ;
```

const type, const pointer type (2)

const int * **p;**

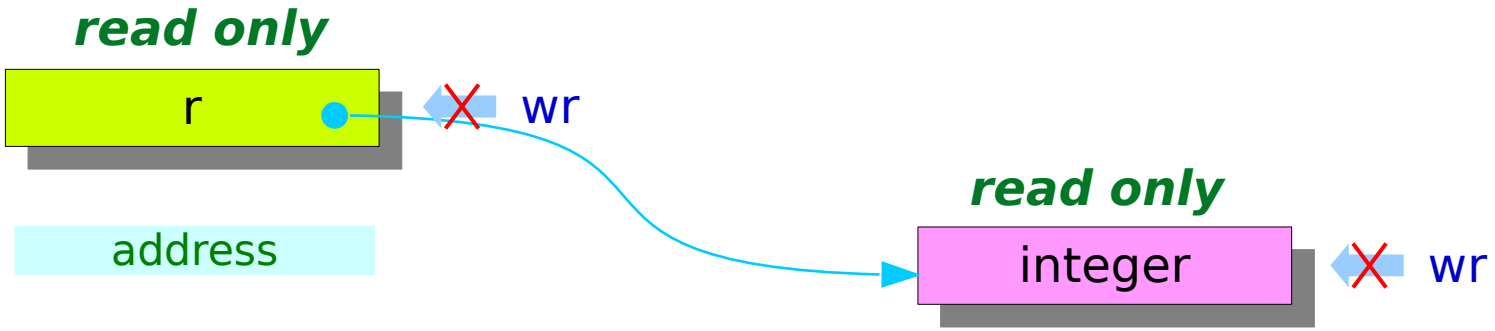
int * **const q** **;**



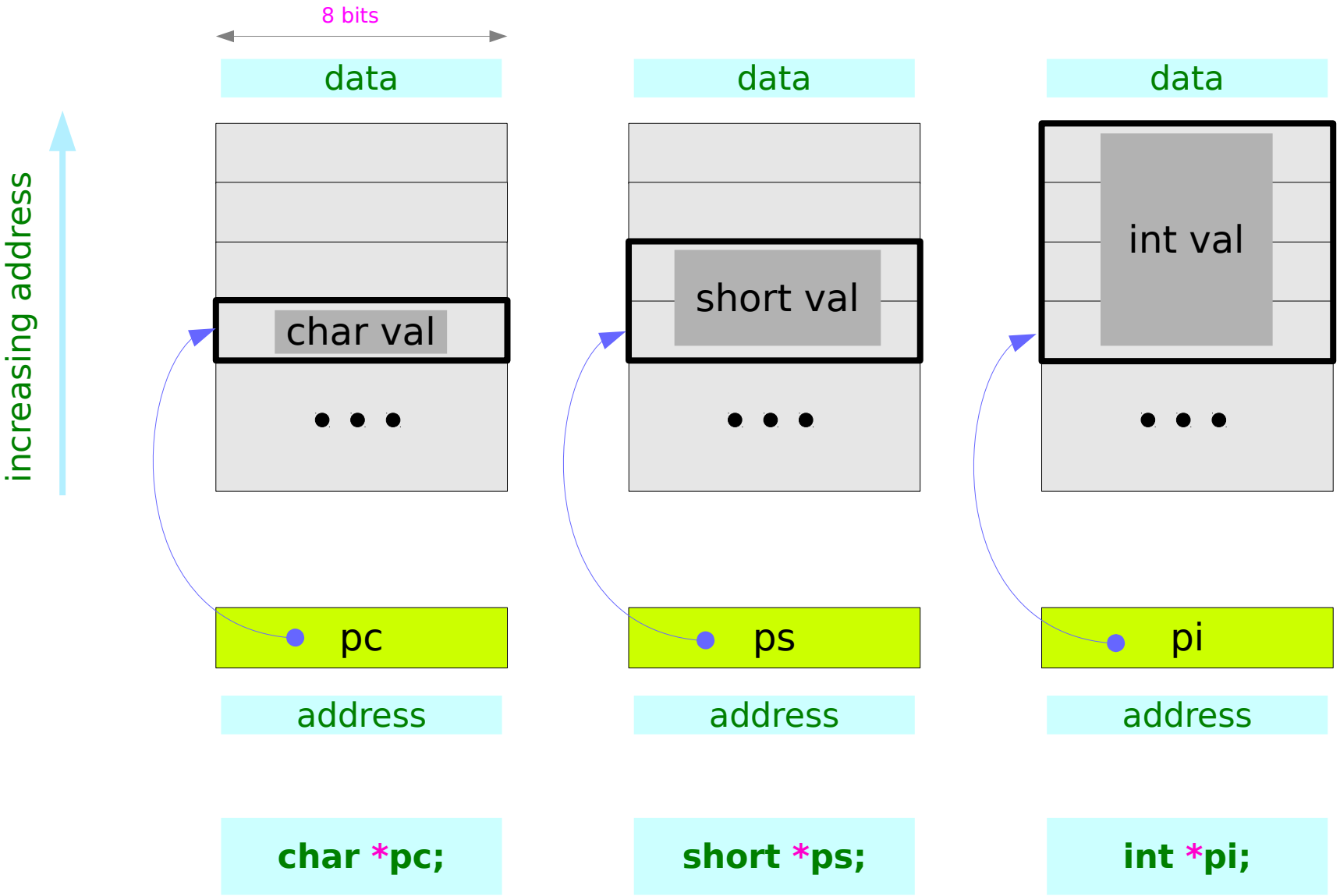


const type, const pointer type (3)

