

# 汇编语言程序设计

## 习题课2

助教: 李灯举([hello@lidengju.com](mailto:hello@lidengju.com))

## 大纲

- ▶ 第六次作业
- ▶ 第七次作业

对每次作业分为「知识点」和「错题讲解」

## 第六次作业—知识点

- ▶ 分支与循环
- ▶ 包括无条件、条件转移指令、循环控制指令
- ▶ 以上为基础进行分支与循环结构的程序设计
- ▶ 循环结构是分支结构的一种**特例**
- ▶ 实现分支结构的**核心原理**在于，能够**根据标志位状态**决定程序的一个流程。
- ▶ 指令对这些标志位**只读**

## 第六次作业—题目3

- 按要求分析下列程序的执行结果

```
MOV  AL, -5
MOV  BL, -3
CMP  AL, BL      ; Compare AL和BL大小
        JGE  L1      ; 大于等于?
        JMP  L2      ; 否, 顺序执行, 到L2
L1:    MOV  AH, 00H
        JMP  L3
L2:    MOV  AH, 01H  ; 传送
L3:    .....
```

- (AH) =? → 01H

# 第六次作业—题目5

## 分析并说明程序片段功能

```
*****
ARY1 DB 35, 23, -6, -21, 90, 65, 73, -54, -7
LEN EQU $ - ARY1 ; ARY1长度
*****

LEA BX, ARY1 ; 取有效地址
XOR AL, AL ; 统计计数器清零操作
MOV CX, LEN ; 初始化循环计数器
JCXZ EX1 ; 通过CX=0来转移

L1: TEST [BX], 01H ; [BX]最低位是否为1?
JNZ L2 ; 如果ZF不为零
JMP L3 ; 顺序执行

L2: INC AL ; 增加计数
L3: INC BX ; BX指向数组下一个字节
LOOP L1
EX1: *****
```

程序功能: 统计补码数组ARY1中奇数的数量（也可解释为统计ARY1中最低位为1的数据个数），统计结果保存在AL寄存器中。

# 第六次作业—题目9

- ▶ 假设有两个长度为8个字的补码，分别保存在字类型数组DA1和DA2中，低地址保存低位，试设计一个程序实现两个长补码的减法，减法结果保存在字类型数组RES1中

```
DATA SEGMENT                                ; 定义数据段
DA1 DW 0143H, 435AH, FA1CH, 7754H, 6A11H, B2FFH, 1B2BH, 0FF1H
DA2 DW E227H, A82BH, 33BAH, 72B6H, 1F21H, 2614H, 4FC1H, 2CBAH
LEN EQU 8
RES1 DW LEN DUP(?)
DATA ENDS
```

```
CODE SEGMENT
BEGIN: MOV AX, DATA
      MOV DS, AX
      MOV CX, LEN          ; 定义循环长度
      XOR BX, BX           ; 指针清零
      CLC                  ; 进位标志位清除
LOP1: MOV AX, DA1[BX]      ; 使用AX寄存器16位, 存储一个字
      SBB AX, DA2[BX]      ; 带进位的减法
      MOV RES1[BX], AX    ; 保存结果
      ADD BX, 2           ; 指针位移
      LOOP LOP1           ; 根据CX寄存器循环, CX = 0?
      MOV AH, 4CH         ; 程序结束, 返回DOS
      INT 21H
CODE ENDS
END BEGIN
```

第六次作业—题目10

- 假设有一个长度为4个字的无符号数，保存在字类型数组DA1中，低地址保存低位，试设计一个程序实现这个长无符号数除以8的功能，运算结果仍然保存在数组DA1中。

```
DATA SEGMENT                                ; 定义数据段
DA1 DW 7B24H, F2BBH, 114FH, 6A82H
LEN EQU $-DA1
DATA ENDS

CODE SEGMENT
BEGIN:  MOV  AX, DATA
        MOV  DS, AX
        MOV  CH, 3                          ; 外层循环，控制log8 = 3
LOP1:   MOV  CL, 3                          ; 内层循环，控制数组指针
        MOV  BX, LEN-2                     ; 指针指向最后一个元素
        SHR  DA1[BX], 1                    ; 右移1位，除以2
LOP2:   SUB  BX, 2                          ; 指针指向低1位元素
        RCR  DA1[BX], 1                    ; 使用带进位的循环右移，1位
        DEC  CL                            ; 内循环计数自减
        JNZ  LOP2                          ; 执行3次
        DEC  CH
        JNZ  LOP1                          ; 外循环
        MOV  AH, 4CH
        INT  21H

CODE ENDS
END BEGIN
```



