

第5章 循环程序设计

主讲：许万茹





本章教学内容

1

利用分支指令和LOOP指令完成
循环程序的设计

2

多重循环程序设计

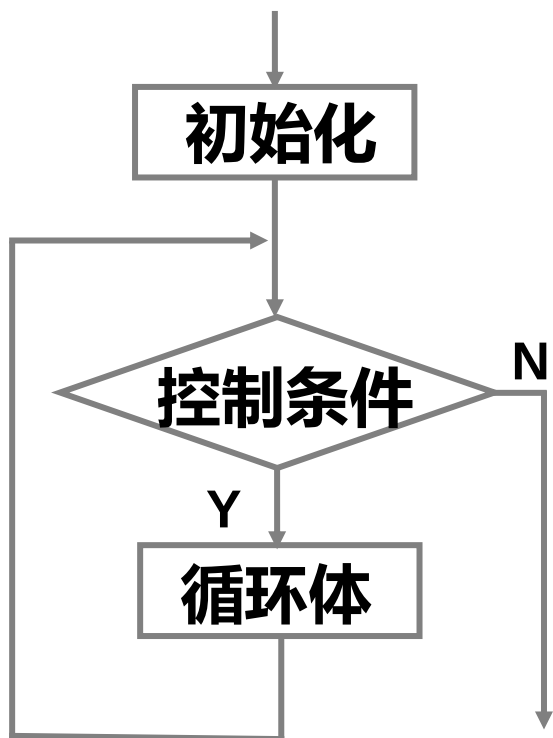
重点:

