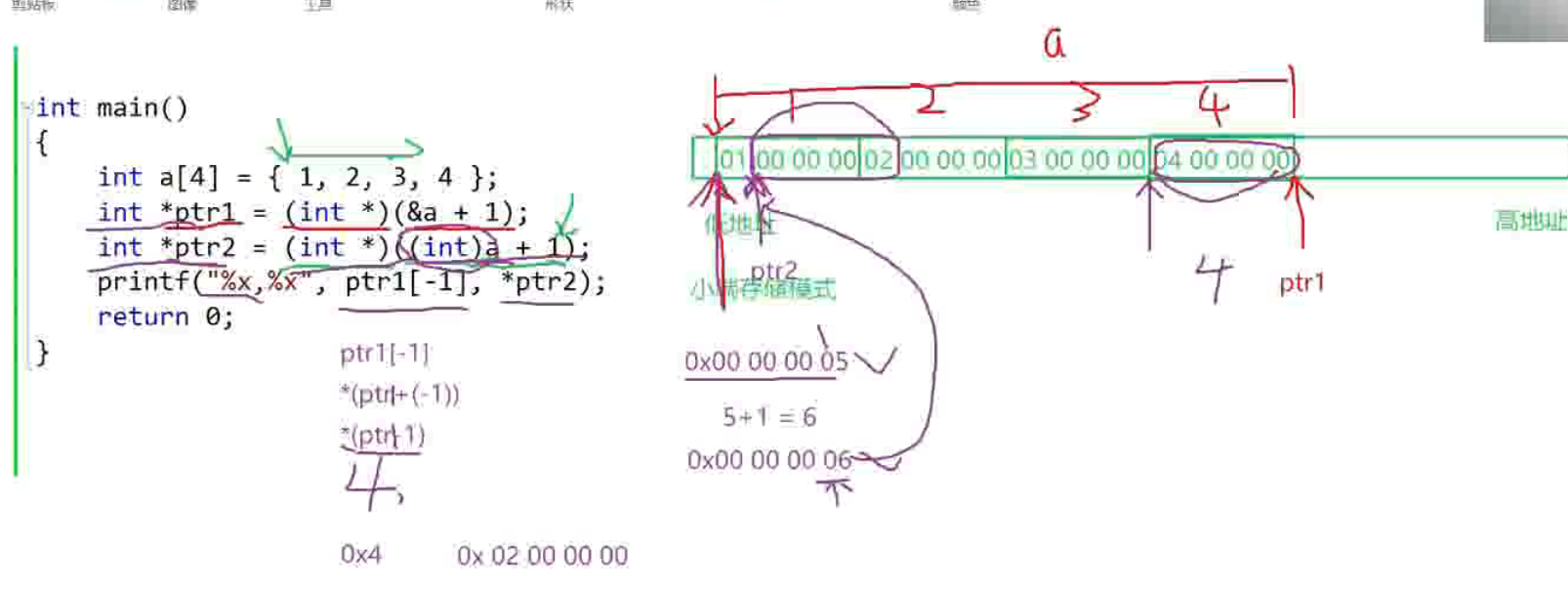
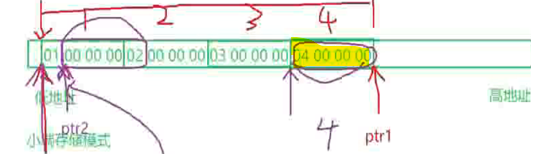
1.