## 第七次作业—知识点

- ▶ 子程序调用指令和返回指令
- ▶ 调用指令(段内, 段间, 直接, 间接)
- ▶ 返回指令(段内, 段间, 有参, 无参)
- ▶ 子程序的设计
  - 框架(子程序名称, **PROC, ENDP, RET**)
  - 独立性和通用性
  - 参数传递(寄存器传递[**SI, DI**], 堆栈传递方式, 数据区传递方式)
  - 现场保护(调用子程序时, 有关寄存器、标志位当时的状态被称为**CPU现场**。)
- ▶ 系统调用



第七次作业—题目5

- ▶ 设计子程序，实现：
- ▶ 接收主程序传递的待处理的数据地址，待处理数据类型，位编号等入口参数。
- ▶ 将待处理数据中指定编号的二进制位清零。
- ▶ 其他数据位不受影响。

利用堆栈传递参数，子程序堆栈



第七次作业—题目5

- 设计子程序，实现：
- 接收主程序传递的待处理的数据地址，待处理数据类型，位编号等入口参数。
- 将待处理数据中指定编号的二进制位清零。
- 其他数据位不受影响。

```
CLEAN_BIT PROC
    PUSH BP                ; 将BP作为现场进行保护
    MOV BP, SP             ; 将BP指向当前栈顶
    PUSH AX                ; 现场保护
    PUSH BX
    PUSH CX
    PUSH DX
    MOV BX, [BP+8]          ; 取数据地址
    MOV DX, [BP+6]          ; 取待处理数据类型
    MOV CX, [BP+4]          ; 取数据位编号
    CMP DX, 8               ; 按照数据类型获取待处理数据
    JZ L8bit
    JMP L16bit

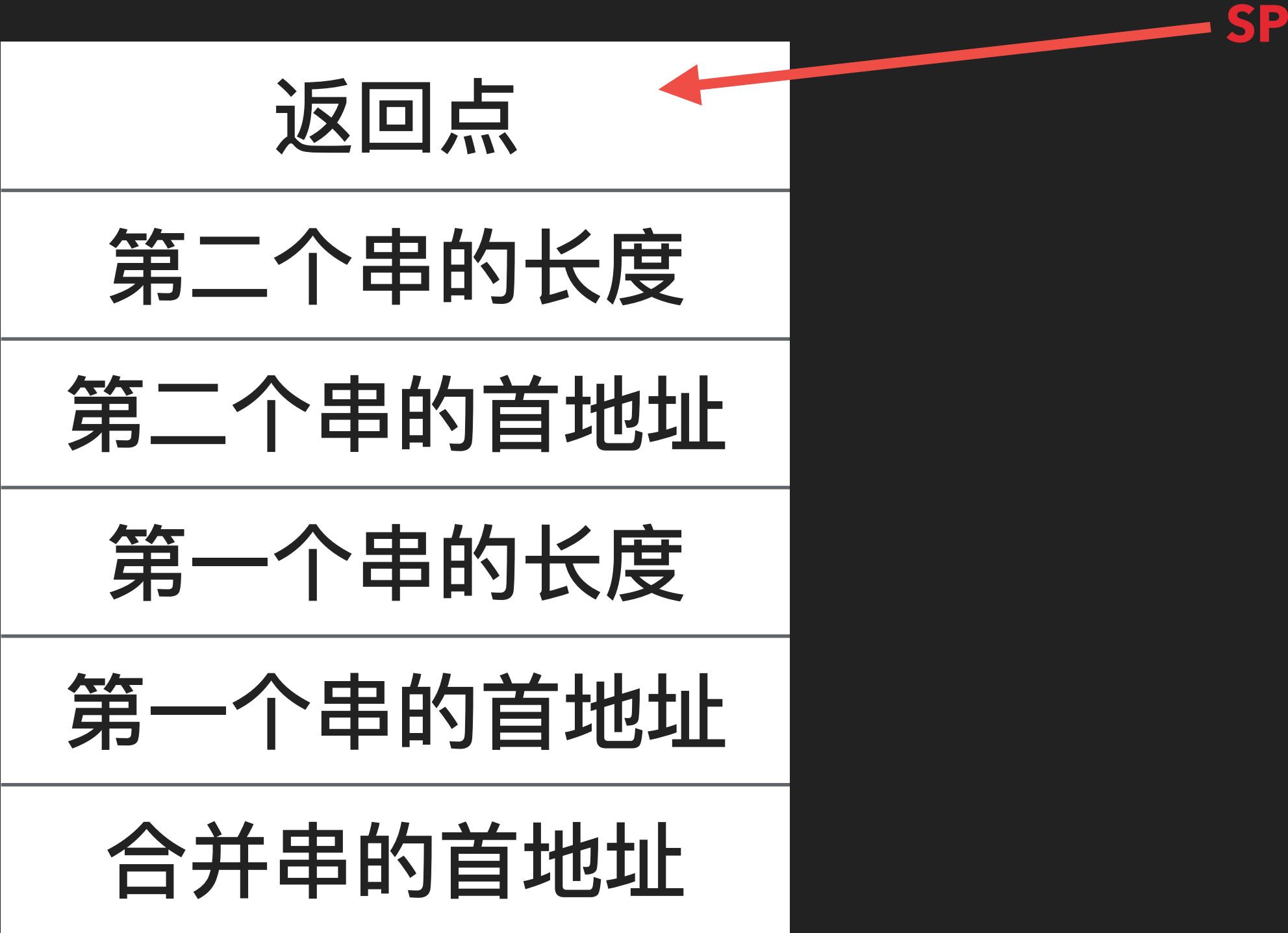
L8bit: MOV AL, BYTE PTR [BX]
        XOR AH, AH
        JMP L3

L16bit: MOV AX, WORD PTR [BX]
L3:     ROR AX, CL           ; 将(AX) 中待处理数据位循环移位至第0位
        AND AX, OFFFEH     ; 将AX的第0位清零，其余不影响
        ROL AX, CL         ; 各数据复原位
        CMP DX, 8          ; 根据数据类型保存结果
        JZ L4
        JMP L5
L4:     MOV BYTE PTR[BX], AL
        JMP Lexit
L5:     MOV WORD PTR[BX], AX
Lexit:  POP DX              ; 现场恢复
        POP CX
        POP BX
        POP AX
        POP BP
        RET 6              ; 返回时清除入口参数
CLEAN_BIT ENDP
```

