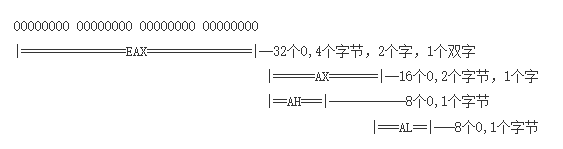
**实验收获和总结**

**冒泡排序是一种较为简单的排序算法，需要使用嵌套循环。每一个外循环会将未排序数据中的最大值排到末尾，每一个小循环会将相邻两个数比较大小，从而使较大的数下沉，较小的数上浮。在这次实验中收获了如下知识：**

1. **本题需要使用条件转移指令，比较指令（CMP），交换指令（XCHG）。CMP和XCHG的两个操作数不能同时为内存中的数据，但可以一个是寄存器，一个是内存数据。所以，在比较和交换数据的时候，需要将其中一个内存数据放到寄存器中，从而完成两个元素的交换。**

**2.数组的定义：在data段对数组进行定义时可以直接给出整数数组的值，比如array DWORD 100, 101, 102, 103，也可以先用DUP伪指令为数组赋初值，比如array BYTE 20 DUP（10）；同时，字符串也可以当作字符的数组，在给字符串进行定义的时候要注意如果不是用DUP伪指令则需要在结尾加0。对于除了字符串利用末尾0来判断结束以外，其他数组可以在创建时，建立一个内存sum来存储数组的长度，每存储一个元素加四或者加一（根据元素类型决定），需要访问数组尾时，便可以直接访问array[（sum-1）\*4]；或者利用sum赋值给ecx，通过loop来遍历。**

**3.数组的操作：在内存中，数组可存在于栈，数据段及动态内存中，其寻址用“基址+偏移量”实现，其中偏移量通过数组的数据类型来决定。如利用变址寄存器或者变址寄存器加上立即数（变址操作数），操作内存：array[ecx\*TYPE array]。同时要注意mov指令不允许内存之间进行操作，并且数组元素类型的大小和要放入的寄存器要匹配，如果不匹配可以用寄存器的一部分，如低位部分或者高位部分来匹配数组大小，以eax为例，其关系如下图。**