

## 汇编语言模拟试题及答案

一，单项选择题（在每小题的四个备选答案中，选出一个正确的答案，并将其号码填在题干后的括号内，每小题 1 分，共 20 分）

1. 指令 JMP FAR PTR DONE 属于            参考答案为:C

- A. 段内转移直接寻址
- B. 段内转移间接寻址
- C. 段间转移直接寻址
- D. 段间转移间接寻址

[解析]略

2. 下列叙述正确的是            参考答案为:C

- A. 对两个无符号数进行比较采用 CMP 指令，对两个有符号数比较用 CMPS 指令
- B. 对两个无符号数进行比较采用 CMPS 指令，对两个有符号数比较用 CMP 指令
- C. 对无符号数条件转移采用 JAE/JNB 指令，对有符号数条件转移用 JGE/JNL 指令
- D. 对无符号数条件转移采用 JGE/JNL 指令，对有符号数条件转移用 JAE/JNB 指令

[解析]对于无符号数和有符号数的比较都采用 CMP 指令；

CMPS 指令是串比较指令；

对两个无符号数的条件转移指令应是：JAE、JNB、JBE、JNA；

对两个有符号数的条件转移指令应是：JGE、JNL、JLE、JNG。

3. 一个有 128 个字的数据区，它的起始地址为 12ABH: 00ABH，请给出这个数据区最末一个字单元的物理地址是            参考答案为:C

- A. 12CSBH
- B. 12B6BH
- C. 12C59H
- D. 12BFEH

[解析]末字与首字相隔  $(128-1=)$  127 个字，且每个字占用 2 个字节，因此末字单元的物理地址应为：

首字单元的物理地址 +  $(128-1) \times 2$

即  $12ABH \times 10H + 00ABH + (128-1) \times 2 = 12C59H$ 。

4. 在下列指令的表示中，不正确的是            参考答案为:C

- A. MOV AL, [BX+SI]
- B. JMP SHORT DONI
- C. DEC [BX]
- D. MUL CL

[解析]当只有一个存储器操作数时，这个操作数的类型不明确，例如选项 C 中的 [BX]，没有明确的说明访问该存储单元的类型，此时存储器操作数就必须需用类型说明，如

DEC BYTE PTR [BX] 或 DEC WORD PTR [BX]

但是在题目的选项 C 中，没有指出存储器操作数类型，所以该指令是不正确的；而其它选项中的指令均是正确的。

5. 在进行二重循环程序设计时, 下列描述正确的是 参考答案为:A

- A. 外循环初值应置外循环之外; 内循环初值应置内循环之外, 外循环之内
- B. 外循环初值应置外循环之内; 内循环初值应置内循环之内
- C. 内、外循环初值都应置外循环之外
- D. 内、外循环初值都应置内循环之外, 外循环之内

[解析]循环参数置初值, 应在进入循环以前给出。

6. 条件转移指令 JNE 的测试条件为 参考答案为:A

- A. ZF=0
- B. CF=0
- C. ZF=1
- D. CF=1

[解析]略。

7. 8086CPU 在基址加变址的寻址方式中, 变址寄存器可以为 参考答案为:D

- A. BX 或 CX
- B. CX 或 SI
- C. DX 或 SI
- D. SI 或 DI

[解析]略

8. 已知 BX=2000H, SI=1234H, 则指令 MOV AX, [BX+SI+2]的源操作在 ( ) 中。 参考答案为:A

- A. 数据段中偏移量为 3236H 的字节
- B. 附加段中偏移量为 3234H 的字节
- C. 数据段中偏移量为 3234H 的字节
- D. 附加段中偏移量为 3236H 的字节

