DATA SEGMENT ;定义数据段

ARRAY DB 22,46,32,72,84,16,156 ;定义一串数据

MAXN1 DB 0 ;将存放最大值的变量初始化为0

DATA ENDS

CODE SEGMENT ;定义代码段

ASSUME DS:DATA,CS:CODE ;说明代码段、数据段

START:

MOV AX,DATA

MOV DS,AX ;给DS赋初值

MOV CX,6 ;置循环控制数

LEA DI,ARRAY ;将ARRAY表示的偏移地址送到DI

MOV DL,[DI] ;将第一个操作数送到寄存器中

JCXZ LAST

AGIN:

INC DI

CMP DL,[DI]

JAE NEXT

MOV DL,[DI]

NEXT:

LOOP AGIN

LAST:

MOV MAXN1,DL

MOV AH,4CH

INT 21H

CODE ENDS

END START

; 刚开始用字作为单位存储，且INC只一次，显然不够。后来发现可以用字节存储

2

DATA SEGMENT ;定义数据段

ARRAY DB 22,46,32,72,84,16,156 ;定义一串数据

MAXN1 DB 0 ;将存放最大值的变量初始化为0

DATA ENDS

CODE SEGMENT ;定义代码段

ASSUME DS:DATA,CS:CODE ;说明代码段、数据段

START:

MOV AX,DATA

MOV DS,AX ;给DS赋初值

MOV CX,6 ;置循环控制数

LEA DI,ARRAY ;将ARRAY表示的偏移地址送到DI

MOV DL,[DI] ;将第一个操作数送到寄存器中

JCXZ LAST

AGIN:

INC DI

CMP DL,[DI]

JGE NEXT ;带符号数比较，DL>[DI]时跳转，跳过更新环节

MOV DL,[DI]

NEXT:

LOOP AGIN

LAST:

MOV MAXN1,DL

MOV AH,4CH

INT 21H

CODE ENDS

END START

; 刚开始用字作为单位存储，且INC只一次，显然不够。后来发现可以用字节存储

;HEX:16 2E 10 48 54 9C 只有最后一个数最小

3

DATA SEGMENT ;定义数据段

ARRAY DW 58,25,45,73,64,43 ;定义一串数据

SUM DW 0 ;将存放最大值的变量初始化为0

DATA ENDS

CODE SEGMENT

ASSUME DS:DATA,CS:CODE ;说明代码段、数据段

START:

MOV AX,DATA

MOV DS,AX ;给DS赋初值

MOV AX,0;

MOV CX,(LENGTH DATA)-1 ;置循环控制数,比数组元素数少1

;报错：must be associated with data

LEA DI,ARRAY ;将ARRAY表示的偏移地址送到DI

AGIN:

ADD AX,DI

JCXZ LAST

DEC CX

INC DI

INC DI

LOOP AGIN

LAST:

MOV SUM,AX

MOV AH,4CH

INT 21H

CODE ENDS

END START

3改

DATA SEGMENT ;定义数据段

ARRAY DW 58,25,45,73,64,43 ;定义一串数据

SUM DW 0 ;将存放最大值的变量初始化为0

DATA ENDS

CODE SEGMENT

ASSUME DS:DATA,CS:CODE ;说明代码段、数据段

START:

MOV AX,DATA

MOV DS,AX ;给DS赋初值

MOV AX,0;

MOV CX,6

;报错：must be associated with data

LEA DI,ARRAY ;将ARRAY表示的偏移地址送到DI

AGIN:

ADD AX,[DI];刚开始没加方框

INC DI

INC DI

LOOP AGIN ;该条指令执行后CX自动-1，手动写反而出错

LAST:

MOV SUM,AX

MOV AH,4CH

INT 21H

CODE ENDS

END START

;当以CX=1状态来到LOOP时，LOOP不会继续循环，而是将CX-1然后跳出循环

4

DATA SEGMENT ;定义数据段

D1 DB 08,04,03,03,05

D2 DB 09

SUM DB 6 DUP(0) ;将存放最大值的变量初始化为0

DATA ENDS

CODE SEGMENT

ASSUME DS:DATA,CS:CODE ;说明代码段、数据段

GO:

MOV AX,DATA

MOV DS,AX ;给DS赋初值

MOV SI,D2

DEC SI ;指向D1的最后一个数据05

MOV BL,D2;;把乘数放入BL

MOV DI,OFFSET SUM ;目标存放地址放入DI

MOV DX,0;因后续进位存放在DL，先清零

MOV CX,5 ;做乘法的次数一共5次

NEXT:

MOV AL,[SI] ;将被乘数从低位移入

DEC SI

MUL BL

AAM;乘法的ASCII调整,AX中存放非压缩乘积，乘积的低位在AL，高位在AH

ADD AL,DL ;加上进位

AAA ;作加法的ASCII调整，把可能的进位加到AH上

MOV DL,AH;低位向高位的进位要留起来

MOV [DI],AL ;低位存入目的存储单元

INC DI

LOOP NEXT

MOV AH,4CH

INT 21H

CODE ENDS

END GO

4改

DATA SEGMENT ;定义数据段

D1 DB 08,04,03,03,05

D2 DB 09 ;这两个十进制数相乘，不超过6位

SUM DB 6 DUP(0) ;将存放最大值的变量初始化为0,预先不知道结果多长，

DATA ENDS

CODE SEGMENT

ASSUME DS:DATA,CS:CODE ;说明代码段、数据段

GO:

MOV AX,DATA

MOV DS,AX ;给DS赋初值

MOV SI,OFFSET D1 ;忘加offset

MOV BL,[D2];;把乘数放入BL,开始时忘加方框

MOV DI,OFFSET SUM ;目标存放地址放入DI

MOV DX,0;因后续进位存放在DL，先清零

MOV CX,5 ;做乘法的次数,设置为SUM单元的长度

NEXT:

MOV AL,[SI] ;将被乘数从低位移入

INC SI

MUL BL

AAM;乘法的ASCII调整,AX中存放非压缩乘积，乘积的低位在AL，高位在AH

ADD AL,DL ;加上进位

AAA ;作加法的ASCII调整，把可能的进位加到AH上

MOV DL,AH;低位向高位的进位要留起来

MOV [DI],AL ;低位存入目的存储单元

INC DI

LOOP NEXT

MOV [DI],AH

MOV AH,4CH

INT 21H

CODE ENDS

END GO

LENGTH和SIZE D1都返回1？？？

5

DATA SEGMENT ;定义数据段

STR1 DB 30H,31H,32H,33H,34H,35H,36H,37H,38H,39H,40H,41H,42H,43H,44H,45H

STR2 DB 16 DUP(?)

DATA ENDS

CODE SEGMENT

ASSUME DS:DATA,CS:CODE ;说明代码段、数据段

GO:

MOV AX,DATA

MOV DS,AX ;给DS赋初值

MOV ES,AX ;第一次忘了给ES赋初值

MOV SI,OFFSET STR1

MOV DI,OFFSET STR2

MOV CX,16

CLD

REP MOVSB

MOV AH,4CH

INT 21H

CODE ENDS

END GO

6

DATA SEGMENT ;定义数据段

D1 DB 08,04,03,03,05

D2 DB 09 ;这两个十进制数相乘，不超过6位

SUM DB 6 DUP(0) ;将存放最大值的变量初始化为0,预先不知道结果多长

OTHER DB 0DH,0AH,'$' ;为了使用字符串输出指令，加上回车、换行和$

;第一次多打了一个逗号，多出来一个00

DATA ENDS

CODE SEGMENT

ASSUME DS:DATA,CS:CODE ;说明代码段、数据段

GO:

MOV AX,DATA

MOV DS,AX ;给DS赋初值

MOV SI,OFFSET D1 ;忘加offset

MOV BL,[D2];把乘数放入BL,开始时忘加方框

MOV DI,OFFSET OTHER ;目标存放地址放入DI

DEC DI;指到SUM单元的最后一位，低位放后面，便于后续正序输出

MOV DX,0;因后续进位存放在DL，先清零

MOV CX,5 ;做乘法的次数

NEXT:

MOV AL,[SI] ;将被乘数从低位移入

INC SI

MUL BL

AAM;乘法的ASCII调整,AX中存放非压缩乘积，乘积的低位在AL，高位在AH

ADD AL,DL ;加上进位

AAA ;作加法的ASCII调整，把可能的进位加到AH上

MOV DL,AH;低位向高位的进位要留起来

ADD AL,30H;ASCII数字字符的范围是30H(0)~39H(9)