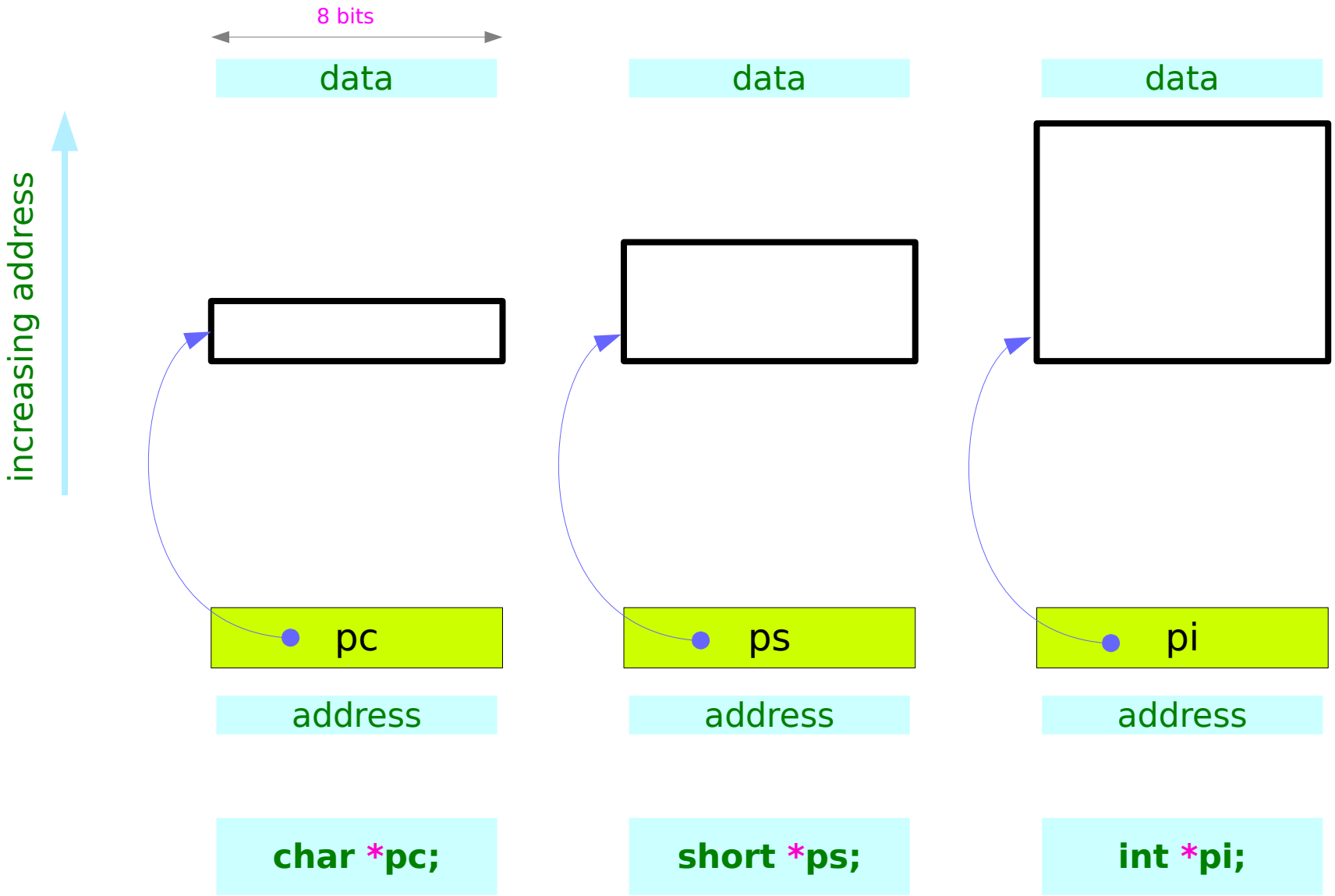
```
const int * const r ;
```



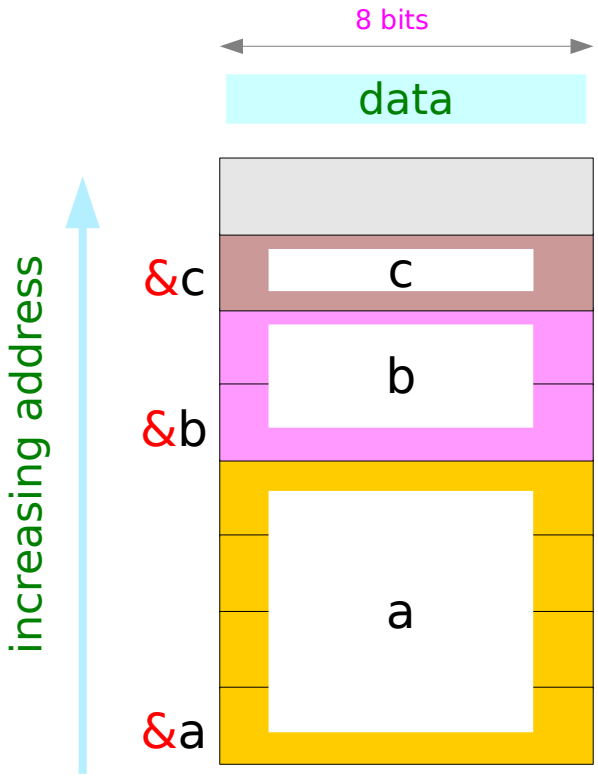
Pointer Types and Associated Data



Pointer Types

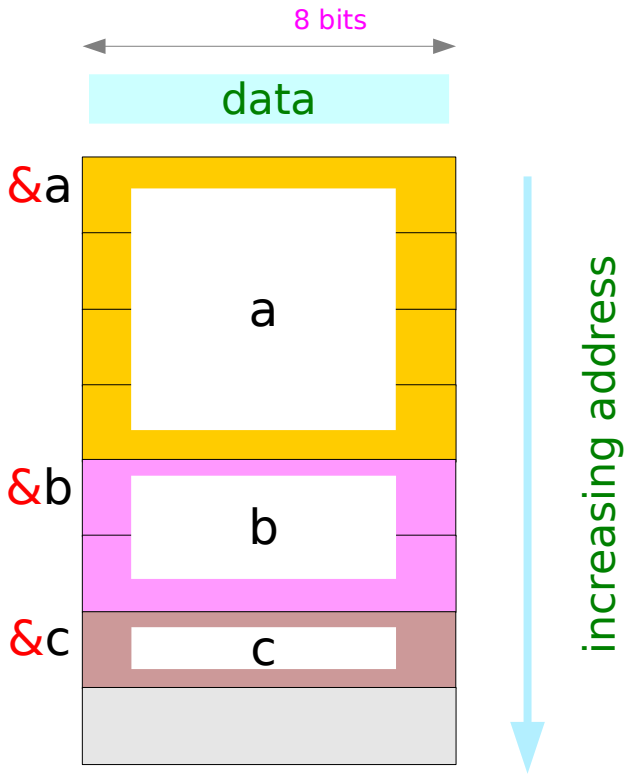


