

一、填空题

1. 中央处理器包括 运算器、控制器、寄存器组。
2. 物理地址是指实际的 20 位主存单元地址，每个存储单元对应唯一的物理地址。
3. 用二进制代码书写指令和程序的语言，称为 机器语言。
4. 计算机中的指令由 操作码 和 操作数 两部分组成。
5. 把 A 和 B 两个寄存器的内容进行异或运算，若运算结果是 0，那么 A、B 寄存器的内容必定相同。
6. AL, BL 中存放着压缩的 BCD 数，(AL)=18H, (BL)=47H, 执行如下两条指令 ADD AL, BL 和 DAA 后，(AL)=65H。
7. 当指令“SUB AX, BX”执行后，CF=1，说明最高有效位 有借位；对 有符号 数，说明操作结果溢出。
8. FLAG 状态标志寄存器中，AF 为 辅助进位 标志，CF 为 进位 标志，SF 为 符号 标志，ZF 为 零 标志，OF 为 溢出 标志，PF 为 奇偶性 标志。
9. 指出下列指令源操作数的寻址方式：
MOV AX, ARRAY[SI] 变址寻址
MOV AX, ES: [BX] 寄存器间接寻址
MOV AX, [200H] 直接寻址
MOV AX, [BX+DI] 基址变址寻址
MOV AX, BX 寄存器寻址
MOV AX, 1200H 立即数寻址
MOV AX, 20[BX+SI] 基址变址寻址
MOV AX, [DI+20] 变址寻址
10. 现有 (DS)=2000H, (BX)=0100H, (SI)=0002H, (20100H)=12H, (20101H)=34H, (20102H)=56H, (20103H)=78H, (21200H)=2AH, (21201H)=4CH, (21202H)=0B7H, (21203H)=65H, 下列指令执行后填入 AX 寄存器的内容：

(1) MOV AX, 1200H; (AX) = 1200H

(2) MOV AX, BX; (AX) = 0100H

(3) MOV AX, [1200H]; (AX) = 4C2AH

(4) MOV AX, [BX]; (AX) = 3412H

(5) MOV AX, 1100[BX]; (AX) = 4C2AH

(6) MOV AX, [BX][SI]; (AX) = 7856H

(7) MOV AX, 1100[BX][SI]; (AX) = 65B7H

11. 已知 'A' 的 ASCII 是 41H, 那么 'E' 的 ASCII 是 45H。
12. Ends 是段定义结束伪指令, 与 Segment 配套使用。
13. 逻辑地址由段基值和 偏移量 组成。将逻辑地址转换为物理地址的公式是 段基值*16+偏移量。其中的段基值是由 段寄存器 存储。
14. 一个有 16 个字的数据区, 它的起始地址为 70A0: 0100H, 那么该数据区的最后一个字单元的物理地址为 70B1E H。
15. DEBUG 命令中, 显示内存命令 D, 显示寄存器命令 R, 汇编命令 A, 执行命令 G。
16. 根据操作数的个数, 汇编指令可以分为 无操作数指令, 单操作数指令 和 双操作数指令。
17. 令 (AL)=98H, (BL)=12H, 则执行指令 MUL BL 后, (AX)= 0AB0H, OF= 1, CF= 1。

二、单项选择

1. 下列四个寄存器中, 不能作为间接寻址的寄存器是 (**C**)
- A、BX
B、BP
C、DX
D、DI
2. 存储器每段最长 (**B**)
- A、64B
B、64KB
C、16MB
D、1MB
3. 下列指令中, 正确的是 (**A**)
- A、ADD AX, [BX+SI]**
B、MOV AH, BX
C、MOV CS, AX
D、NOT CX, AX
4. 表示一条指令的存储单元的符号地址称 (**A**)
- A、标号**

- B、变量
 - C、类型
 - D、偏移量
5. 下列不是变量的类型的是 (**B**)
- A、字节型
 - B、字符型**
 - C、字型
 - D、双字型
6. 分别写出下列寄存器的位数 AX, BH, CX, CS (**D**)
- A、8 位、16 位、32 位、16 位
 - B、16 位、8 位、32 位、16 位
 - C、8 位、16 位、16 位、32 位
 - D、16 位、8 位、16 位、16 位**
7. INC 指令不影响 (**B**) 标志。
- A、OF
 - B、CF**
 - C、SF
 - D、ZF
8. 假定 (SS)=2000H, (SP)=0100H, (AX)=2107H, 执行指令 PUSH AX 后, 存放数据 21H 的物理地址是 (**D**)
- A、20102H
 - B、20101H
 - C、200FEH
 - D、200FFH**
9. 设 (AL)=86, (BL)=68, 执行 SUB AL, BL 指令后, 正确的结果是 (**D**)。
- A、SF=1
 - B、ZF=1
 - C、CF=1
 - D、OF=1**
10. 将累加器 AX 的内容清零的正确操作是 (**C**)。
- A、ADD AX, 0
 - B、OR AX, AX
 - C、XOR AX, AX**
 - D、CMP AX, AX
11. 一个完整的计算机系统通常应包括 (**C**)。
- A、系统软件和应用软件
 - B、计算机及其外部设备