[解析]指令中的源操作数的寻址方式是相对基址变址寻址, 且由于其中的基址寄存器采用的是 BX, 所以源操作数是在数据段中, 其偏移量为:

$$BX+SI+2=2000H+1234H+2=3236H$$

9. 执行如下程序:

```
MOV AX, 0
MOV BX, 1
MOV CX, 100
A: ADD AX, BX
INC BX
LOOP A
HLT
```

执行后 (BX) = 参考答案为:C

- A. 99
- B. 100
- C. 101
- D. 102

[解析]1) 在进入循环之前, 有三条指令分别初始化寄存器: AX 为 0、BX 为 1、CX 为 100;

2) 由于循环的控制部分采用的是 LOOP 指令, 该指令有使 CX 减 1, 并判

断其是否为 0 的功能，所以可知该循环的执行次数为 100 次；

3) 循环体的主要功能是每次循环使 AX 中的内容累加 BX 的内容，并使 BX 的内容加 1。

4) 直到最后一次（即第 100 次）循环时，加到 AX 中的 BX 的内容为 100，且使 BX 加 1，即变为 101 后退出循环体。所以当执行完该段程序后，BX 的内容为 101。

5) 该段程序的功能是  $(AX) = 1 + 2 + \dots + 100 = 5050$

10. 上题的程序执行后  $(AX) =$  参考答案为:B

A. 5000

B. 5050

C. 5100

D. 5150

[解析]请参看上题中的分析。

正确答案：B。

11. 对于下列程序段：

AGAIN: MOV AL, [SI]

MOV ES: [DI], AL

INC SI

INC DI

LOOP AGAIN

也可用\_\_\_\_\_指令完成同样的功能。 参考答案为:A

A. REP MOVSB

B. REP LODSB

C. REP STOSB

D. REPE SCASB

[解析]该段程序的功能是将 SI 所指向的数据段中的数据顺序传送给 DI 所指向的附加段的一段数据区中，并且每次传送一个字节，即实现了字节串传送的功能，所以也可以使用 REP MOVSB 指令来完成同样的功能。

12. 下面指令序列执行后完成的运算，正确的算术表达式应是

MOV AL, BYTE PTR X

SHL AL, 1

DEC AL

MOV BYTE PTR Y, AL 参考答案为:D

A.  $y = x2 + 1$

B.  $x = y2 + 1$

C.  $x = y2 - 1$

D.  $y = x2 - 1$

[解析]该段程序的第一条指令的功能是将 X 传送给 AL；

第二条指令的功能是将 AL 的内容左移 1 位，即使得 AL 的内容扩大 2 倍；

第三条指令的功能是使 AL 的内容减 1；

第四条指令的功能是将 AL 传送给 Y；

综上所述该程序完成的运算是  $Y = 2X - 1$ 。

13. 在一段汇编程序中多次调用另一段程序，用宏指令比用子程序实现起来。 参  
考答案为:B

- A. 占内存空间小, 但速度慢
- B. 占内存空间大, 但速度快
- C. 占内存空间相同, 速度快
- D. 占内存空间相同, 速度慢

[解析]1) 宏指令简化程序的书写, 不节省程序存储空间, 但执行时间比子程序短。

2) 子程序简化程序的书写, 也节省程序的存储空间, 但执行时间比宏指令长。

14. 在程序执行过程中, IP 寄存器中始终保存的是            参考答案为:B

- A. 上一条指令的首地址
- B. 下一条指令的首地址
- C. 正在执行指令的首地址
- D. 需计算有效地址后才能确定地址

[解析]在程序运行过程中, 它始终指向下一条指令的首地址, 它与 CS 代码段寄存器联合确定下一条指令的物理地址。

15. PSW 寄存器中共有 \_\_\_\_\_ 位条件状态位, 有 \_\_\_\_\_ 位控制状态位。            参考答案为:A

- A. 6、3
- B. 3、6
- C. 8、4
- D. 4、8

[解析]PSW 有 6 位条件状态位: AF、CF、ZF、PF、SF、OF;