Little Endian Example

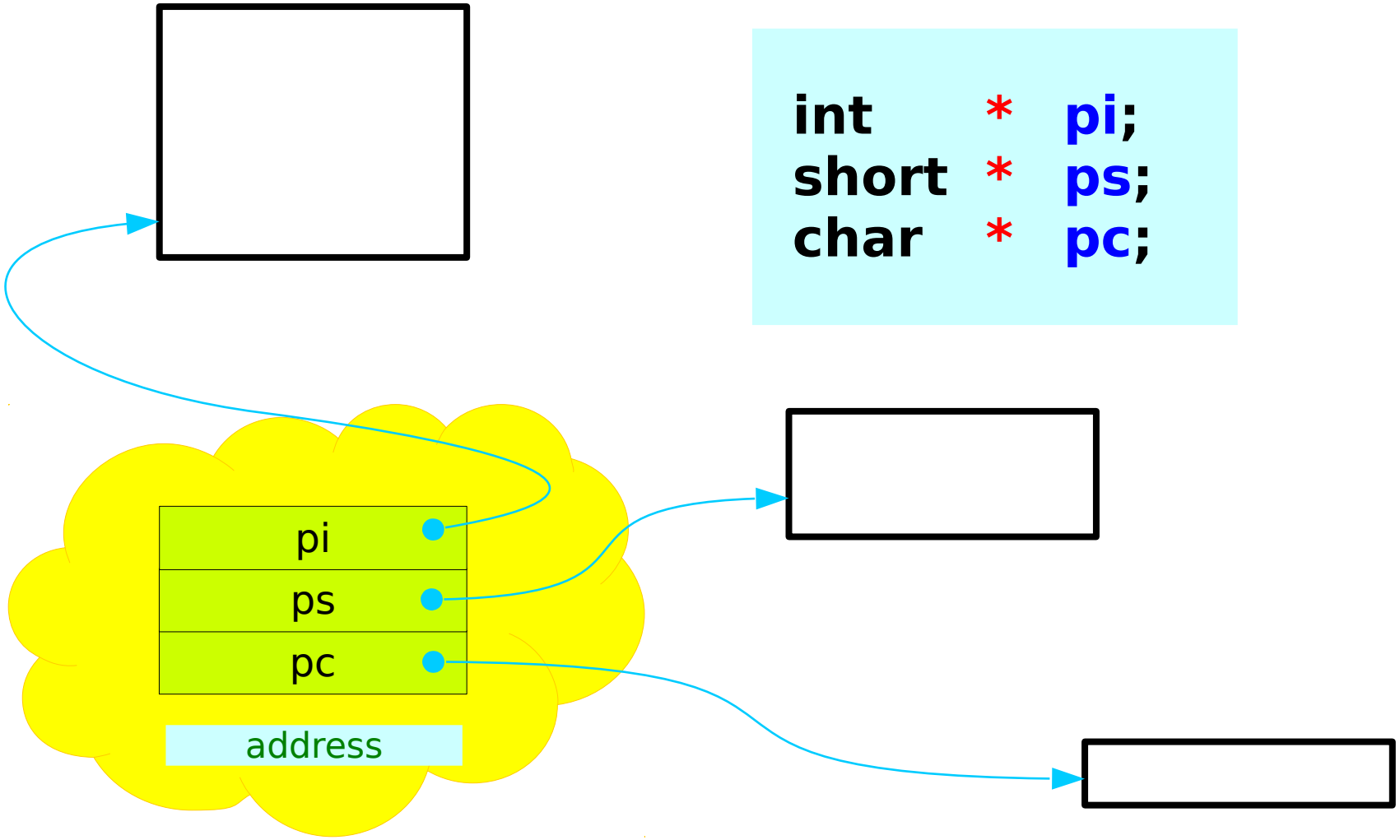



```
int   a;
short b;
char  c;
```

the order of definition

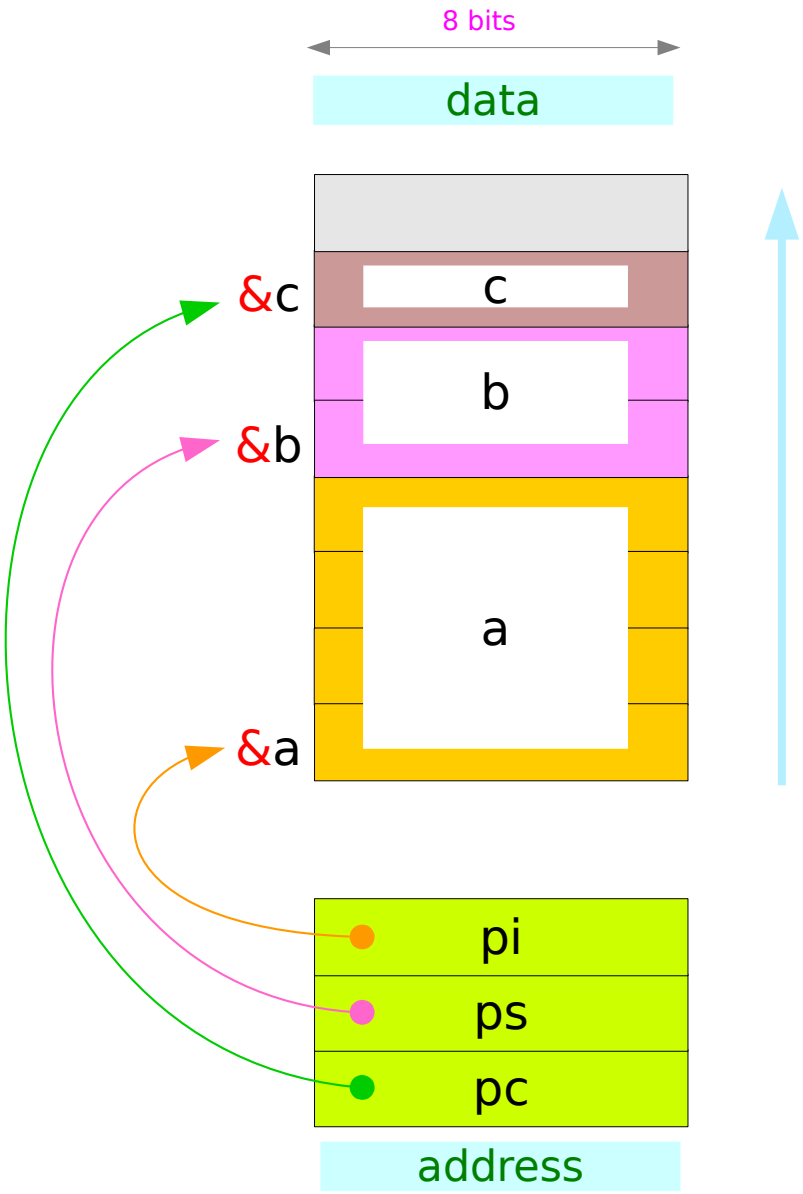


int *, short *, char * type variables



