# 实验一 多位十六进制加法运算实验

CODE SEGMENT ;定义代码段

ASSUME CS:CODE,DS:DATA ;段属性说明，代码段放在CS中，数据段放在DS

BEG: MOV AX,DATA ;把数据（16位的段基址0712H，AH=07，AL=12）放入AX中

MOV DS,AX ; AX中的数据送到DS，对DS进行初始化

MOV SI,OFFSET NUM1 ;把NUM1的偏移地址0000H送到SI（原变址寄存器）

MOV AX,0 ;AX清零，为0000H

ADD AX,[SI+0] ;16位的段基址0712H左移4位后为07120h与偏移地址 0000h相加得到物理地址07120h，把物理地址为07120h的内容1111与AX相加再放入AX

物理地址为DS:SI=0712：0000

ADD AX,[SI+2] ;[SI]+02h获得新的偏移地址，物理地址发生变化为0712:0002，把其内容2222与AX相加为3333放入AX中。

ADD AX,[SI+4] ; 同理，3333与AX相加为6666

MOV [SI+6],AX ; [SI]+06H获得新的物理地址，把所得结果放入该物理地址

JMP $ ;跳回当前指令，JMP无条件转移指令

CODE ENDS ;代码段结束

DATA SEGMENT ;定义数据段

NUM1 DW 1111H ;N1 ;定义变量，为字型数据，NUM1=1111H

NUM2 DW 2222H ;N2

NUM3 DW 3333H ;N3

NUM4 DW 0000H ;N4

DATA ENDS ;数据段结束

END BEG ;源程序结束

功能：三个十六位二制数相加运算，N4 为存放结果，其中 N1 为 1111H、N2 为 2222H、 N3 为 3333H 求N1+N2+N3的运算结果，所以结果应该为 6666H

# 实验二 循环程序实验

CODE SEGMENT ; 代码段

ASSUME CS:CODE ;代码段放在CX里

CON\_A EQU 25 ;定义常数， EQU宏替换，表示遇到CON-A用25替代

CON\_B EQU 12 ; EQU表示遇到CON-B用12替代

START:

MOV AX,0 ; 初始设置AX为00000h

MOV CX,**5** ; CX为00005h，共循环了5次

**INC\_AX:NOP**  ;NOP空指令 ，空运行一次

**INC AX**  ;每执行一次AX加1

**LOOP INC\_AX** ; LOOP表示循环，每循环一次CX减1，CX减到0，结束程序

JMP $

CODE ENDS

END START

功能：通过给 CX 一个数值，再通过 LOOP 作一个判断 CX-1 是否为 0 的转移， 实现程序的循环，循环的内容是执行 AX+1， 所以结果应该为 AX 最后大小为开始时给定 CX 的大小

# 实验三 分支程序实验

CODE SEGMENT

ASSUME CS:CODE

CON\_A EQU 25 ; EQU宏替换

CON\_B EQU 12

START:

MOV AX,CON\_A ; 把CON\_A放到AX中，即为25，AX=19H

MOV BX,CON\_B ; BX=0CH

**CMP AX,BX**  CMP REG,REG 比较指令，不产生结果，但影响标志位 OF、SF、ZF、PF、CF=0，AF=1，根据标志位确定大于、小于或等于，操作数1-操作数2，

AX-BX=0001 1001 - 0000 1100 =0000 1101

**JNC MO\_T**  ;AX > BX 跳转 JNC 判别进位标志位，CF=0跳转，即AX>BX 就跳到MO\_T: JMP $

**JE EQUA** ;AX = BX 跳转 JE 相等跳转 if ZF = 1 then jump

**JC LESS** ;AX < BX 跳转 JC进位位为1跳转，即AX < BX 跳转

MO\_T: JMP $ ; MO\_T:、EQUA: 、LESS: 冒号前面的这些为标号

EQUA: JMP $

LESS: JMP $

CODE ENDS

END START

功能：本实验是通过改变两个变量 CON\_A 和 CON\_B 的大小，实现用 CMP 指令对不同标示 位的影响的一个转移，分别设有大于、等于和小于

# 实验四 内存块移动实验

①无重叠部分

CODE SEGMENT ;定义代码段

ASSUME CS:CODE ;

START:

MOV SI,1000H ;

MOV CX,100 ;CX=64H，为LOOP的计数值（循环次数）

MOV AL,1 ;AL=01H

/把AL的数据送到SI，存入 从1000H 开始的 100 个字节数据为 1 到 100，数值分别存储到物理地址为DS:SI=0700：1000H至7000:1063H的存储单元中/