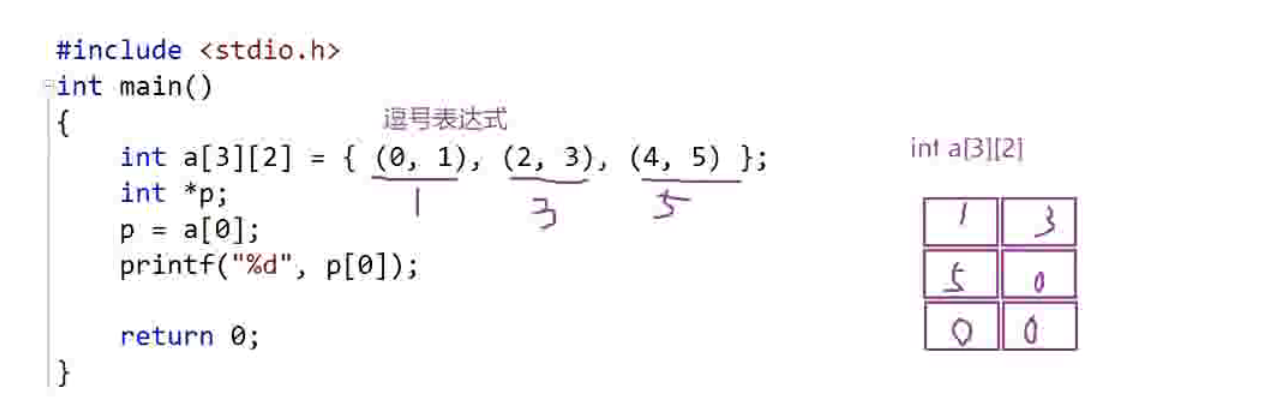
Int \*prt2=(int \*)((int)a+1);//（int）a表示将数组a的首元素地址强制类型转化成int型将一堆十六进制（例0x00000005）地址转化成int整型 (5) 然后+1(6) 再强制类型转化成int\*整型指针(0x00000006)此时只和首元素地址差一个字节 ptr2又是个整型指针所以\*ptr2时输出0x02000000(因为小端倒着存)



Int \*ptr1=(int \*)(&a+1);//ptr1[-1]==\*(ptr1-1) 所以ptr1[-1]输出4 (0x00000004)

