【教材 4.5】根据下列程序段

```
        ……
        MOV    SI,      OFFSET  BUF1
        MOV    DI,      OFFSET  BUF2
        MOV    CX,      n
LOOPA:  MOV    AL,      [SI]
        MOV    [DI],    AL
        INC    SI
        INC    DI
        LOOP   LOOPA            ；使用 CX 计数
        ……
```

绘制流程图，并回答如下(1)至(5)问题，其中 BUF1、BUF2 均为串长度为 n 的字节存储区首址。

(1)该程序段完成了什么工作？

(2)若将指令” MOV CX, n”误写为“MOV CX, 0”，则循环体被执行多少次？

(3)若漏掉了指令“MOV CX, n”，则循环体执行的次数能确定吗？为什么？

(4)若漏掉了指令“INC SI”，则程序运行结果如何？

(5)若不小心将标号 LOOPA 上移了一行，即将标号写在指令” MOV CX, n” 之前，则程序运行情况如何？

【解答】

这是一个，其流程图如图 1 所示。

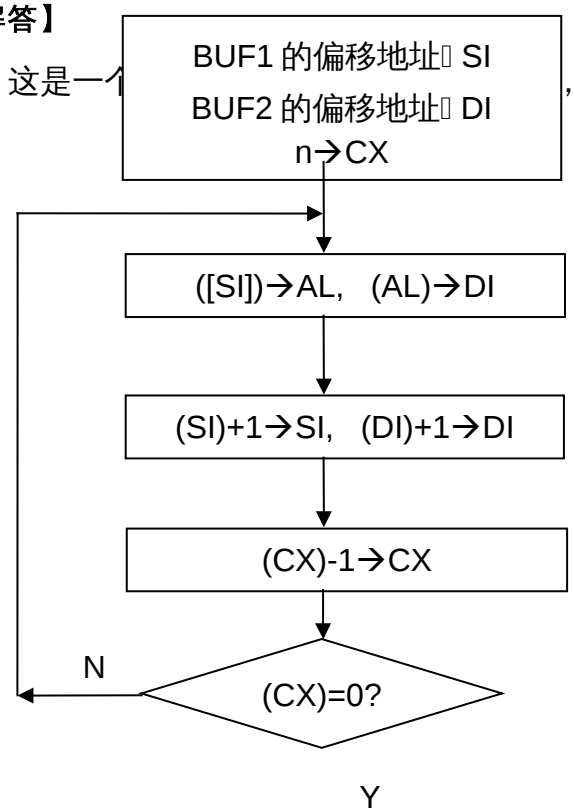

图 1 教材 4.5 的流程图

⑴ 该程序段完成如下工作：将 BUF1 存储区中的 n 个字节数据传送到 BUF2 为首址的存储区中。

(2) 若将指令" MOV CX, n"误写为"MOV CX, 0" ，则当 (CX)=0, 0FFFFH, 0FFFEH，……，1 时执行循环体，即循环体被执行 0FFFFH+1 次。

(3) 若漏掉了指令"MOV CX, n"，则循环体执行的次数无法确定，因为 CX 的初值是随机的，循环体执行次数随着 CX 不同的初值而变化。

(4) 若漏掉了指令"INC SI"，则程序将 BUF1 的第一个字节传送到 BUF2 的 n 个字节存储区。

(5)若不小心将标号 LOOPA 上移了一行，即将标号写在指令" MOV CX, n" 之前，则程序运行情况有以下两种：

a.若 n=1，则循环体执行一次；

b.若 n≠1，则循环体被无限次执行，即造成死循环。

【教材 4.6】有若干行字符串存放在以 BUF 为首址的字节存储区中，最后以空白(0H)作结束标志。现需删除第 5 行的内容，并将删除前后 BUF 区的内容显示出来。试编写其程序。注：每行字符串的结束标志 0DH、0AH。

【解答】

解题思路：假定总行数大于 5，则先寻找第四行的末尾，即第四个行字符串结束标志 0AH，即可得到第 5 行的起始地址；再寻找第五行的末尾字符 0AH，即第六行的第一个字符，将第六行及之后的各行向前移动，覆盖第五行的字符数据。最后从第一行开始显示 BUF 存储区的字符。

寄存器和变量定义如下：

AL:存放当前字符；

SI：取字符指针，初值指向 BUF-1

DI: 送字符指针(第 6 行及其之后的字符要依次送到第 4 个换行符之后)

CX：行数计数器，初值为 4，每循环一次之后其值减 1，当(CX)=0 时，SI 指第 4 个换行符，即第 4 行的行尾。

程序如下：

```
.386
STACK SEGMENT USE16 STACK
    DB  200 DUP(0)
STACK ENDS
DATA SEGMENT USE16
BUF     DB 'LINE1:AAAAAAAAAAAAAAAA',0DH, 0AH
    DB 'LINE2:BBBBBBBBBBBBBB  ', 0DH, 0AH
    DB 'LINE3:CCCCCCCCCCCCCCCC', 0DH, 0AH
    DB 'LINE4:DDDDDDDDDDDD     ',0DH, 0AH
    DB 'LINE5:EEEEEEEEEEEEEEEE', 0DH, 0AH
    DB 'lINE6:FFFFFFFFFFFFFFFF    ',0DH, 0AH
    DB 'LINE7:GGGGGGGGGGGGGGGGGGGGGG', 0DH, 0AH, 0
AFTER DB 0DH,     0AH,     'AFTER******************************AFTER',0DH,
0AH,'$'
N=5
DATA ENDS
CODE SEGMENT USE16
ASSUME CS:CODE, DS:DATA, SS:STACK
BEGIN: MOV    AX, DATA
        MOV    DS, AX
        LEA         SI,  BUF
LOP2:  MOV    DL, [SI]
```

```
        CMP     DL, 0
        JE      LOP1
        MOV     AH, 2
        INT     21H
        INC     SI

        JMP         LOP2；    显示删除前的字符串

LOP1:   LEA         DX,AFTER
        MOV     AH,9

        INT     21H；     在删除前后的字符中间显示 AFTER*******AFTER

        LEA         SI,  BUF-1
        MOV     CX,N-1
        CMP     CX,0

        JE      NEXT22；判断是不是删除第一行，若是直接寻找第二行的开
始

NEXT1:      INC     SI
        CMP     [SI],           BYTE PTR 0AH
        JNE         NEXT1

        LOOP    NEXT1   ；  寻找第 N-1 行的换行符

NEXT22:     MOV     DI, SI    ；        使 DI 指向第 N-1 个换行符

NEXT2:      INC     SI
        CMP     [SI],       BYTE PTR 0AH

        JNE         NEXT2   ；  寻找第 N 个换行符

NEXT3:      INC     SI
        INC     DI
        MOV     AL,         [SI]
        MOV     [DI],           AL
        CMP     AL, 0

        JNE         NEXT3；将第 N 个换行符之后的字符依次移到第 N-1 个换
行符后

        LEA         SI,  BUF
PLAY:   MOV     DL,         [SI]
        OR      DL, DL
        JZ      EXIT
        MOV     AH,         2
        INT     21H
```

```
        INC     SI
        JMP         PLAY；  显示删除后的内容
EXIT:   MOV     AH, 4CH
        INT     21H
CODE ENDS
END BEGIN
```