

### 1. 显示过程参数

写一个名称为 **ShowParams** 的过程，使其能够显示调用 **ShowParams** 的过程的运行时栈 (run-time stack) 中 32 位参数的地址和十六进制数值。

例如，假设我们有通过以下语句调用了 **MySample**，并且传递了三个参数：

```
INVOKE MySample, 1234h, 5000h, 6543h
```

**MySample** 内部用以下方法调用了 **ShowParams**：

```
MySample PROC first:DWORD, second:DWORD, third:DWORD
    paramCount = 3
    INVOKE ShowParams, paramCount
```

**ShowParams** 应当以以下格式进行输出：

```
Stack parameters:
-----
Address 0012FF80 = 00001234
Address 0012FF84 = 00005000
Address 0012FF88 = 00006543
```

提交的代码中，应当包含 **ShowParams** 过程及上述调用 **ShowParams** 的示例代码。

### 2. 字符串连接

写一个名为 **Str\_concat** 的过程，把源字符串 (sourceStr) 连接在目标字符串 (targetStr) 之后 (调用该过程时，会保证目标字符串中有足够的空间容纳源字符串。实现 **Str\_concat** 时不能调用 **sprintf** 等字符串处理的系统库函数)。传递的参数是指向源字符串和目标字符串的指针。下面是一个可能的调用示例：

```
.data
targetStr BYTE "ABCDE",10 DUP(0)
sourceStr BYTE "FGH",0
.code
INVOKE Str_concat, ADDR targetStr, ADDR sourceStr
```

提交的代码中，应当包含 **Str\_concat** 过程及调用 **Str\_concat** 的上述示例代码，要求按顺序分别输出源字符串、目的字符串和连接得到的字符串结果，每个占一行，无需输出多余的信息。

### 3. 数组低八位冒泡排序

用汇编程序设计语言实现针对 **WORD 数组** 的 **BubbleSort** 过程，将该 **WORD 数组** 按照**每个元素低八位的大小**从小到大的顺序进行排序 (即：将数组中每个元素的低 8 位取出，当作一个**无符号整数**，作为数组原始元素大小的判断标准)。要求在 **BubbleSort** 过程中存在一个变量，在内层循环中如果一对值进行了交换，则把该变量置 1，如果在某一遍处理后发现没有交换任何数值，则退出排序过程。调用 **BubbleSort** 时会传入数组的地址与长度作为参数。

提交的代码中，应当包含 **BubbleSort** 过程及调用 **BubbleSort** 的代码，其中包含两组不同的测试数据 (如一组可能的测试数据为：myArray WORD 3, 1, 7, 5, 2, 9, 4, 3)。需要输出每组测试数据排序前的内容和排序后的结果，每个占一行，行内用空格隔开。

将 1~3 题的源代码文件 (.asm) 和可执行程序 (.exe) 打成压缩包提交，每个文件以题目序号命名 (1.asm、1.exe、2.asm……)，压缩包命名为“学号\_姓名\_汇编 4.rar”或“学号\_姓名\_汇编 4.zip”。