循环程序的结构（初始化、循环体和循环控制）

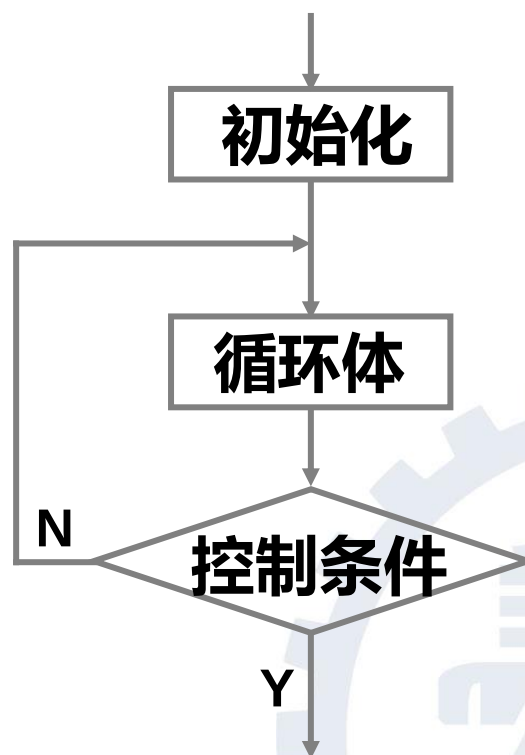
如何用分支指令和循环指令设计循环程序



循环程序的结构



DO-WHILE 结构



DO-UNTIL 结构



北京交通大学
BEIJING JIAOTONG UNIVERSITY



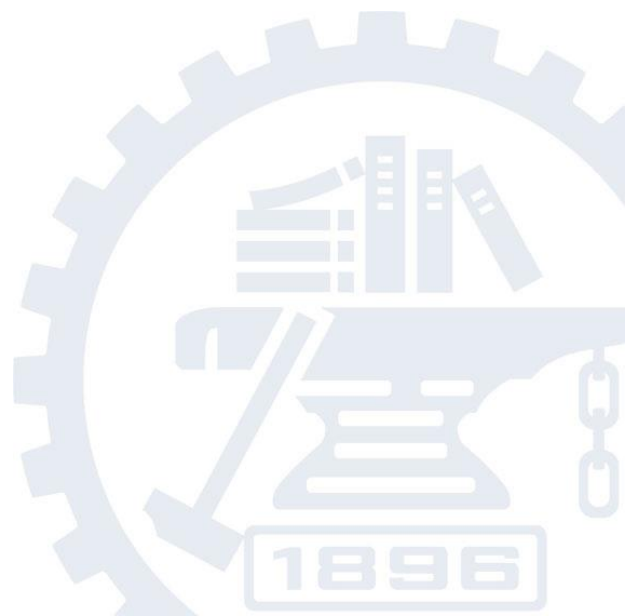
初 始 化：设置循环的初始状态

循 环 体：循环的工作部分及修改部分

控制条件：计数控制

特征值控制

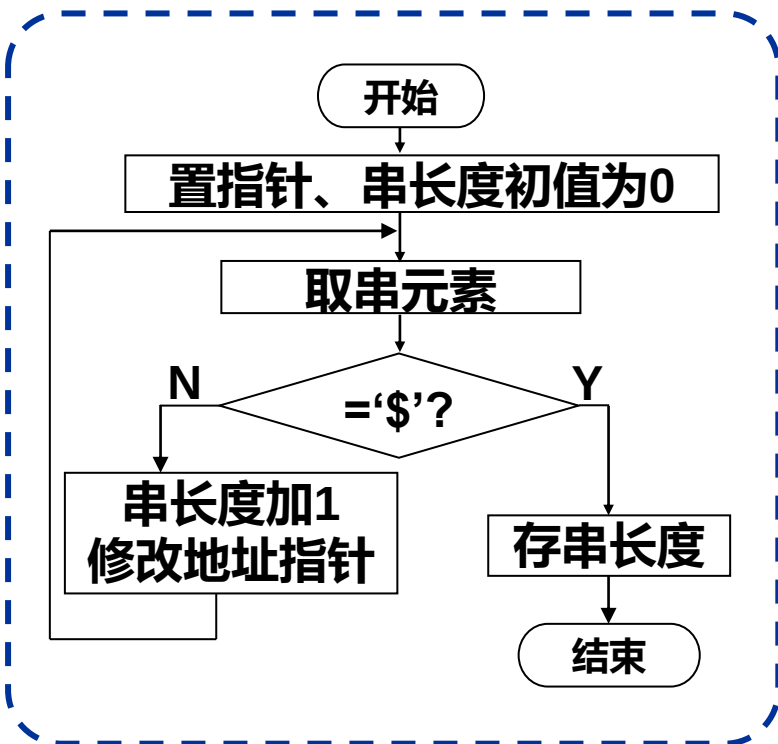
地址边界控制





用分支指令控制循环

例1：在STR开始的缓冲区中存放有一个字符串，计算该字符串的长度并存入LEN单元。



```
DATA SEGMENT
STR DB 'computer$'
LEN DB ?
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
START:MOV AX,DATA
        MOV DS,AX
        LEA SI,STR           ; 串首地址
        XOR BL,BL           ; 计数器清0
        LOP:MOV AL,[SI]     ; 取一个字节
        CMP AL,24H          ; 和'$'进行比较
        JZ STOP             ; 相等则结束
        INC BL              ; 否则计数器加1
        INC SI              ; 地址指针加1
        JMP LOP             ; 转回到LOP
STOP:MOV LEN,BL             ; 存储字符个数
        MOV AH,4CH
        INT 21H
CODE ENDS
END START
```