MOV [DI],AL ;低位存入目的存储单元

DEC DI

LOOP NEXT

ADD AH,30H;这时候已经是最后的高位了，要加30H来从非压缩BCD码转换到ASCII码

MOV [DI],AH;此时DI已经指到最高位

CMP AH,30H

JNE LAST

INC DI ;若最高位为0.从次高位开始输出（结果要么是5位要么是6位）

LAST:

MOV DX,DI; 字符串的首址放到DX，准备输出

MOV AH,9 ;调用输出字符串的功能

INT 21H;输出字符串

MOV AH,4CH

INT 21H

CODE ENDS

END GO

7

DATA SEGMENT ;定义数据段

STR1 DB ‘HELLO12345’

RESULT1 DW 0

RESULT2 DB 0

DATA ENDS

EXTRA SEGMENT;定义附加段

STR2 DB ‘HELLO1?345’ ;两串不完全相同

EXTRA ENDS

CODE SEGMENT

ASSUME DS:DATA,CS:CODE,ES:EXTRA ;说明代码段、数据段

GO:

MOV AX,DATA

MOV DS,AX ;给DS赋初值

MOV AX,EXTRA

MOV ES,AX;给ES赋初值

MOV SI,OFFSET STR1;忘加offset

MOV DI,OFFSET STR2

REPE CMPSB

JZ EQQ

DEC SI

MOV BX,SI

MOV [RESULT1],BX

MOV AL,[SI]

MOV [RESULT2],AL

JMP LAST

EQQ:

MOV [RESULT1],0

LAST:

MOV AH,4CH

INT 21H

CODE ENDS

END GO

7改

第一次实验：两串不相同

DATA SEGMENT ;定义数据段

STR1 DB 'HELLO12345'

RESULT1 DB 0

RESULT2 DB 0

DATA ENDS

STACK SEGMENT STACK

NO DB 0

STACK ENDS

EXTRA SEGMENT;定义附加段

STR2 DB 'HELLO1?345' ;两串不完全相同

EXTRA ENDS

CODE SEGMENT

ASSUME DS:DATA,CS:CODE,ES:EXTRA ;说明代码段、数据段

GO:

MOV CX,10;第一次忘了置次数

MOV AX,DATA

MOV DS,AX ;给DS赋初值

MOV AX,EXTRA

MOV ES,AX;给ES赋初值

MOV SI,OFFSET STR1;

MOV DI,OFFSET STR2

REPE CMPSB

JZ EQQ;若不完全一样，就不跳转

DEC SI;恢复到不一样的地方

MOV BX,SI;传送不一样的字节地址

MOV [RESULT1],BL ;只有变量才可以用LOW指令

MOV AL,[SI]

MOV [RESULT2],AL

JMP LAST

EQQ:

MOV [RESULT1],0;若完全相同，RESULT1置0

LAST:

MOV AH,4CH

INT 21H

CODE ENDS

END GO

第二次实验：两串相同

DATA SEGMENT ;定义数据段

STR1 DB 'HELLO12345'

RESULT1 DB 0

RESULT2 DB 0

DATA ENDS

STACK SEGMENT STACK

NO DB 0

STACK ENDS

EXTRA SEGMENT;定义附加段

STR2 DB 'HELLO12345' ;两串不完全相同

EXTRA ENDS

CODE SEGMENT

ASSUME DS:DATA,CS:CODE,ES:EXTRA ;说明代码段、数据段

GO:

MOV CX,10;第一次忘了置次数

MOV AX,DATA

MOV DS,AX ;给DS赋初值

MOV AX,EXTRA

MOV ES,AX;给ES赋初值

MOV SI,OFFSET STR1;

MOV DI,OFFSET STR2

REPE CMPSB

JZ EQQ;若不完全一样，就不跳转

DEC SI;恢复到不一样的地方

MOV BX,SI;传送不一样的字节地址

MOV [RESULT1],BL ;只有变量才可以用LOW指令

MOV AL,[SI]

MOV [RESULT2],AL

JMP LAST

EQQ:

MOV [RESULT1],0;若完全相同，RESULT1置0

LAST:

MOV AH,4CH

INT 21H

CODE ENDS

END GO