PSW 有 3 位控制状态位: IF、TF、DF

16. 下列指令执行时出错的是            参考答案为:A

- A. ADD BUF1, BUF2
- B. JMP DWORD PTR DAT [BX]
- C. MOV AX, [BX+DI] NUM
- D. TEST AL, 08H

[解析]ADD 指令的两个操作数不能同时是存储器操作数, 所以选项 A 中的指令是错误的。

17. 已知 (AX) = 1234H, 执行下述三条指令后, (AX) =            参考答案为:D

MOV BX, AX

NEG BX

ADD AX, BX

- A. 1234H
- B. 0EDCCH
- C. 6DCCH
- D. 0000H

[解析]第一条指令的功能是将 AX → BX, 即 BX 中的内容为 1234H;

第二条指令的功能是对 BX 求补, 即 BX 中的内容变为 -1234H;

第三条指令的功能是 AX + BX → AX, 即 1234H + (-1234H) = 0 → AX;

所以执行三条指令后, AX 的内容为 0。

18. 在下列指令中, \_\_\_\_\_ 指令的执行会影响条件码中的 CF 位。            参考答案为:D

- A. JMP NEXT
- B. JC NEXT
- C. INC BX

D. SHL AX, 1

[解析]JMP、JC 指令不任何影响状态标志位；

INC 指令执行结果除不影响 CF 外，其余状态标志位(AF、OF、PF、SF、ZF)都受影响。

SHL 指令使 CF=0, OF=0, 影响 ZF、SF、PF 标志位，AF 无定义。

19. 串指令中的目的操作数地址是由\_\_\_\_\_提供。 参考答案为:C

A. SS: [BP]

B. DS: [SI]

C. ES: [DI]

D. CS: [IP]

[解析]略

20. 将 DX 的内容除以 2，正确的指令是 参考答案为:C

A. DIV 2

B. DIV DX, 2

C. SAR DX, 1

D. SHL DX, 1

[解析]选项 A 中的指令执行的操作是: AX/2;

选项 B 中的指令书写格式不正确;

选项 C 中的指令执行的操作是:将 DX 的内容右移一位，将使 DX 的内容缩小为原先的 1/2，即将 DX 的内容除以 2;

选项 D 中的指令执行的操作是:将 DX 的内容左移一位，将使 DX 的内容扩大为原先的 2 倍，即将 DX 的内容乘以 2。

[解析](1)指令: SHR DX, 1 使 DX 的内容向右移动了一位，因此 DX 的内容变为 0000'0000'0101'1100B，即 005CH。

(2)指令: RCR DL, 1 使 DL 的内容向右循环带进位移动了一位，DH 的内容不变，因此 DX 的内容变为 0000'0000'1101'1100B，即 00DCH。

(3)指令: RCL DX, CL 使 DX 的内容向左循环带进位移动了 3 位，因此 DX 的内容变为 0000'0101'1100'1100B，即 05CCH。

2. 写出完成下述功能的程序段:

(1) 传送 25H 到 AL 寄存器

(2) 将 AL 的内容乘以 2

(3) 传送 15H 到 BL 寄存器

(4) AL 的内容乘以 BL 的内容

并请写出最后 (AX) =?

答: (1) MOV AL, 25H

(2) SHL AL, 1

(3) MOV BL, 15H

(4) MUL BL

最后, (AX) = 612H = 1554

[解析]1) 将 AL 的内容左移 1 位，将使它的内容扩大 2 倍，即使 AL 的内容乘以 2。

2) 最后, AX 的内容应为 25H\*2\*15H=612H=1554

3. 写一程序段，求双字长数 DX:BX 的相反数。

答: NEG DX

NEG AX

SBB DX, 0

[解析]求双字长数 DX:AX 相反数, 即对其求补, 所以应实现的操作是:

$(DX:AX) \leftarrow 0 - (DX:AX)$

即 0:0

-) DX:AX

将上面的算式分解后, 即为对 DX 求补, 对 AX 求补, 并从 DX 中减去将 AX 求补时产生的借位。所以相应的指令就是:

NEG DX

NEG AX

SBB DX, 0

三、程序分析题 (每小题 6 分, 共 36 分)

1. 现有如下程序段:

```
DATA SEGMENT                                DA3 EQU 1234H
ORG 0020H                                    DA4 EQU $-DA2
DA1 DW 12H, 23H                              DA5 DB 31H, 32H
DA2 DB 11H, 21H                              DATA ENDS
```

注意: 程序是从左到右看, 先把左边的程序段看完了再看右边的程序段。下面的程序段以相同的方法进行读解。

请回答: (1) 该程序段已占有的存储字节数是多少?