用专用的循环指令控制循环

涉及到的循环指令：

LOOP 无条件循环指令

LOOPZ / LOOPE

LOOPNZ / LOOPNE

条件循环指令

◆ 执行步骤：

(1) $(CX) \leftarrow (CX) - 1$

(2) 检查是否满足测试条件，如满足则
 $(IP) \leftarrow (IP) + 8$ 位位移量，实行循环；
不满足则 IP 不变，退出循环。

注意

LOOP指令执行时CX是否为0不影响ZF和CF的变化，所以LOOP指令和LOOPNZ指令才有区别！



循环指令: LOOP OPR

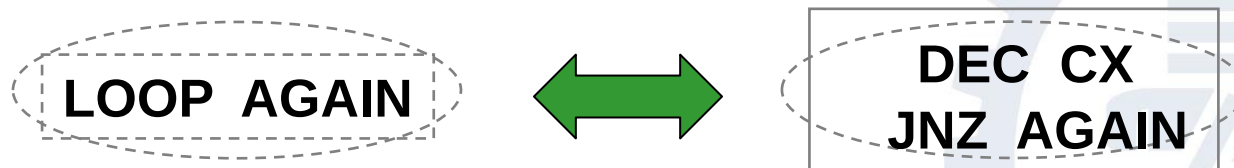
测试条件: (CX) $\neq$ 0

为零或相等时循环指令: LOOPZ(LOOPE) OPR

测试条件: ZF=1 且 (CX) $\neq$ 0

不为零或不相等时循环指令: LOOPNZ(LOOPNE) OPR

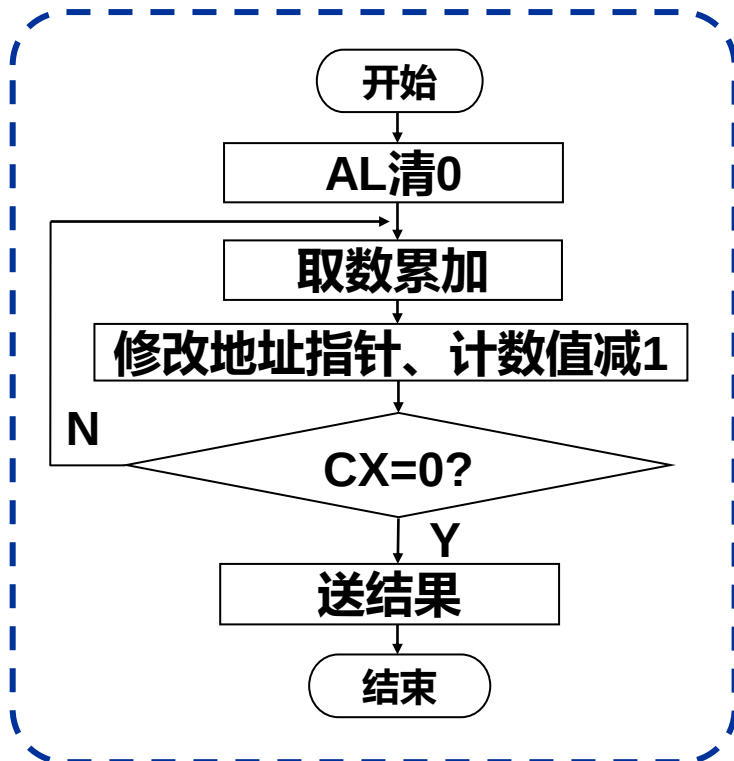
测试条件: ZF=0 且 (CX) $\neq$ 0



AGAIN是一个地址标号



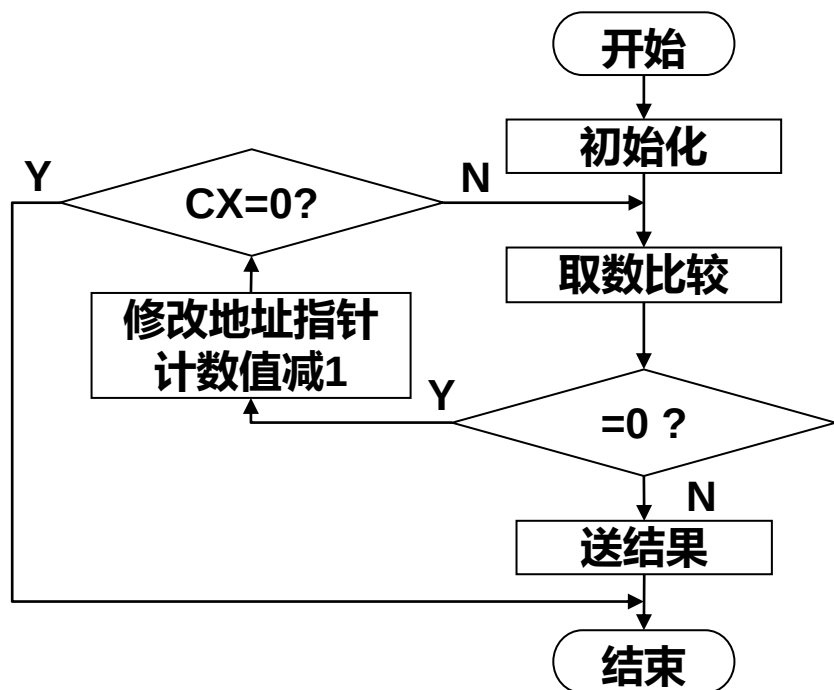
例2：求以BUF为首地址的10个内存单元的无符号数据和。已知其和小于等于255，将结果存入第11个内存单元



```
DATA SEGMENT
    BUF DB 12H,38H,46H,0BH,09H,41H,32H,56,02H,26H
    RES DB ?
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
START: MOV AX,DATA
        MOV DS,AX
        MOV AL,0    ; 存放累加之和
        MOV CX,0AH  ; 累加次数
        LEA BX,BUF  ; 数据表的首地址
        LP: ADD AL,[BX] ; 累加
            INC BX    ; 地址增1
            LOOP LP   ; 若CX-1不为0, 则继续循环
        MOV RES,AL  ; CX-1=0, 则存累加和
        MOV AH,4CH
        INT 21H
CODE ENDS
END START
```