- C、硬件系统和软件系统
- D、系统硬件和系统软件

12. 计算机的软件系统一般分为（ A ）两大部分。

- A、系统软件和应用软件
- B、操作系统和计算机语言
- C、程序和数据
- D、DOS 和 Windows

13. 用高级语言编写的程序（ C ）。

- A、只能在基本种计算机上运行
- B、无需经过编译或解释，即可被计算机直接执行
- C、具有通用性和可移植性
- D、几乎不占用内存空间

14. 下面几个不同进制的数中，最小的数是（ C ）。

- A、1001001B
- B、75
- C、37D
- D、A7H

15. 十进制正数 38 的八位二进制补码是（ D ）。

- A、00011001
- B、10100110
- C、10011001
- D、00100110

16. MOV AX, ES: [BX][SI]的源操作数的物理地址是（ B ）。

- A、 $16d \times (DS、+(BX、+(SI、$
- B、 $16d \times (ES、+(BX、+(SI、$
- C、 $16d \times (SS、+(BX、+(SI、$
- D、 $16d \times (CS、+(BX、+(SI、$

17. 下列数据中，可能是八进制数的是（ B ）。

- A、488
- B、317
- C、597
- D、189

18. 计算机语言有许多种，其中与硬件直接相关的是（ C ）

- A、网络语言
- B、操作系统
- C、机器语言
- D、高级语言

19. 执行指令 MOV AL, [BP] 时, 所读取的数据应在 (C)
- A、CS 段中
 - B、DS 段中
 - C、SS 段中**
 - D、ES 段中
20. 和外存相比, 内存的特点是 (A)。
- A、容量小、速度快、成本高**
 - B、容量小、速度快、成本低
 - C、容量大、速度快、成本高
 - D、容量大、速度快、成本低
21. 运算器的主要功能是 (C)。
- A、算术运算
 - B、逻辑运算
 - C、算术运算与逻辑运算**
 - D、函数运算
22. 汇编程序的循环控制指令中, 隐含使用 (C) 寄存器作为循环次数计数器。
- A、AX
 - B、BX
 - C、CX**
 - D、DX
23. 下面关于寄存器不正确的表述是: (C)
- A、AX 寄存器是算术运算的主要寄存器, 又称为累加器
 - B、BP、SP 通常作为地址指针寄存器
 - C、IP 指令指针寄存器可以通过 MOV 命令改变其值**
 - D、标志位寄存器中的状态标志位会受算术单元的运算结果而改变。
24. 用来存放下一条将要执行的指令地址的寄存器是 (B)
- A、SP
 - B、IP**
 - C、BP
 - D、CS
25. 完成将 BX 清零, 并使标志位 CF 清零, 下面错误的指令是 (B)
- A、SUB BX, BX
 - B、MOV BX, 00H**
 - C、XOR BX, BX
 - D、AND BX, 00H

三、判断

- 1、当运算的结果有奇数个 1 时，PF=0。(☒)
- 2、MOV AL, BX 是正确指令。(☒)
- 3、SP 的内容可以不指向堆栈的栈顶。(☒)
- 4、DW 伪指令用来定义双字。(☒)
- 5、MOV 指令要影响标志位。(☒)
- 6、同一物理地址，可以有不同的逻辑地址。(☒)
- 7、1KB=1000B。(☒)
- 8、存储器中的各段不允许重叠。(☒)
- 9、立即数不能作为目的操作数。(☒)
- 10、所有的条件转移指令都不影响标志位。(☒)
- 11、汇编语言是一种高级程序语言。(☒)
- 12、1 个 cpu 的寻址能力为 8kb, 那么它的地址总线宽度为 12。(☒)
- 13、伪指令也有相对应的机器指令。(☒)
- 14、SS:SP 有时候可以不指向栈顶元素。(☒)
- 15、jmp 的短转移是一种段间转移。(☒)
- 16、loop 指令能够实现段间转移。(☒)
- 17、add 指令只影响 ZF 标志位的取值。(☒)
- 18、push 指令可以将一个字节压入栈中。(☒)

四、程序分析

- 1、写出指令运行结果

MOV AL, 80H

MOV BL, 08H

MUL BL

则 (AX) = 0400H。

该指令完成 AL 寄存器和 BL 寄存器的内容相乘，结果存入 AX 寄存器中。

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 (<=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

【汇编语言程序设计】期末考试-试卷及参考答案

课程编号: H61030005 课程名称: 汇编语言程序设计 考试形式: 闭卷

学院: 信息工程 专业: 计算机科学与技术 考试日期: _____

题号	一	二	三	四	五	六	七	八	九	十	总分	累分人
题分	20	12	18	20	30						100	签名
得分												

考生注意事项: 1、本试卷共 7 页, 请查看试卷中是否有缺页或破损。如有立即举手报告以便更换。
2、考试结束后, 考生不得将试卷、答题纸和草稿纸带出考场。

一、单项选择题(每小题 2 分, 共 20 分)