(2) 符号名 DA4 的值是多少? 符号名 DA5 的值是多少?

答: (1) 该程序段已占有的存储字节数是 8 个字节。

(2) 符号名 DA4 的值为 2, 符号名 DA5 的值是 0026H。

[解析]1) 伪指令 ORG 0020H, 设置存储单元起始偏移地址为 0020H;

2) 符号名 DA1 的值 (即所对应存储单元的偏移地址) 是 0020H, 该数据区的类型为字, 有 2 个字节数据, 所以共占用 4 个字节;

3) 符号名 DA2 的值 (即所对应存储单元的偏移地址) 是 0024H, 该数据区的类型为字节, 有 2 个字节数据, 所以共占用 2 个字节;

4) 经 EQU 定义的符号名 DA3、DA4 不占用存储空间, “\$” 用来表示当前地址计数器的值 (0026H), 可它们的值分别为 DA3=1234H, DA4=2。

5) 符号名 DA5 的值 (即所对应存储单元的偏移地址) 是 0026H, 该数据区的类型为字节, 有 2 个字节数据, 所以共占用 2 个字节;

6) 综上所述, 该程序段共占有的存储字节数 8 个字节。

2. 执行下列程序段后, 回答问题。

```
ORG 1221H MOV BX,          OFFSET NUM
NUM DW 12H                  MOV CL, BYTE PTR NUM+1
...                          MOV CH, TYPE NUM
MOV AX, NUM                 HLT
```

请问: (AX) = \_\_\_\_\_, (BX) = \_\_\_\_\_, (CX) = \_\_\_\_\_。

答: (AX)=0012H (BX)=1221H (CX)=0200H

[解析]1) 指令 MOV AX, NUM 的功能是将 NUM 单元的字内容 12H 传送给 AX, 使得 (AX)=0012H;

2) 指令 MOV BX, OFFSET NUM 的功能是将 NUM 单元的偏移地址 1221H 传送给 BX, 使得 (BX)=1221H;

3) 指令 MOV CL, BYTE PTR NUM+1 的功能是将 NUM+1(1222H)单元的字节内容(即 NUM 字单元的高字节)00H 传送给 CL, 使得(CL)=00H;

4) 指令 MOV CL, TYPE NUM 的功能是将 NUM 字单元的类型字节数 2 传送给 CH, 使得(CH)=02H;

5) 综合第 4 和第 5 点, 可知 CX 寄存器的内容为 0200H。

3. 阅读如下程序:

```
ADRR DB XXH, XXH, ....., XXH      CMP ADRR[BX], 0
NUM EQU $ -ADRR                     LOOPZ DON
RETT DB ?                            JNZ NEXT
...                                  MOV BL, OFFH
MOV CX, NUM                          NEXT: MOV RETT, BL
MOV BX, -1                           HLT
DON: INC BX
```

请问: (1) 该程序完成的功能是什么?

(2) 程序执行后指出 RETT 字节单元中的内容是什么?