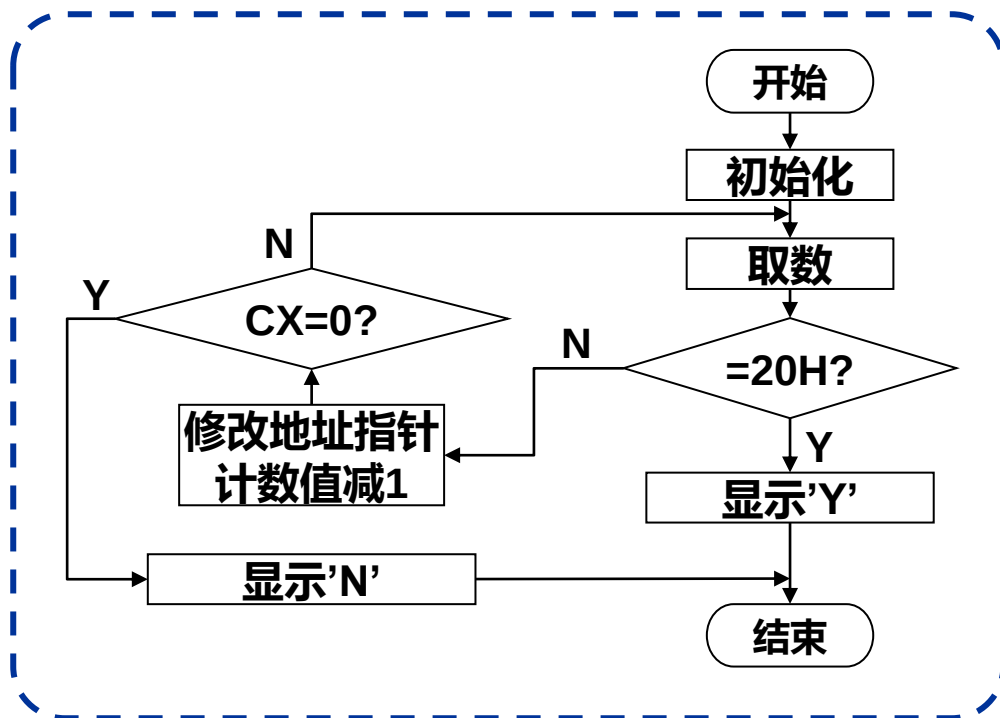
例3：在字节数组中找出第一个非0的数据，并将其下标存入RES单元，假设其下标值小于10



```
DATA SEGMENT
ARR DB 0,0,38H,46H
    DB 89H,67H,0H,92H
CNT EQU $-ARR
RES DW ?
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
START: MOV AX,DATA
      MOV DS,AX
      MOV CX,CNT ; 循环次数,CNT数组长度
      MOV DI,-1 ; 数组下标从0开始
AGAIN: INC DI
      CMP ARR[DI],0 ; 和0比较
      LOOPZ AGAIN ; 为0且没比较完,则循环
      JZ EXIT ; 比较完仍为0,转EXIT
      MOV RES,DI ; 找到了,送下标号
EXIT: MOV AH,4CH
      INT 21H
CODE ENDS
      END START
```



例4：在字符串中从前向后查找空格字符（ASCII码为20H），找到显示Y，否则就显示N



```
DATA SEGMENT
STR DB 'ASDFK LIO OP', '$'
LEN EQU $-STR
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
START:MOV AX,DATA
    MOV DS,AX
    MOV CX,LEN ; 循环次数
    MOV SI,-1 ; 字符串下标从0开始
    MOV AL,20H
NEXT:INC SI
    CMP AL,STR[SI] ; 和20H比较
    LOOPNE NEXT ; 不相等，继续循环
    JNZ NFIND ; CX为0跳转
    MOV DL,'Y'
    MOV AH,2
    INT 21H
    JMP EXIT
NFIND:MOV DL,'N'
    MOV AH,2
    INT 21H
EXIT:MOV AH,4CH
    INT 21H
CODE ENDS
    END START
```



用计数器控制循环

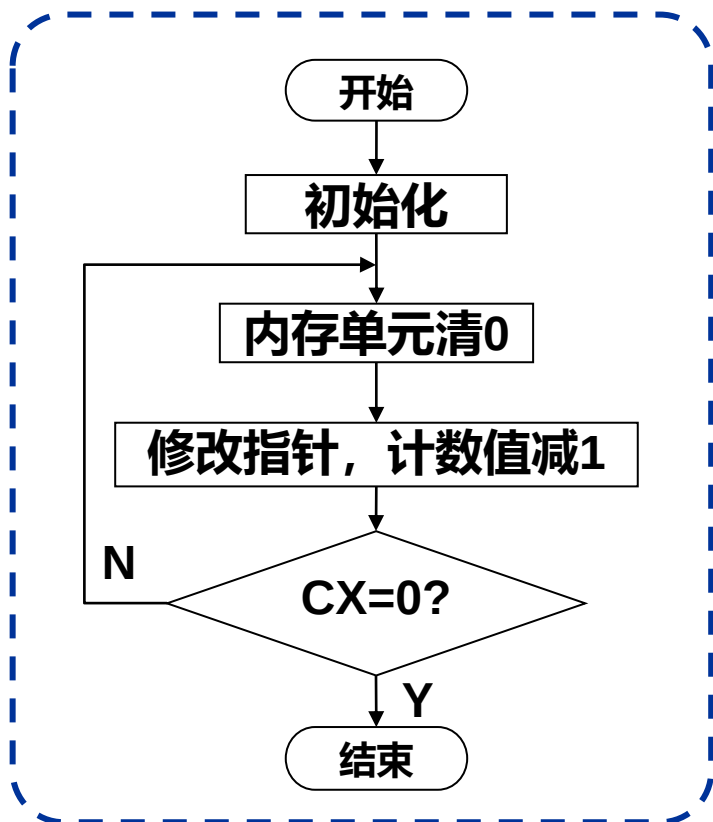
例5：已知数据块的长度，统计数据块中正数和负数的个数

```
DATA SEGMENT
BUF DB -32,25,36,-18,-64,0,-3
COUNT EQU $-BUF
PLUS DB ? ; 存放正数个数
MINUS DB ? ; 存放负数个数
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE, DS:DATA
START:MOV AX,DATA
        MOV DS,AX
        MOV BL,0 ; 负数个数
        MOV DL,0 ; 正数个数
        MOV SI,OFFSET BUF
        MOV CX,0 ; 循环初值
LOP1:MOV AL,[SI] ; 取值
```

```
        CMP AL,0 ; 和0比较
        JGE NEXT0 ; 大于等于0转移
        INC BL ; 小于0, BL加1
        JMP NEXT1
NEXT0:INC DL ; 大于等于0, DL加1
NEXT1:INC SI ; 指针加1
        INC CX ; 计数加1
        CMP CX,COUNT ; 比较次数
        JL LOP1 ; 没结束, 则继续
        MOV MINUS,BL ; 存负数个数
        MOV PLUS,DL ; 存正数个数
        MOV AH,4CH
        INT 21H
CODE ENDS
END START
```