**PU\_IN: MOV [SI],AL** ;

INC AL AL＋1

INC SI SI+1

**LOOP PU\_IN**

**MOV CX,100** ;CX赋值100，作后续循环跳转指令计数值，

**MOV SI,1000H** ;

**MOV DI,1100H** ; 设置目的块地址，把从1000H地址开始的100个数移动 从1100H开始的内存处。

/把从偏移地址为1000H开始的存储单元内容传送到AL，AL再传送到从偏移地址为1100H开始的存储单元中，一个个存储单元进行转移，实现内存块的移动。/

FADR: MOV AL,[SI] ;重新设置SI（源指针）、DI地址

MOV [DI],AL

INC SI ;SI+1，SI源变址寄存器

INC DI ;DI+1，DI目标变址寄存器

DEC CX ;CX自减1

JNE FADR ; JNE条件转移指令，当ZF=0（运算结果不为0），即CX≠0，转至标 号FADR处执行，若ZF=1即CX=0则向下执行

JMP $ ;IMP无条件转移指令

CODE ENDS

END START

②有重叠部分

把1000H的100个数移到1030H

30H=48（把1移到48,2移到49……）

CODE SEGMENT ;定义代码段

ASSUME CS:CODE ;

START:

MOV SI,1000H ;

MOV CX,100 ;CX=64H，为LOOP的计数值（循环次数）

MOV AL,1 ;AL=01H

/把AL的数据送到SI，存入 从1000H 开始的 100 个字节数据为 1 到 100，数值分别存储到偏移地址为1000H至1063H的存储单元中/

**PU\_IN: MOV [SI],AL** ;

INC AL AL＋1

INC SI SI+1

**LOOP PU\_IN**

**MOV CX,100** ;CX赋值100，作后续循环跳转指令计数值，

**MOV SI,1000H+100** ;

**MOV DI,1030H+100** ; 设置目的块地址

/把偏移地址为1000H到1063H内的100个数传送到1030H到1093H内，因为有重叠部分，源数据块地址小于目的数据块，所以从数据块末地址开始传送。/

FADR: MOV AL,[SI] ;重新设置SI（源指针）、DI地址

MOV [DI],AL

DEC SI ;SI-1，SI源变址寄存器

DEC DI ;DI-1，DI目标变址寄存器

DEC CX ;CX自减1

JNE FADR ; JNE条件转移指令，当ZF=0（运算结果不为0），即CX≠0，转至标 号FADR处执行，若ZF=1即CX=0则向下执行

JMP $ ;IMP无条件转移指令

CODE ENDS

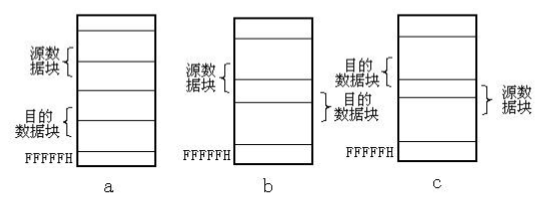
END START

功能：程序要求把内存中源数据块传送到目的数据块。

①a两个数据块分离的情况，数据的传送从据块的首址开始，或者从数据块的末址开始均可。②两个数据块有部分重叠的情况，要先把数据块重叠的部分转移。

b当源数据块首址小于目的块首址时，从数据块末地址开始传送数据；

c当源数据块首址大于目的块首址时，从数据块首地址开始传送数据。



# 实验五 十六进制转 BCD 实验

;将 AX=0FFFFH=65535 拆为 5 个 BCD 码（非压缩的）,并存入 RESULT 开始的 5 个单元

CODE SEGMENT

ASSUME CS:CODE, DS:DATA

START:

MOV AX, DATA ;设置段寄存器

MOV DS, AX

MOV DX,0000H ;DX清零

MOV AX, 65535 ;65535送入AX中

MOV CX, 10000 ; 10000=2710H

DIV CX ;DIV无符号数除法指令，AX/CX，余数放在DX里，商放在AL里

MOV RESULT, AL ; 把商从AL中送到RESULT，得 WAN 位数 06

MOV AX,DX ;把余数从DX中送到AX

MOV DX,0000H ; 先把DX的数据放在AX里，DX再清零，则可以达到扩充的效果，变为32位

MOV CX, 1000

DIV CX ;AX/CX

MOV RESULT+1, AL ; 除以 1000=03E8H, 得 QIAN 位数 5

MOV AX,DX

MOV DX,0000H

MOV CX, 100

DIV CX

MOV RESULT+2, AL ; 除以 100, 得 BAI 位数 5

MOV AX,DX

MOV DX,0000H

MOV CX, 10

DIV CX

MOV RESULT+3, AL ; 除以 10, 得 SHI 位数 3

MOV RESULT+4, DL ; 得 GE 位数 5

JMP $

CODE ENDS

DATA SEGMENT

**RESULT Db 5 DUP(?)** ; 问号表示内存值还没有初始化，

为5个字节型的数据保留存储单元

DATA ENDS

END START

连续输出06 05 05 03 05

MOV AX, 00714h ;段地址开始于0714:0000无偏移

功能：

# 实验六 由 1 到 100 求和实验

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

BEG: MOV AX,DATA

MOV DS,AX

MOV SI,OFFSET total ;把total的偏移地址送到SI

MOV CX,100 ;CX=100作为循环计数用

MOV AX,0 ;AX清零，用来存放运算结果

MOV BX,1 ;BX=00001H

add\_100: ADD AX,BX ;AX+BX，累加的结果存入AX中

INC BX ;BX自加1（通过循环一次增到101,101不参与运算）

LOOP add\_100 ;对add\_100进行循环，循环次数一共为100次

MOV [SI],AX ;循环结束后，把AX的结果送到SI中

JMP $

CODE ENDS

DATA SEGMENT

total DW 0000H ; 保留了两个字节的存储单元

DATA ENDS

END BEG

功能：由是 1 到 100 的 100 个数想加，通过CX进行循环计数，累加器AX进行累加求和得total=1+2+3+4........+99+100 = 5050 =13BA H

# 实验七 数据排列实验

CODE SEGMENT

ASSUME CS:CODE,DS:DATA,SS:STACK

START:

MOV AX,DATA

MOV DS,AX

MOV DX,COUNT-1 ; COUNT-1=6-1=5送到DX，因为数组长度为6即有6个数据，用冒 泡排序法只需要比较5次即可

MOV BL,0FFH ;BX用于判断冒泡过程是否完成，赋判断指标初值0FFH

AGAINO: CMP BL, 0 ;BL与0比较，判断是否完成冒泡排序

JE DONE ;若ZF=1，即两束相等，则跳转到DONE，如果没有则继续向下执行

XOR BL,BL ;BL进行取反为00H

MOV CX,DX ;循环次数

/CX作为LOOP循环的计数，第一轮冒泡过程要比较5次，此后冒泡过程分别比较4/3/2/1次，每比较完一次，把较小的数据向低一位送去，直到最小的数据处于数组的第一位/

MOV SI,COUNT-1 ;SI=5每轮冒泡比较都从最后一个数开始与之前一个数进行比较

AGAIN1: MOV AL,ARRAY[SI] ;取数组中的第SI位数据放到AL（最低位为第0位）

CMP AL,ARRAY[SI-1] ;AL中的数据即第SI位数据与数组中的第SI-1位数据进行比较

JAE UNCH ;（CMP是减法比较）CF=0第SI位大于或等于第SI-1位则跳转到UNCH，否则，为小于第SI-1位数据继续向下执行

EXCH: XCHG ARRAY[SI-1],AL ; 交换两个数据的位置， AL中存放较大的数

MOV ARRAY[SI],AL ;把AL中的数送到第SI位

MOV BL,0FFH ;当发生交换时，让BL=0FFH

UNCH: DEC SI ;SI自减1，

LOOP AGAIN1 ;循环执行AGAIN1

DEC DX ;CX=DX-1

JNZ AGAINO ;ZF=0级CX≠1跳转到AGAINO

DONE: JMP $

CODE ENDS

DATA SEGMENT

ARRAY DB 25,46,3,75,5,30 ; 25,46,3,75,5,30 = 19H,2EH,03H,4BH,05H,1EH

COUNT EQU $-ARRAY ; $-ARRAY表示这个数组的长度，为6，所以COUNT=6

DATA ENDS

STACK SEGMENT PARA STACK 'STACK' ；定义堆栈段

DB 60 DUP(?) ;为60个字节型变量保留内存单元

STACK ENDS

END START

功能：用“冒泡排序”法对25,46,3,75,5,30这一数组进行由小到大的排列，将一个数与前面的数进行比较，如果前面的数比较大，则进行交换，如此将所有数比较一遍后，最大的数就会在数列的最后面。再进行下一轮比较，找出第二大数据，如此下去，直到全部数据由小到大排列完。