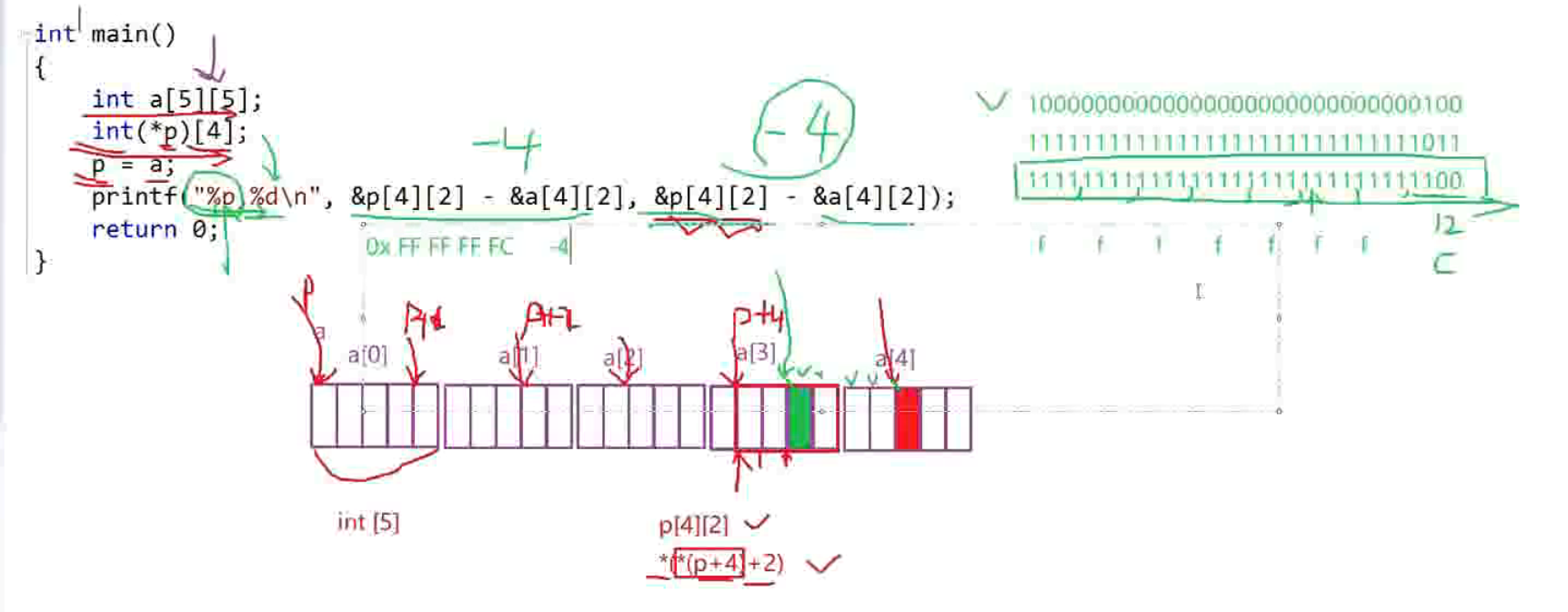


2.



数组在初始化的时候用的小括号（）并且用逗号隔开 所以是逗号表达式 逗号表达式的结果是最后一个式子的结果 所以数组实际只存了1 3 5

P[0]表示\*(p+0)

3. 

&a[4][2]就是正常的取第四行第二列元素的地址

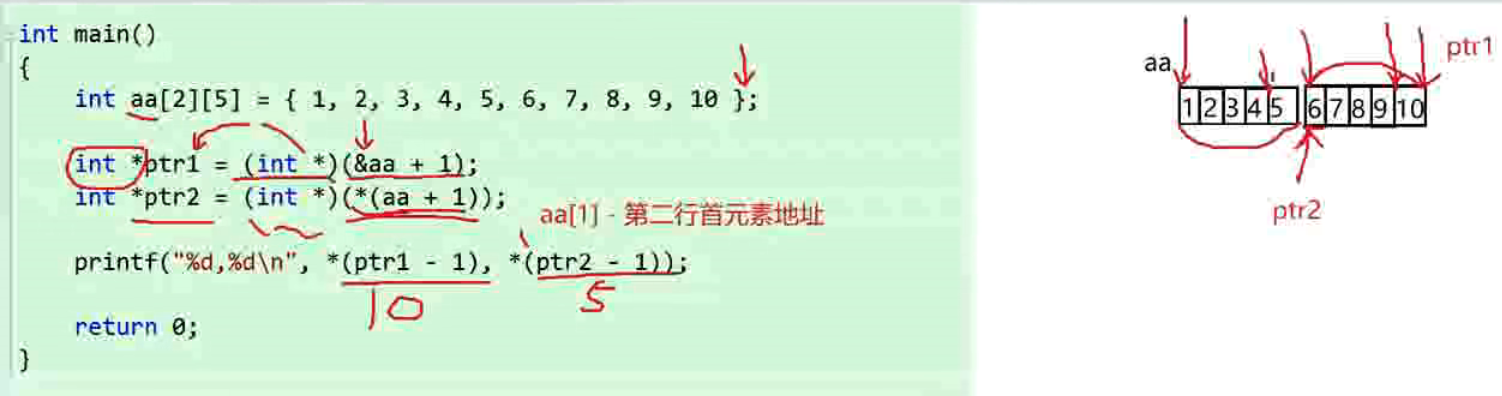
Int(\*p)[4];p=a;表示将数组a的首元素地址赋给指针数组p（该指针数组有四个变量）