例6：将BUF单元开始的100个字节存储单元全部清0

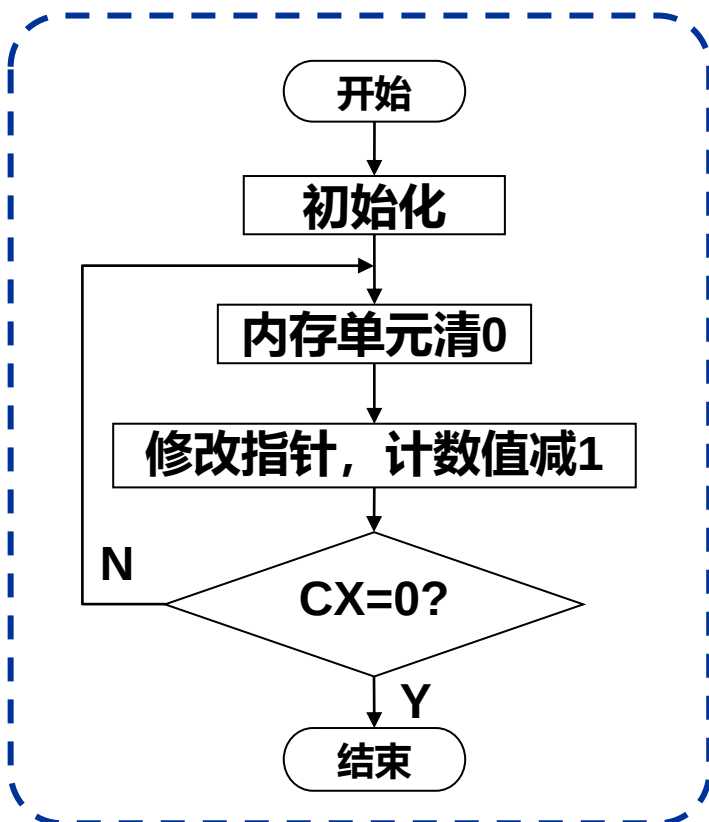


```
DATA SEGMENT
    BUF DB 100 DUP (?)
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
START:MOV AX,DATA
        MOV DS,AX
        MOV BX,OFFSET BUF    ; 地址指针
        MOV CX,64H           ; 计数初值
LP:MOV BYTE PTR [BX],0      ; 清0
        INC BX                ; 地址加1
        LOOP LP              ; CX减1不为0, 则继续
        MOV AH,4CH
        INT 21H
CODE ENDS
END START
```



例6：将BUF单元开始的

含义：利用循环次数作为控制条件
应用场合：循环次数已知
分类：正计数法和倒计数法
设计方法：初值放入CX



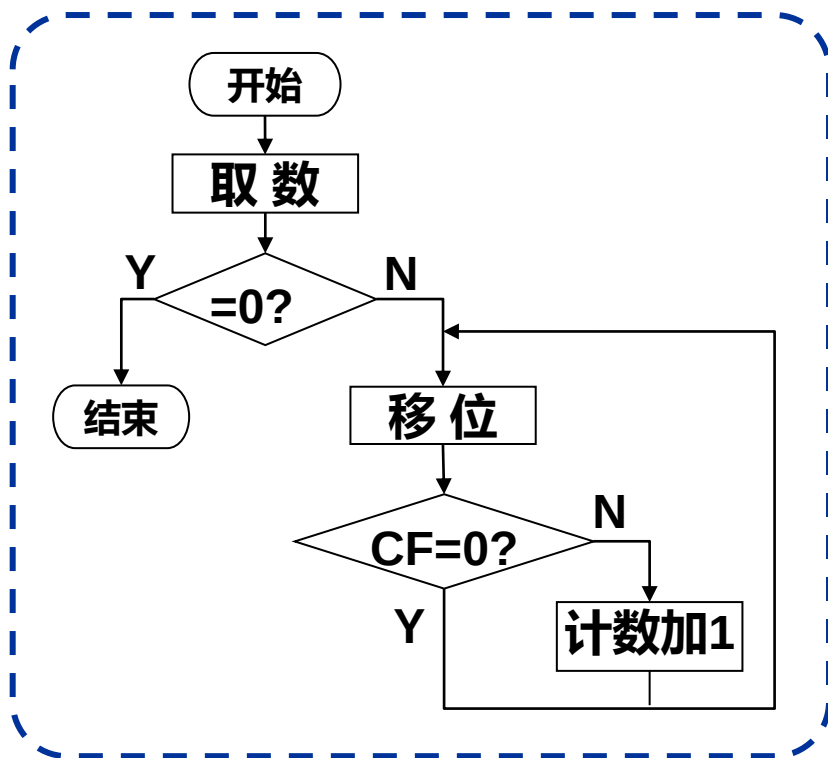
```
DATA SEGMENT
    BUF DB 100 DUP (?)
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
START:MOV AX,DATA
        MOV DS,AX
        MOV BX,OFFSET BUF ; 地址指针
        MOV CX,64H ; 计数初值
LP:MOV BYTE PTR [BX],0 ; 清0
        INC BX ; 地址加1
        LOOP LP ; CX减1不为0, 则继续
        MOV AH,4CH
        INT 21H
CODE ENDS
END START
```