第七次作业—题目7

- ▶ 设计子程序，实现：
- ▶ 合并两个字符串为一个字符串

利用堆栈传递参数，子程序堆栈



第七次作业—题目7

- 设计子程序，实现：
- 合并两个字符串为一个字符串

```
Link-string Proc
    push BP
    mov BP,SP
    push AX ;现场保护
    push CX
    push SI
    push DI
    mov DI, [BP+12] ;取合并串首地址
    mov SI, [BP+10] ;取第一个串首地址
    mov CX, [BP+8] ;取第一个串长度
```

```
LopString1: mov AL, [SI] ;第一个串首字符
            mov [DI], AL ;放入合并串首地址单元
            inc SI ;自增
            inc DI
            loop LopString1 ;CX用于循环
            mov SI, [BP+6] ;取第二个串首地址
            mov CX, [BP+4] ;取第二个串长度

LopString2: mov AL, [SI] ;第二个串首地址
            mov [DI], AL ;放入合并首地址单元
            inc SI ;自增
            inc DI
            loop LopString2 ;CX用于循环
            pop DI ;恢复现场
            pop SI
            pop CX
            pop AX
            pop BP

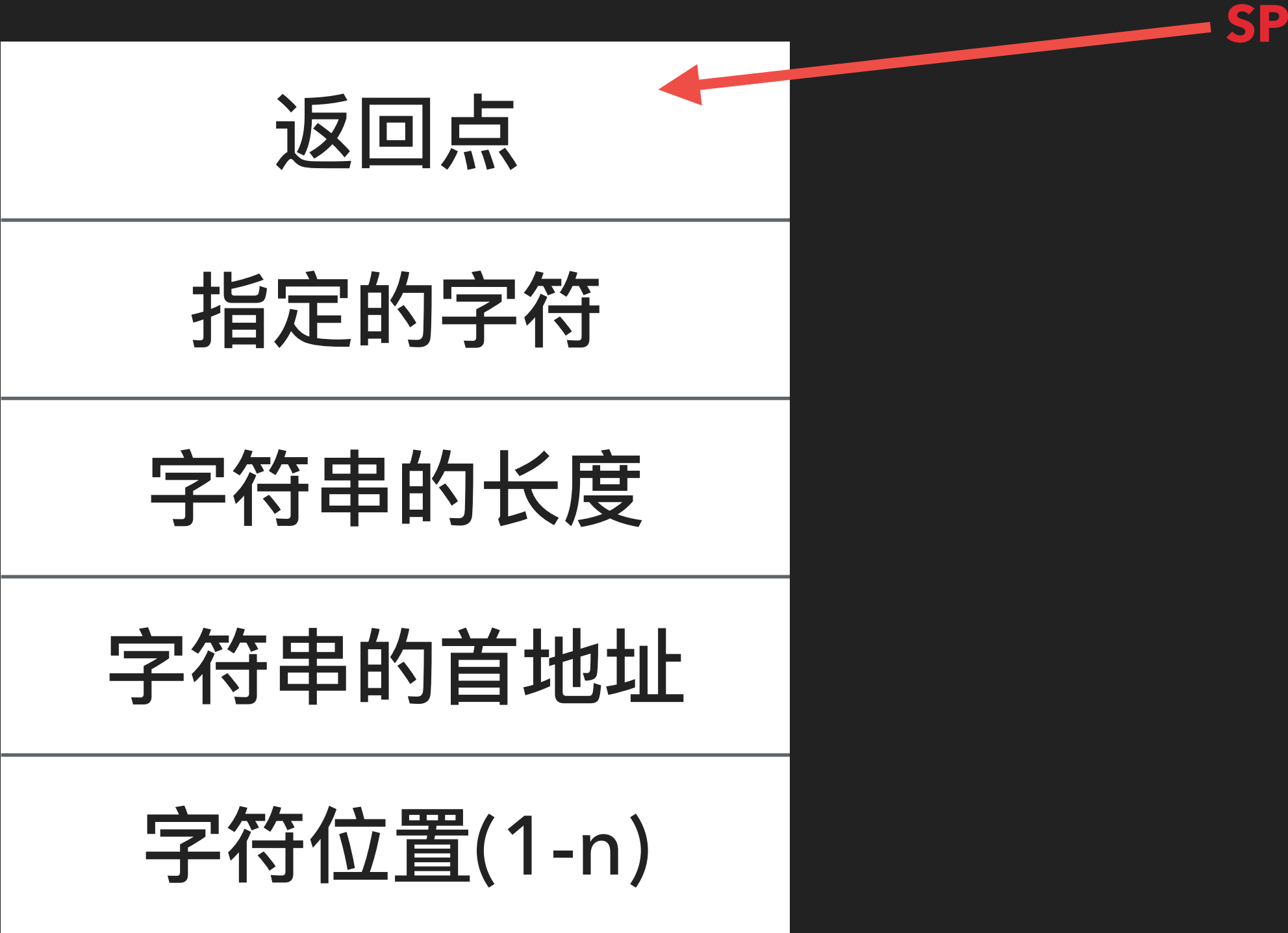
            RET 10 ;返回时清除入口参数

Link-string Endp
```

第七次作业—题目7

- ▶ 设计子程序，实现：
- ▶ 查找指定字符，返回字符位置

利用堆栈传递参数，子程序堆栈





第七次作业—题目8

- 设计子程序，实现：
- 查找指定字符，返回字符位置

```
FindASCIIcode PROC
    push    BP                ; 定义指针
    mov     BP, SP
    push    AX                ; 保护现场
    push    CX
    push    SI
    mov     SI, [BP+8]        ; 取字符串首地址
    mov     CX, [BP+6]        ; 取字符串长度
    mov     AX, [BP+4]        ; 指定的字符
    xor     AH, AH
```

```
Lop1:    cmp     AL, [SI]      ; 比较当前和目标字符
         pushf                ; 保存标志位状态
         inc     AH            ; AH记录字符位置
         inc     SI            ; 自增
         popf                ; 恢复标志位状态
         loopnz  Lop1          ; 当CX=0或ZF=1时结束
         JZ      Lfind         ; 当ZF=1则找到跳转Lfind
         mov     AH, 0FFH
Lfind:   mov     SI, [BP+10]    ; 取字符位置单元
         mov     SI, AH        ; 将AH中的内容送入SI
         pop     SI            ; 恢复现场
         pop     CX
         pop     AX
         pop     BP
         RET     8              ; 返回时清除入口参数
FindASCIIcode ENDP
```

THANKS A LOT.

Q&A

李灯举([hello@lidengju.com](mailto:hello@lidengju.com))