答: (1) 在以 ADRR 为起始地址的数组中, 查找第一个非 0 数据。

(2) 若找到则将第一个非 0 数据在数组中的偏移量送入 RETT 单元中; 若无非 0 数据, 则将 OFFH 送入 RETT 单元中。

[解析]1) 程序的前 3 条指令, 用于定义数据:

以 ADRR 为起始地址的数组;

NUM 的值为 ADRR 数组的数据个数;

RETT 是一个空的字节单元

2) 该程序的主体是一个循环, 该循环的主要功能是在数组 ADRR 中第一个非 0 数据。如果是由于找到, 则将第一个非 0 数据在数组中的偏移量送入 RETT 单元中; 若没找到非 0 数据, 则将 OFFH 送入 RETT 单元中。

4. 现有一子程序:

```
SUB1 PROC                                PLUS: TEST BL, 80H
TEST AL, 80H (检测 AL 中的数值最高位) JE EXITO
JE PLUS                                  XCHANGE: XCHG AL, BL
TEST BL, 80H                            EXITO: RET
JNE EXITO                               SUB1 ENDP
JMP XCHANGE
```

试回答: (1) 子程序的功能是什么?

(2) 若调用子程序前 AL=9AH, BL=77H, 则返回主程序时, AL=\_\_\_\_\_, BL=\_\_\_\_\_。

答: (1) 子程序的功能是判别 AL 和 BL 中的数是否为异号数, 当为异号时交换, 否则不变。

(2) AL=77H, BL=9AH

[解析]1) 数据是以二进制的形式存储在计算机中的。数据对应的二进制数的最高位是数据的符号位: 若为 0 表示该数为正数, 若为 1 表示该数为负数;

2) TEST AL, 80H 和 TEST BL, 80H 指令是在测试 AL 和 BL 的最高位 (即符号位) 的状态是 1 还是 0。

3) 若两个数的最高位 (即符号位) 的状态不同的时候 (即两数为异号), 则执行 XCHANGE 标号所对应语句, 即交换 AL 和 BL 中的内容。

4) 否则若两个数的最高位(即符号位)的状态相同的时候(即两数为同号),则执行 EXIT0 标号所对应语句,程序结束,即 AL 和 BL 中的内容不被。

5) 所以第二问中, AL=9AH,BL=77H: AL 中的数据最高位为 1, BL 中的数据最高位为 0, 即两个数为异号, 所以要交换 AL 和 BL 中的内容, 即 AL=77H,BL=9AH。

5. 阅读程序段:

BLOCK DB 20H, 1FH, 08H, 81H, OFFH...	LOP1: CMP AL, [SI+1]
RESULT DB ?	JNG NEXT
...	MOV AL, [SI+1]
START: LEA SI, BLOCK	NEXT: INC SI
MOV CX, [SI]	LOOP LOP1
INC SI	MOV RESULT, AL
MOV AL, [SI]	HLT

请问: (1) 该程序完成的功能是什么?

(2) 该程序循环的次数是多少?

答: (1) 该程序完成的功能是从 20H (32) 个有符号数中找出最小的数送入 RESULT 单元中。

(2) 该程序的循环次数是 1FH (31)。

[解析]1) 指令 LEA SI, BLOCK 的功能是使 SI 指向了 BLOCK 数组的首单元;

2) 指令 MOV CX, [SI] 的功能是将 SI 指向单元的内容 20H 传送给 CX, 即(CX)=20H; 其后又有指令 DEC CX, 使得 CX 的内容变为 1FH; 再看循环体的控制部分采用的指令是 LOOP, 可知, 该循环的执行次数为 1FH 次;

3) 循环体的主要功能是将 AL 中的内容与[SI]所指向数组的每一个数据进行比较, 从数组中选出最小数保存在 AL 中。

4) 由于比较后采用的转移指令是 JNG, 所以该程序处理的显然是一组有符号的数据。

6. 阅读如下程序段, 回答所提出的问题, 假定 X1, X2, XN 为无符号数。

```
ORG 100H
BLOK DW X1, X2, ..., XN
COUNT EQU ( $-BLOK ) /2
RESULT DW COUNT, ?
LEA BX, BLOK-2
MOV CX, RESULT
XOR AX, AX
GOON: ADD BX, 2
CMP AX, [BX]
JAE NEXT
MOV AX, [BX]
NEXT: LOOP GOON
MOV RESULT+2, AX
程序段执行完, 回答下述问题
RESULT+2 字存储单元的内容=_____
BX=_____
程序循环次数=_____
```