含义：用转移指令来判断循环条件
应用场合：循环次数是不知道或不确定

按问题的条件控制循环

例7：记录某个字节存储数据单元中1的个数，并把结果存入RES中

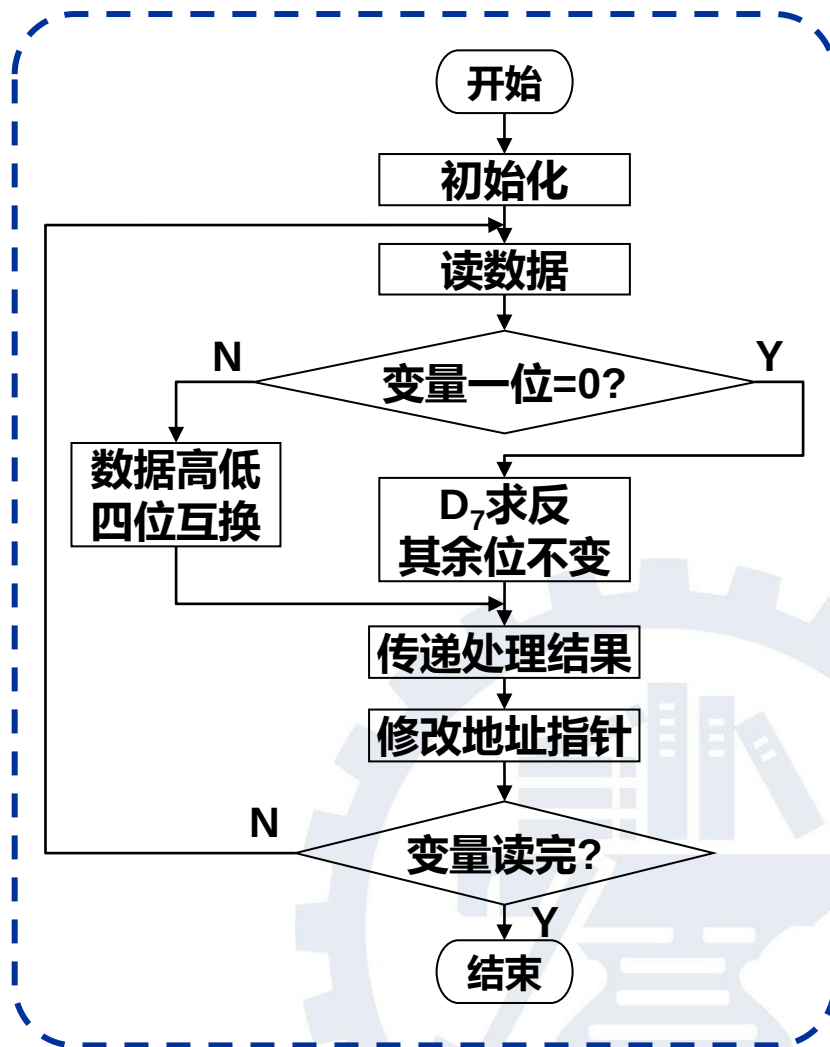


```
DATA SEGMENT
    NUM DB 75H
    RES DB ?
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
START:MOV AX,DATA
    MOV DS,AX
    MOV BL,NUM
    XOR DL,DL ; DL清零
AGAIN:TEST BL,0FFH
    JZ NEXT ; 判断数据是否0
    SHR BL,1 ; 逻辑右移1位
    ADC DL,0 ; 带进位加法
    JMP AGAIN
NEXT:MOV RES,DL
    MOV AH,4CH
    INT 21H
CODE ENDS
    END START
```



按逻辑变量控制循环

例8：在以BUF为起始地址的内存中放有若干个字节型无符号数，假定某逻辑变量的长度为一个字节（其值为10010101），它的D0~D7位对应着BUF~BUF+7单元内容的运算。即某位为0，则将相应单元内容的最高位求反，其它位不变；而某位为1，则将相应单元内容之高低四位互换。