Not a sized representation

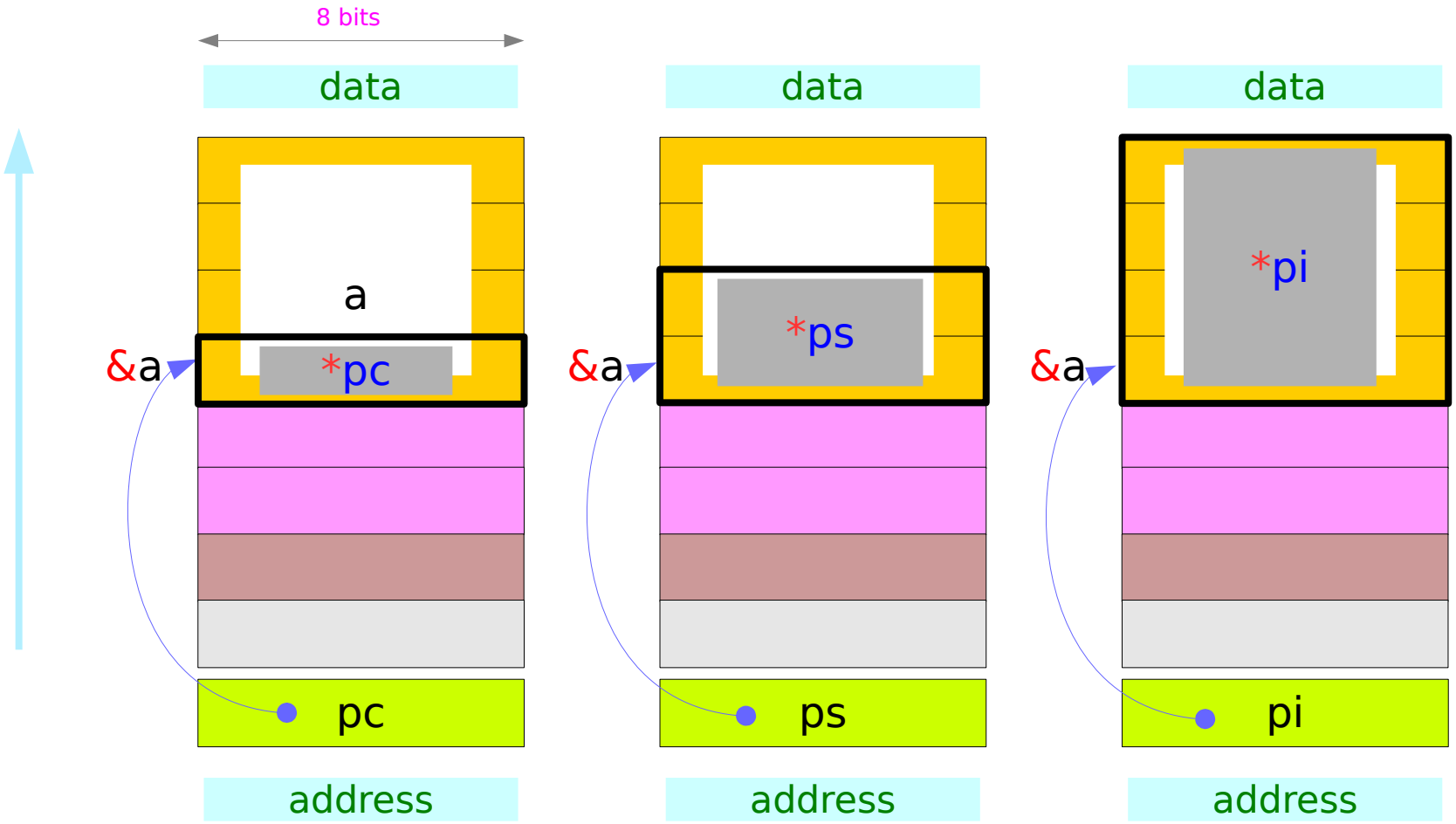
Pointer Variable Assignment




```
char   *  pc;
short  *  ps;
int    *  pi;

int    a;
short  b;
char   c;
```

```
pi = &a;