&p[4][2]==\*(\*(p+4)+2), \*(p+4)表示的是对第p+4数组解引用得到第p+4数组的首元素地址

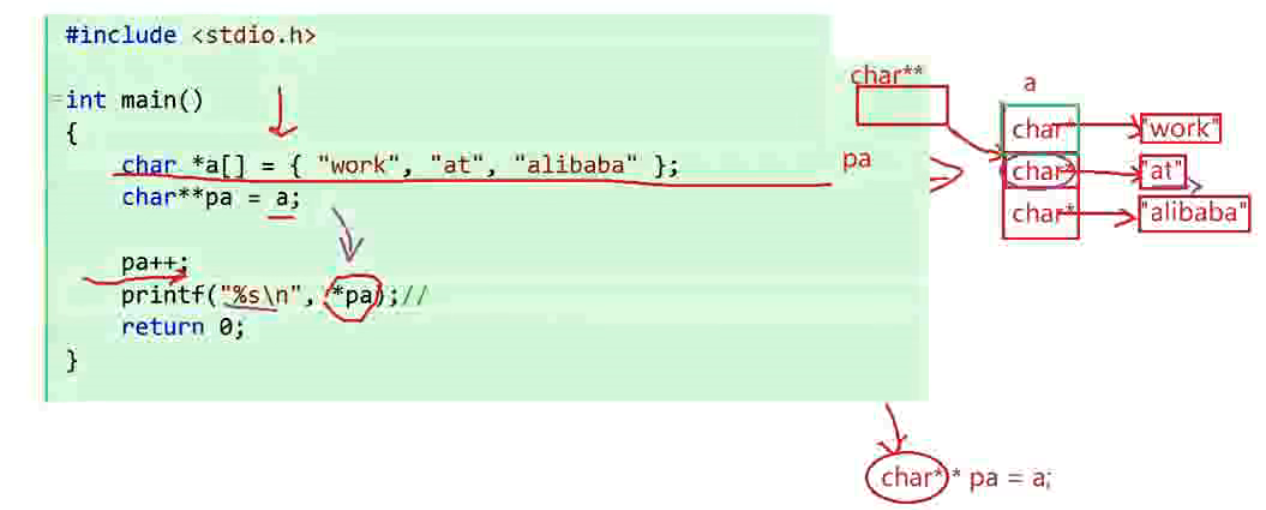
\*(\*(p+4)+2)表示对第p+4数组首元素地址+2再解引用

%d输出：最后指针-指针既两个地址相减得到的是两个指针之间的元素个数 ：-4

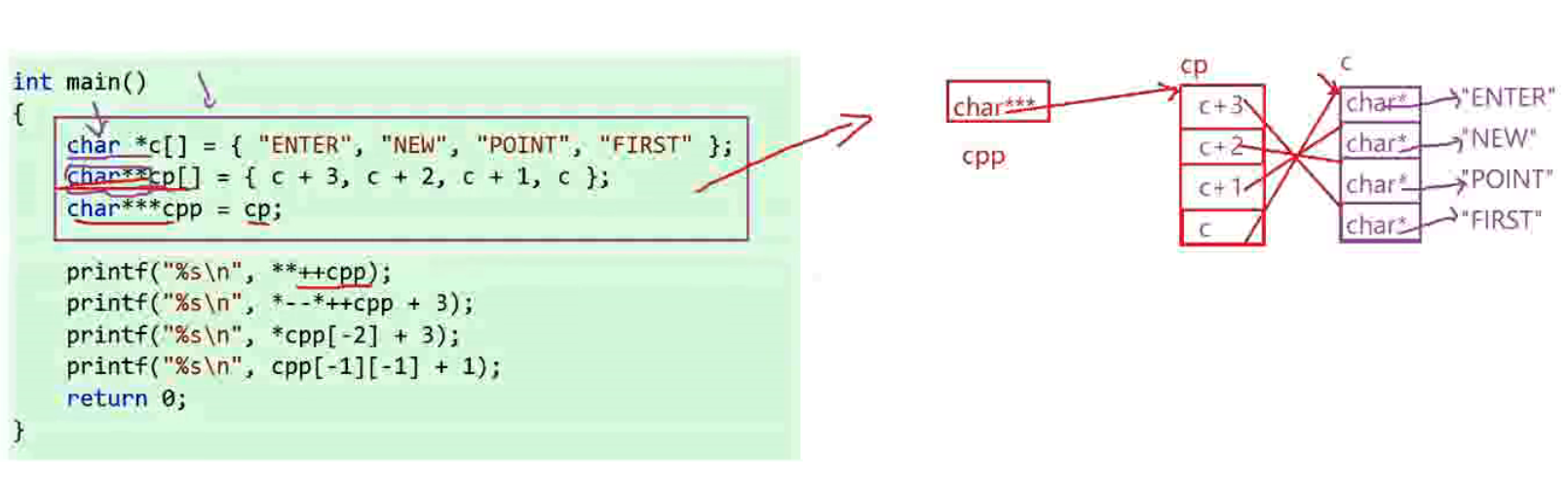
%p输出：%p格式符会输出 ***指针本身的值***，也就是指针指向的地址值。 该输出为16进制形式，具体输出值取决于指针指向的实际地址值 –-**指针指向实际地址值为-4，我们认为-4是原码将-4转换成补码后再转化成16进制**