含义：用转移指令来判断循环条件

应用场合：控制转入不同的循环支路

方法：把逻辑变量送入寄存器中，以逻辑变量各位的状态作为执行某段程序的标志

DATA SEGMENT

BUF DB 75H,12H,87H,98H

DB 81H,56H,73H,51H

B EQU 8

C EQU 10010101B

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START:MOV AX, DATA

MOV DS,AX

MOV AH,B; 指示数据个数

MOV CH,C

LEA BX,BUF

LP:MOV AL,[BX]

SHR CH,1

JNC NEXT; CF=0跳转

MOV CL,4

ROL AL,CL; 高低位交换

JMP RES

NEXT:XOR AL,80H; 高位取反

RES:MOV [BX],AL

INC BX

DEC AH; 数据个数-1

JNZ LP; 未读完则继续

MOV AH,4CH

INT 21H

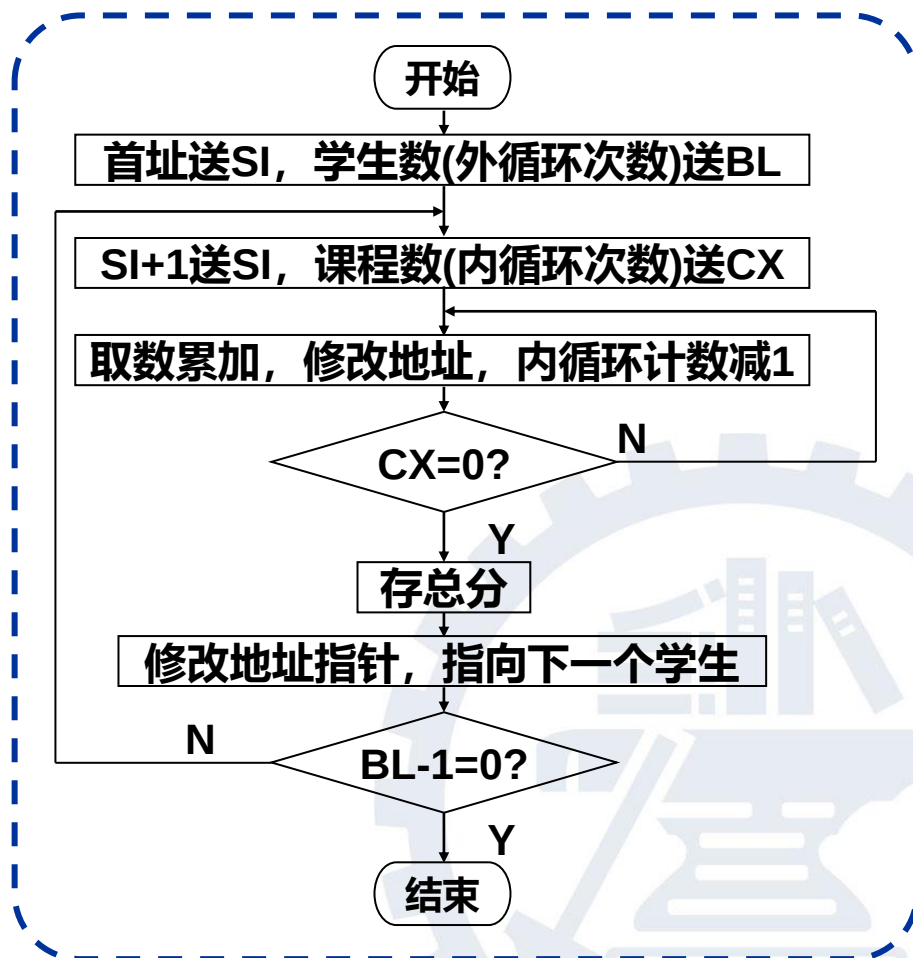
CODE ENDS

END START



多重循环程序设计

例9：设在以EXST为首址的存储区中依次存放着某考区245个理科生的七门成绩，现要统计每个考生的总成绩，并将其存放在该考生单科成绩之后的两个单元





```
DATA SEGMENT
```

```
EXST DB 01,75,82,84,92,78,49,85,00,00
```

```
DB 02,83,92,63,76,82,58,69,00,00
```

```
...
```

```
DATA ENDS
```

```
CODE SEGMENT
```

```
ASSUME CS:CODE,DS:DATA
```

```
MOV AX,DATA
```

```
MOV DS,AX
```

```
START:LEA SI,EXST
```

```
MOV BL,245
```

```
LOP2: MOV CX,7
```

```
XOR AX,AX
```

```
INC SI
```

```
LOP1: ADD AL,[SI]
```