答: RESULT+2 字存储单元的内容是 MAX(X1, X2, ... XN)



$BX = 100H + 2 \times (COUNT - 1)$

程序循环次数 = COUNT

[解析]1) 该程序段的开始 4 条指令定义了以下数据:

以 BLOK 为起始地址的字数组, 该数组首单元的偏移地址 (即 BLOK) 为 100H;  
COUNT 的值为 BLOK 字数组的数据个数;

RESULT 存储区的第一个字的内容为 COUNT, 即 BLOK 字数组的数据个数; 第二个字的内容为空。

2) 循环体的主要功能是从 BLOK 字数组中找出最大数送入 RESULT 的第二个字单元中;

3) 循环次数为数组中数据的个数, 即 COUNT 个。

4) 当循环结束后, BX 中将存放最后一个字数据的偏移地址: 首字偏移地址 + 末字与首字的相隔的字节数, 即  $100H + 2 \times (COUNT - 1)$ 。

四, 程序填空题 (注意: 下列各小题中, 每空只能填一条指令! 每小题 6 分, 共 12 分)

1. 在表 TABLE 处存放着 N 个无符号字节数, 求表中前 10 个字节数的总和并保存在 AX 寄存器中。如果总和值超过 AX 寄存器的容量, 则置 AX 寄存器内容不变。请在程序中的空格处填入适当指令使其完整。

LEA BX, TABLE	(2)
MOV CX, 10	JC END0
MOV SUM, AX	DON: INC BX
MEXT: (1)	(3)
ADC AH, 0	ENDO: HLT
JNC DON	

答: (1) ADD AL, [BX]

(2) MOV AX, SUM

(3) LOOP NEXT

[解析]1) 程序的前三条指令的功能分别是:

将表 TABLE 的首址传送给 BX;

将循环次数传送给计数器 CX;

将 AX 的原值保存在 SUM 单元中;

以上指令实现了设置循环初始参数的功能。

2) 标号 NEXT 所对应的第一个空白处就是循环体的第一条指令。根据题目的要求——求表中前 10 个字节数的总和, 可知循环每次应实现将表 TABLE 中的数据按字节加法的方式进行累加到 AL 中, 并且将字节加法产生的进位加到 AH 中。所以第一个空白处应填入指令:

ADD AL, [BX]

3) 程序的第五条指令实现的是判断求得的和是否超过 AX 寄存器的容量。

4) 如果有进位 (即已超过 AX 寄存器的容量), 则执行第二个空白处的指令, 根据题目当出现此种情况时, 使 AX 内容不变, 即恢复其原值, 因此第二个空白处应填入指令: MOV AX, SUM。

5) 如果没有进位 (即未超过 AX 寄存器的容量), 则转 DON。程序将使 BX 的内容加 1, 即使其指向下一个表中的字节数据。接下来就应该将计数器 CX 减 1, 并判断是否继续下一次循环。因此, 第三个空白处应填入指令: LOOP NEXT。

2. BEG 为起始地址的存储区域中存放有 100 个字节数据, 现将其首、尾颠倒过来, 重

新排放这 100 个字节数据。请在程序中的空格处填入适当的指令使其完整。

```
MOV CX, 32H
LEA SI, BEG
MOV DI, SI
(1)
DON: MOV AL, [SI]
(2)
MOV [SI], AL
(3)
DEC DI
LOOP DON
HLT
```

答:(1) ADD DI, 99

(2) XCHG AL, [DI]

(3) INC SI

[解析]1) 题目要求: 将以 BEG 为起始地址的区域中的 100 个字节数据首尾颠倒存放。例如, 如果原先为 X1,X2, X3,..., X98, X99, X100, 则首尾颠倒后将变为 X100, X99, X98,...,X3, X2,X1。