ps = &b;
pc = &c;
```

Pointer Type Casting

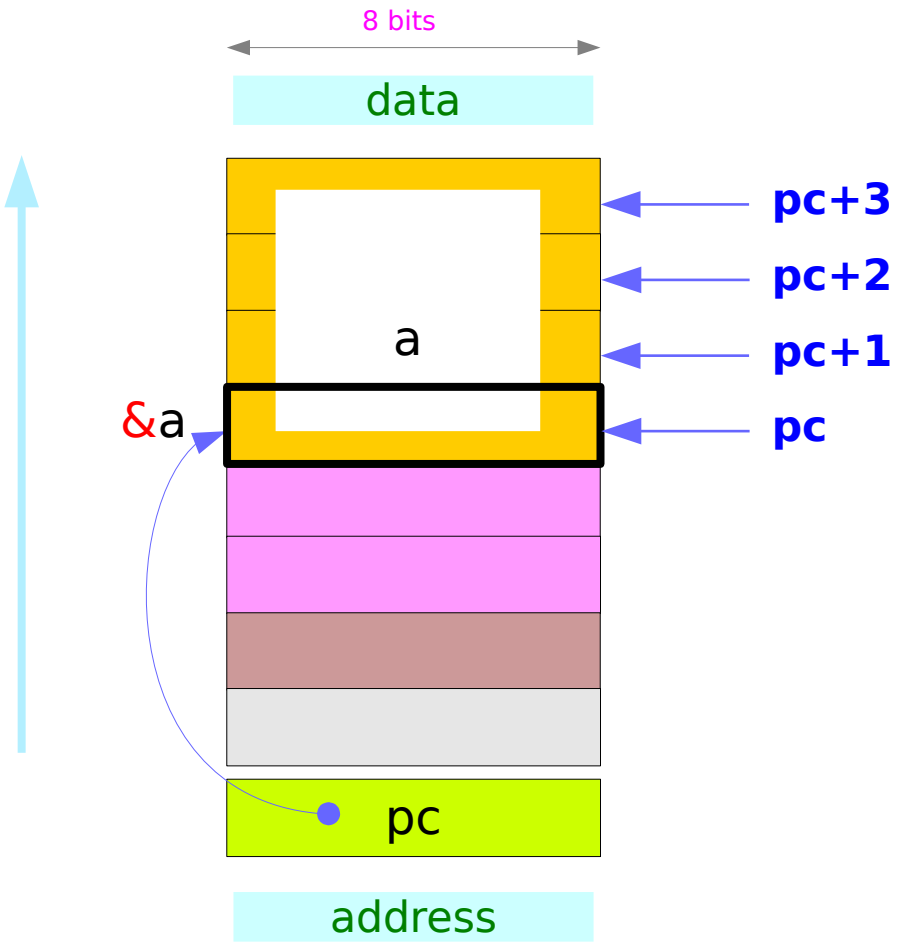


char *pc;
pc = (char *) &a

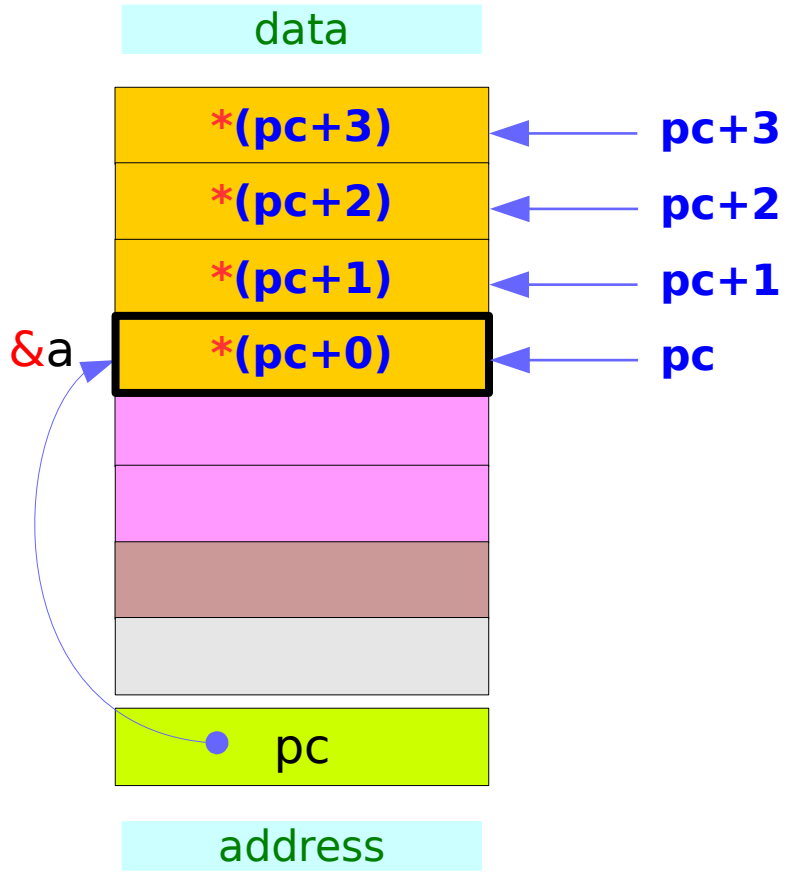
short *ps;
ps = (short *) &a

int *pi;