; 数据表首地址

; 245个学生，外循环次数

; 七门课成绩，内循环次数

; 清0，存总成绩

; 跳过准考证号

; 单科成绩累加



含义：指循环体内还有循环，也就是循环嵌套
注意：(1) 不允许循环结构交叉；(2) 转移指令只能从循环结构内转出或可在同层循环内转移

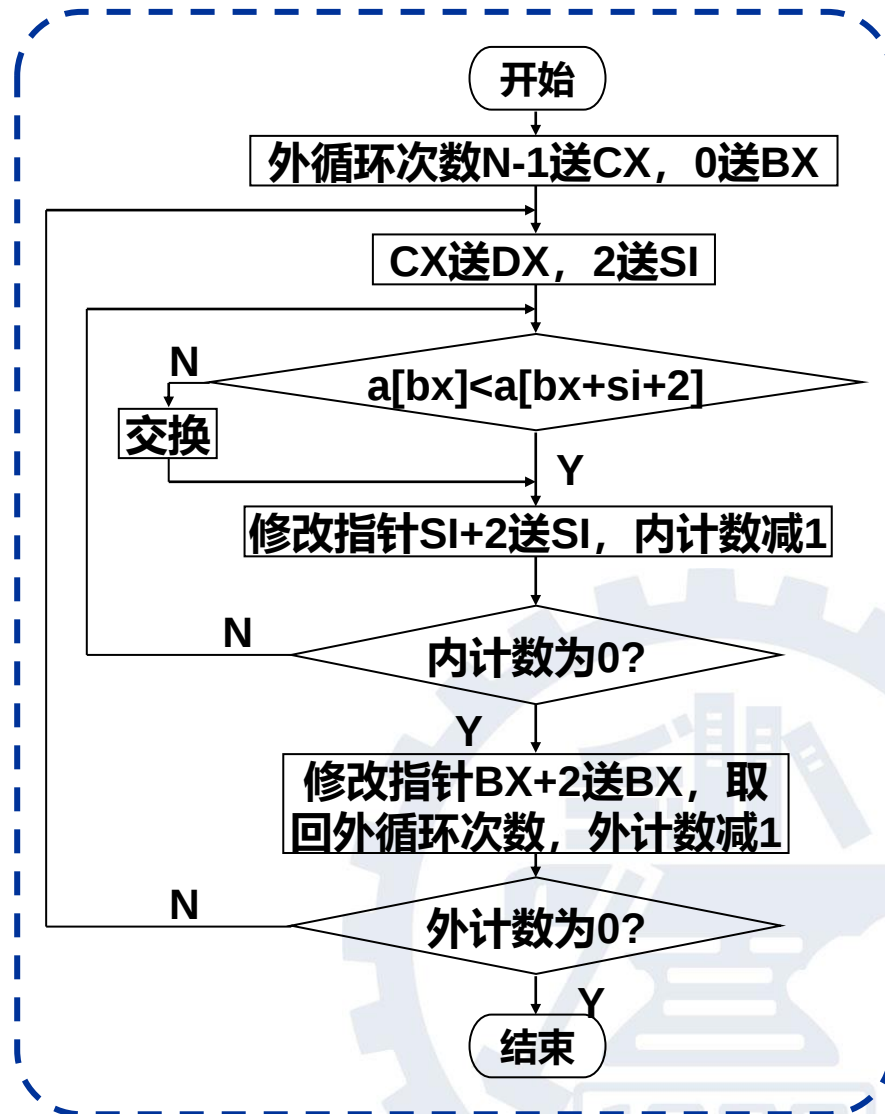
```
ADC AH,0           ; 加进位位
INC SI             ; 修改地址指针
LOOP LOP1          ; 没累加完单科成绩，则继续
MOV WORD PTR [SI],AX ; 累加完，存总成绩
INC SI             ; 跳过存总成绩的2个单元
INC SI
DEC BL             ; 外循环次数减1
JNZ LOP2           ; 不为0，则求下个学生总成绩
MOV AH,4CH
INT 21H
CODE ENDS
END START
```



例10（自学）：将N个不同的无符号数 $a_1, a_2, \dots, a_n$ 由小到大进行排序。若每个数占一个字，则N个数可定义如下：

A DW $a_1, a_2, a_3, \dots, a_n$

它们的内存分配分别为：
 $A[0], A[2], A[4], \dots, A[2n]$





北京交通大学

BEIJING JIAOTONG UNIVERSITY



取N=10

```
DATA SEGMENT
    A DW 1223,83,456,355,89
      DW 948,5,123,789,567
    CNT EQU ($-A)/2
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
START: MOV AX,DATA
      MOV DS,AX
      MOV CX,CNT-1
      MOV BX,0
LOOP1: MOV DX,CX; 暂存
      MOV SI,2
LOOP2: MOV AX,A[BX]
```

```
      CMP AX,A[BX+SI]
      JBE L1
      XCHG AX,A[BX+SI]
      MOV A[BX],AX
L1: ADD SI,2
      LOOP LOOP2; 控制内循环
      ADD BX,2
      MOV CX,DX
      LOOP LOOP1; 外循环CX-1
      MOV AH,4CH
      INT 21H
CODE ENDS
END START
```



1、在STR到STR+99单元中存放着一个字符串，试编写程序测试该字符串中是否有数字，若有将CL置1，否则CL置0（要求在debug中调试出该程序）。

2、在字节数组中找出第一个负数，并将该负数存入RES单元中；假设该数组中包含20个带符号数，且至少有1个负数（要求在debug中调试出该程序）。



作业