2) 颠倒的方法是: 将 X1 与 X100 对调、X2 与 X99 对调、X3 与 X98 对调、....., 即将前一半的数据与后一半相应位置的数据对调。这个过程每次处理两个数据, 所以共有  $(100/2=)$  50 对数据对调。

3) 根据以上分析可知, 这个循环将进行 50 (32H) 次, 每次将相应的两个数据对调, 这两个数据可分别由两个寄存器来指向。

4) 阅读程序可知, SI 寄存器指向前一半中的某个数据, DI 寄存器指向后一半中对应的数据。在循环的初始化部分, 使应计数器 CX 的值为 50 (或 32H), SI 指向第一个数, DI 指向最后一个数, 所以第一个空白处应填入指令: ADD DI, 99。

5) 在循环体中, 应首先交换 SI 和 DI 所指向单元的内容, 然后使指针 SI 加 1、DI 减 1, 所以第二、三个空白处应填入指令:

① XCHG AL, [DI]

② INC SI

五, 编制程序题 (第 1 小题 6 分, 第 2 小题 14 分, 共 20 分)

1. 试编写一个汇编语言程序, 要求对键盘输入的小写字母用大写字母显示出来。

答: 1. BEGIN: MOV AH, 1

```
INT 21H
CMP AL, 'a'
JB STOP
CMP AL, 'z'
JA STOP
SUB AL, 20H
MOV DL, AL
MOV AH, 2
INT 21H
JMP BEGIN
```

STOP: RET

[解析]1) 该程序设计的主体应是一个循环——当输入字符的是小写字母 (a ~ z) 时, 就将其转换为大写字母来显示, 此过程循环进行, 直到当输入的不是小写字母时, 程序结束。

2) 输入一个字符的可使用 DOS 的 1 号功能实现:

MOV AH, 1

INT 21H

该 DOS 调用执行完后, 会将输入的字符的 ASCII 码值送入 AL 寄存器。

3) 判断输入的是否为小写字母 (a ~ z) 的方法:

CMP AL, 'a'

JB STOP; 不是小写字母时, 转 STOP

CMP AL, 'z'

JA STOP; 不是小写字母时, 转 STOP

...; 是小写字母时, 将其转换为大写显示

STOP: RET; 不是小写字母时, 程序结束

4) 小写字母比大写字母的 ASCII 码值大 20H, 因此指令 SUB AL, 20H, 将使 AL 中的内容变为大写字母的 ASCII 码值。

5) 输出一个字符的可使用 DOS 的 2 号功能实现:

MOV DL, AL

MOV AH, 2

INT 21H

2. 设在内存单元 SQTAB 首址开始存放了 0~15 的平方数表。要求用直接查表法编一程序, 求出 A 单元中给定数 ( $\leq 15$ ) 的平方值送 B 单元保存。

答: DATA SEGMENT

SQTAB DB 0, 1, 4, 9, 16, 25, 36, 49, 64, 81

DB 100, 121, 144, 169, 196, 225

A DB 12

B DB ?

DATA ENDS

CODE SEGMENT

ASSUME CS: CODE, DS: DATA

START PROC FAR

PUSH DX

XOR AX, AX

PUSH AX

MOV AX, DATA

MOV DS, AX

MOV BX, OFFSET SQTAB

MOV AL, A

XLAT

MOV B, AL

RET

START ENDP

CODE ENDS

END START

[解析]1)根据题目的要求，应首先建立 0~15 的平方数表以及 A 和 B 单元：

SQTAB DB 0, 1, 4, 9, 16, 25, 36, 49, 64, 81；平方数表

DB 100, 121, 144, 169, 196, 225

A DB 12；给定数（≤15）

B DB ?；存放 A 中数的平方值的结果单元

2)A 中给定的数恰好是该数平方值在平方数表中的相对位置，因此，可通过换码指令 XLAT 来实现从表中求平方值：

MOV BX, OFFSET SQTAB

MOV AL, A

XLAT