4. ****

**5.输出at**

****

6.输出POINT ER ST EW

****

char\* 类型的地址为char\*\* char\*\*的地址为char\*\*\*

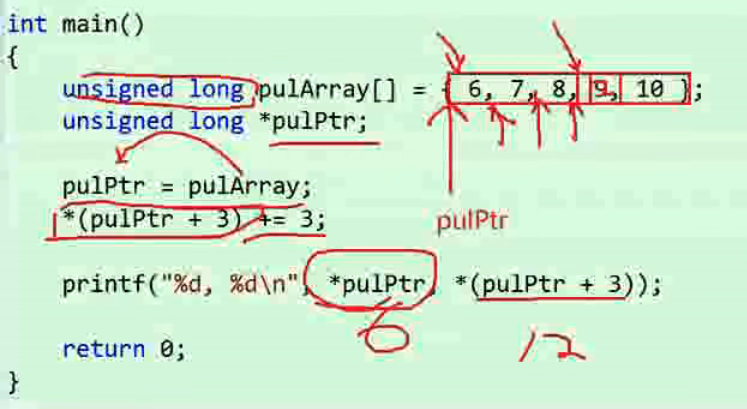
++cpp指向cp中的c+2 \*(++cpp)拿到c+2所指向的c中的POINT的首字符p的地址 \*(\*(++cpp))就是拿到POINT首字符地址内的内容 最后%s以字符串输出就是POINT

按照优先级先++cpp 然后 \*(++cpp) 然后--(\*(++cpp)) 然后\*(--(\*(++cpp))) 最后+3

Cpp[-2]==\*(cpp-2)

Cpp[-1][-1]==\*(\*(cpp-1)-1)

**7.**

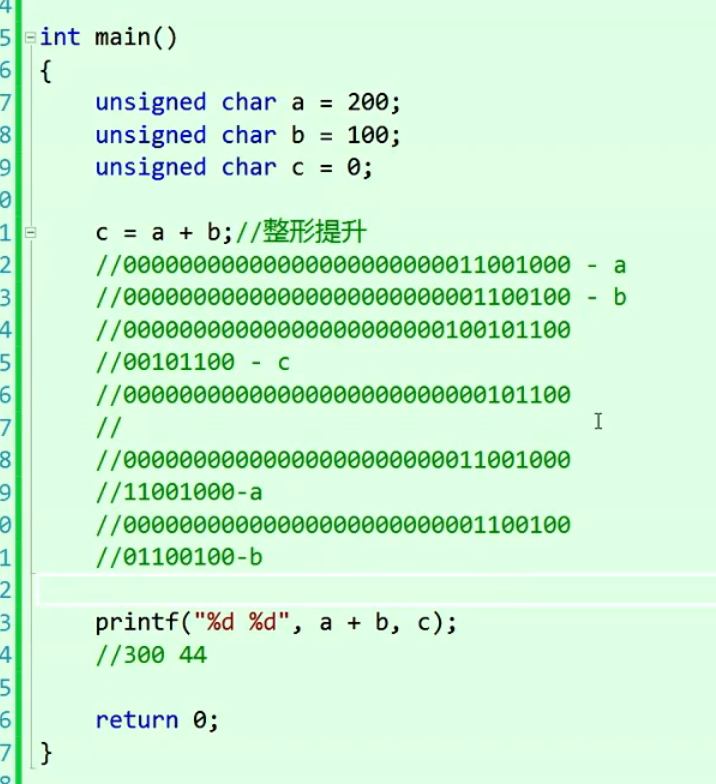


**8.**%d输出有符号的十进制

直接输出%d a+b是输出的十进制数 直接a整型提升+b整型提升后转化成十进制

计算c时 是将a整型提升+b整型提升后存入unsigned char c中char只取后8个字节

%d 输出c时再将c整形提升后的二进制数转化成有符号的十进制



9.char类型的取值范围-128—>127