pi = (int *) &a

Accessing bytes of a variable



char *pc;
pc = (char *) &a

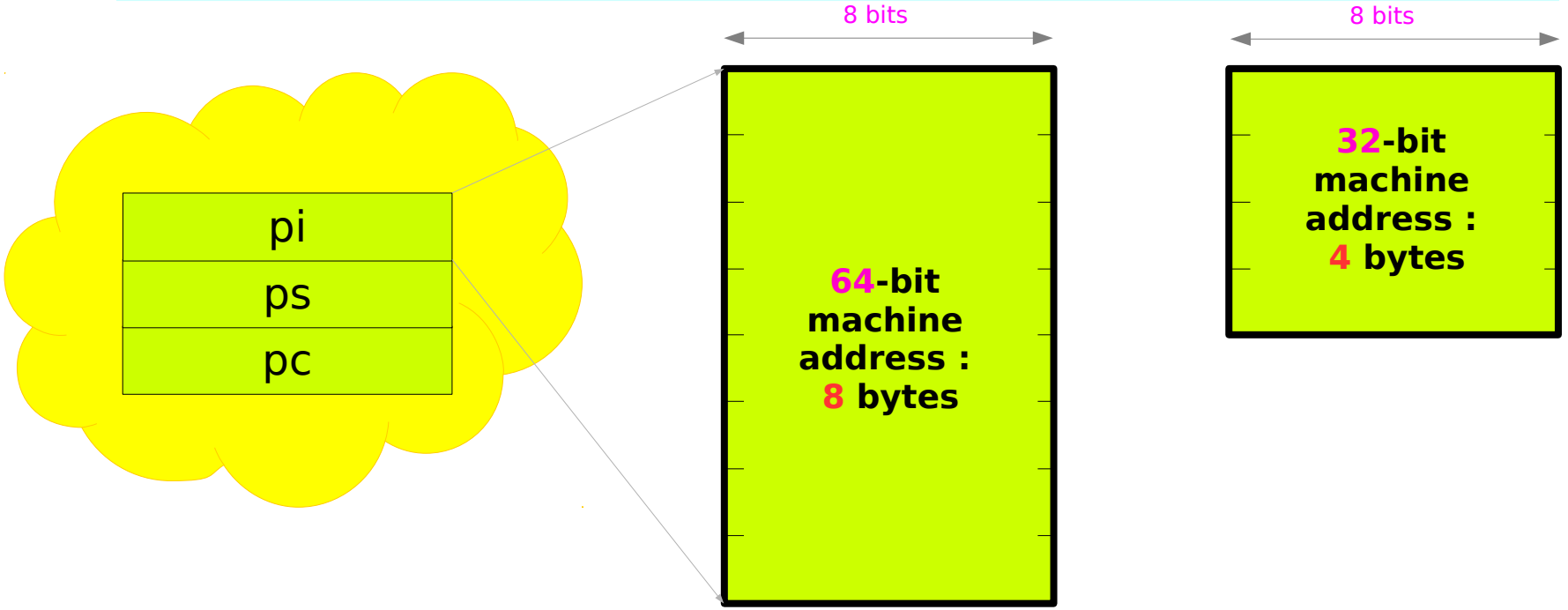


char *pc;
pc = (char *) &a

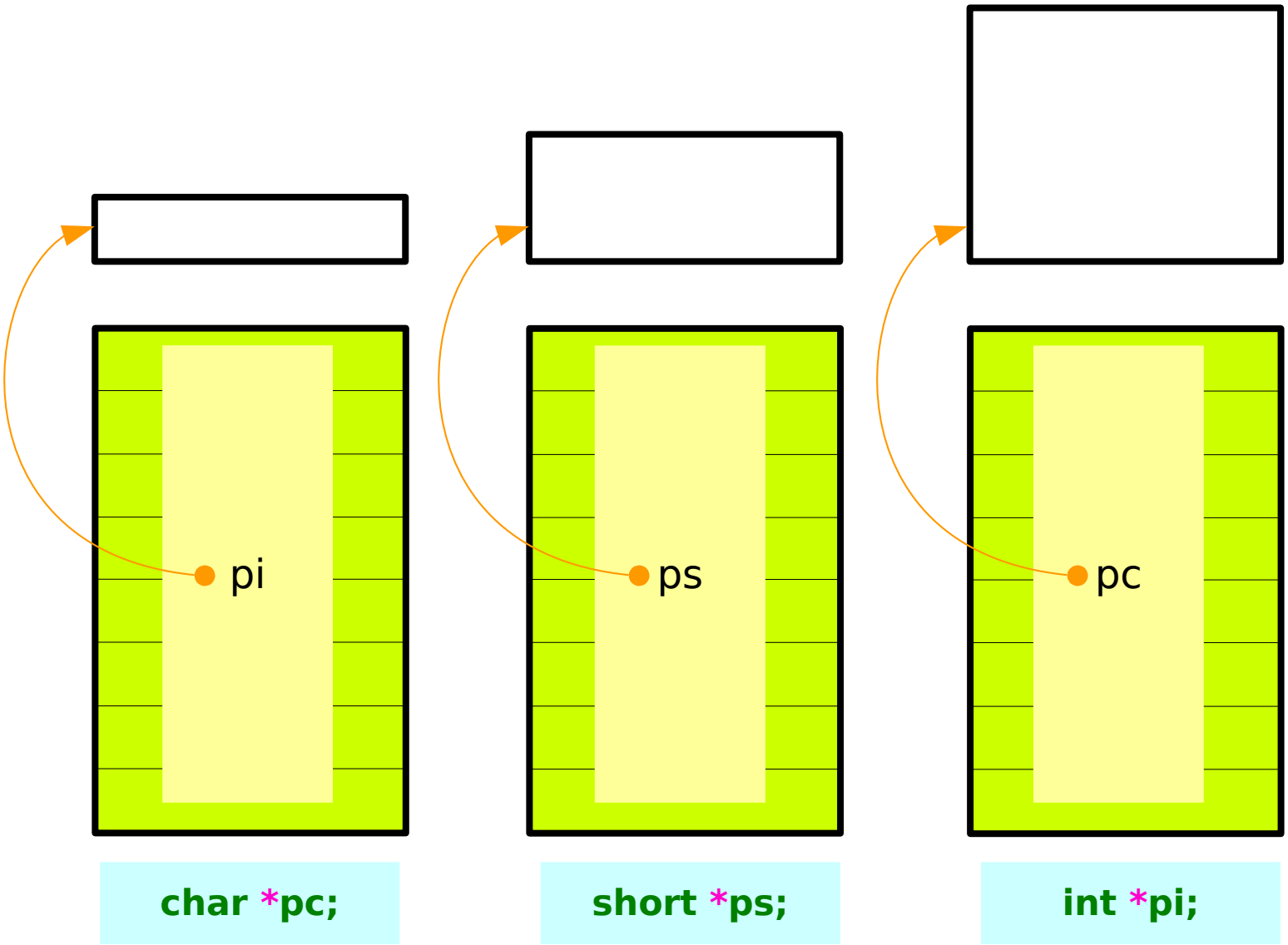
32-bit and 64-bit Address

