

第八周作业

3.34

(1) AX,BX相加结果为9557H, 没有溢出, 转向L1

(2) AX,BX相加结果为0F830H, 没有溢出, 转向L1

(3) AX,BX相加结果为0A355H, 两个正数相加得负数, 溢出, 没有进位, 故转到L2

(4) AX,BX相加结果为6FF3H, 两个负数相加得正数, 溢出, 有进位, 相加后,AX,BX相减结果为0B023H,有借位, 正数减负数得负数, 有溢出, 故转向L5

(5) AX,BX相加结果为05A1FH, 两个负数相加得正数, 溢出, 有进位, 相加后,AX,BX相减结果为94B7H,有借位, 正数减负数得负数, 有溢出, 故转向L5

3.36

首先了解ADC为带进位加法, 首先用不带进位的加法给AX赋值 $2 \cdot p$, 然后用带进位的加法给DX赋值 $2 \cdot p$, 比较带进位加法的结果 $2 \cdot p$ 和 q 两个数的大小关系, 如果 $2 \cdot p$ 小于 q , 则AX赋值为2, 如果 $2 \cdot p$ 小于等于 q , 则AX赋值为1,

5.12

```
data segment
    MEM      dw 1,2,0,3,4,0,5,6,0,7
             dw 90 dup(1)
    changerow db 0dh,0ah,'$'
    cnt      dw 0
data ends

stack segment
    dw 128 dup(0)
stack ends

code segment
    assume cs:code,ds:data,ss:stack
start:
    mov     ax,data
    mov     ds,ax

    mov     si,100*2-2
    mov     bx,-2
    mov     cx,100
loop1:
    add     bx,2
    cmp     MEM[bx],0
    je      deal
    loop    loop1
    jmp     exit
deal:
    mov     di,bx
; 开始向前移动
loop2:
;看是否到了最后一个元素
    cmp     di,si
    je      add_0
    mov     ax,MEM[di+2]
    mov     MEM[di],ax
    add     di,2
    jmp     loop2
add_0:
    mov     word ptr MEM[si],0
    loop    loop1

exit:
```

```

mov     ah,4ch
int     21h

code ends
end start

```

未操作前的数组

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

D:\>link D:\TEST.OBJ; >>C:\62851.LOG
D:\>debug D:\TEST.exe
-t
AX=076C BX=0000 CX=020B DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075C ES=075C SS=076B CS=0789 IP=0003 NV UP EI PL NZ NA PO NC
0789:0003 8ED8 MOV DS,AX
-t
AX=076C BX=0000 CX=020B DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076C ES=075C SS=076B CS=0789 IP=0005 NV UP EI PL NZ NA PO NC
0789:0005 BEC600 MOV SI,00C6
-d 076c:0000
076C:0000 01 00 02 00 00 00 03 00-04 00 00 00 05 00 06 00
076C:0010 00 00 07 00 01 00 01 00-01 00 01 00 01 00 01 00
076C:0020 01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00
076C:0030 01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00
076C:0040 01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00
076C:0050 01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00
076C:0060 01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00
076C:0070 01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00
-▲

操作后的数组

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
D:\>link D:\TEST.OBJ; >>C:\05546.LOG

D:\>debug D:\TEST.exe
-g

Program terminated normally
-d 076c:0000
076C:0000  01 00 02 00 03 00 04 00-05 00 06 00 07 00 01 00  .....
076C:0010  01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00  .....
076C:0020  01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00  .....
076C:0030  01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00  .....
076C:0040  01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00  .....
076C:0050  01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00  .....
076C:0060  01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00  .....
076C:0070  01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00  .....
-d 076c:0070
076C:0070  01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00  .....
076C:0080  01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00  .....
076C:0090  01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00  .....
076C:00A0  01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00  .....
076C:00B0  01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00  .....
076C:00C0  01 00 00 00 00 00 00 00-0D 0A 24 00 00 00 00 00  .....$.
076C:00D0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
076C:00E0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
-a
```