32-bit machine : address : 4 bytes

64-bit machine : address : 8 bytes

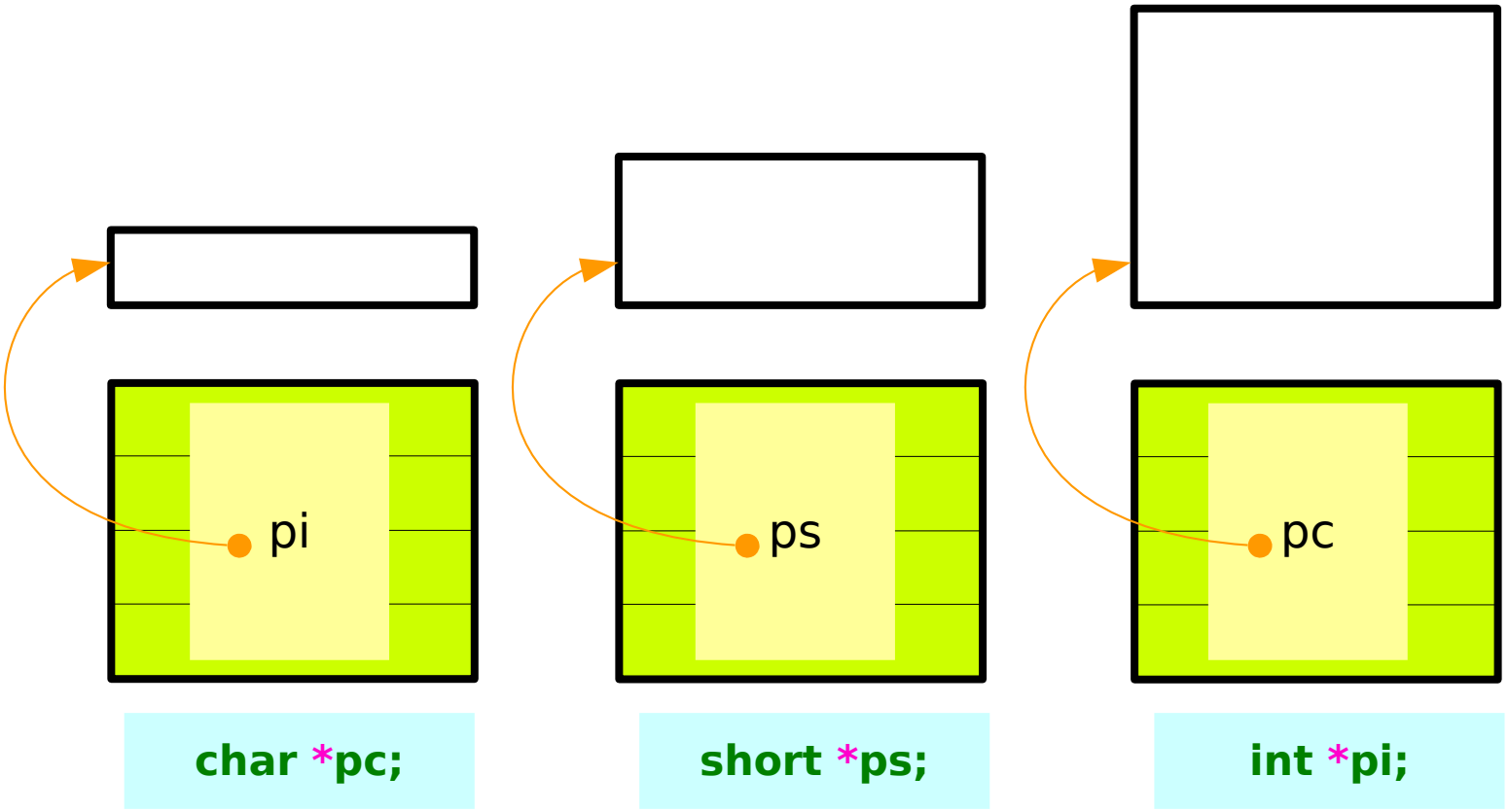


64-bit machine : 8-byte address

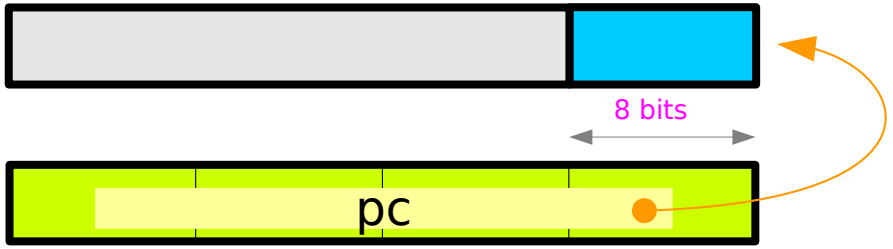


64-bit machine : 8-byte address & data buses

32-bit machine : 4-byte address



64-bit machine : 8-byte address and data buses



char *pc;

short *ps;

int *pi;

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