1. 一个有 128 个字的数据区, 它的起始地址为 12ABH: 00ABH, 请给出这个数据区第一个字单元的物理地址是 (A)。
A. 12B5BH ; B. 12B6BH ;
C. 12C59H; D. 12BFEH。
2. 在下列指令的中, 不能实现把 AX 寄存器内容清零的指令是 (C)。
A. SUB AX, AX;
B. XOR AX, AX;
C. OR AX, 0000H;
D. AND AX, 0000H。
3. 条件转移指令 JNE 的测试条件为 (A)。
A. ZF=0 ; B. CF=0 ;
C. ZF=1 ; D. CF=1 。
4. 8086CPU 在基址加变址的寻址方式中, 变址寄存器可以为 (D)。
A. BX 或 CX ; B. CX 或 SI ;
C. DX 或 SI ; D. SI 或 DI。
5. 设 (AX) =1000H,
NEG AX
NOT AX

执行上述两条指令后, 正确的结果是 (B)。

- A. (AX) =1001H; B. (AX) =0FFFH;
- C. (AX) =1000H; D. (AX) =0111H。

6. 串指令中的目的操作数地址一定是由 A 提供。()

- A. ES: [DI] ; B. SS: [BP];
- C. DS: [SI] ; D. CS: [IP]。

7. 将 DX 的内容除以 2, 正确的指令是 (C)。

- A. DIV 2 ;
- B. DIV DX, 2 ;
- C. SAR DX, 1;
- D. SHL DX, 1。

8. 用户为了解决自己的问题, 用汇编语言所编写的程序, 称为 (B)。

- A. 目标程序; B. 汇编语言源程序;
- C. 可执行程序; D. 汇编程序。

9. 用一条指令仅实现将 $AX \leftarrow BX + SI$ 的方法是 (D)。

- A. XCHG AX, [BX][SI] ;
- B. MOV AX, [BX+SI];
- C. LEA AX, BX[SI] ;
- D. LEA AX, [BX][SI]。

10. 设 SP 初值为 2000H, 执行指令 “PUSH AX” 后, SP 的值是 (C)。

- A. 1FFFH; B. 1998H;
- C. 1FFEh; D. 2002H。

1. A 2. C 3. A 4. D 5. B 6. A 7. C 8. B 9. D 10. C

二、指出下列指令的出错原因, 并改正。(每小题 2 分, 共 12 分)

1. MOV BX, DL; 操作数类型不匹配, 改正为: MOV BL, DL 或 MOV BX, DX

2. MOV CS, AX; 代码段寄存器 CS 不能作目的操作数, 改正为 MOV DS, AX

3. ADD AX, DS; 段寄存器 DS 的内容不能作为加法指令的操作数, 改正为
MOV BX, DS
ADD AX, BX

4. TEST BX, [CX]; 不能使用 CX 实现寄存器间接寻址, 改正为
MOV SI, CX
TEST BX, [SI]

5. SUB [BX], [BP+SI]; 两个操作数不能同为存储器操作数, 且两个操作数的数据类型 不确定,
改正为: MOV AX, [BX]
SUB AX, [BP+SI]
或: MOV AL, [BX]
SUB AL, [BP+SI]

6. SHL DX ; 没有给出移位次数, 改正为 SHL DX, 1 或 SHL DX, CL

三、程序填空题 (注意: 每空只能填一条指令, 并注释说明所填指令的作用! 每空 3 分, 共 18 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

MOV CX, 10

MOV SUM, AX ; 将 AX 的原值保存在 SUM 单元中

MOV AL, 00H

NEXT: (1) ADD AL, [BX] ; 实现将表 TABLE 中的数据按字节累加到 AL 中

ADC AH, 0

JNC DONE

(2) MOV AX, SUM ; 总和值超过 AX 寄存器的容量, 恢复 AX 的原值

JC EXIT

DONE: INC BX

(3) LOOP NEXT ; 将计数器 CX 减 1, 并判断是否继续下一次循环

EXIT: HLT

2. BUFFER 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 ; 循环 50 次

LEA SI, BUFFER

MOV DI, SI

(1) ADD DI, 99 ; DI 指向最后一个字节数据的存储单元的偏移地址

NEXT: MOV AL, [SI]

(2) XCHG AL, [DI]; 交换 SI 和 DI 所指向单元的内容

MOV [SI], AL

(3) INC SI ; 使指针 SI 加 1, 指向下一个字节的数据

DEC DI

LOOP NEXT

HLT

四、程序分析题 (每小题 4 分, 共 20 分)

1. 设 (AX) = X, X 是无符号数且 $0 \leq X < 6000$, 分析下面程序段, 回答问题;

SHL AX, 1 ; 实现 X 乘 2

MOV BX, AX

SHL AX, 1

SHL AX, 1

ADD AX, BX

- (1) 该程序段的功能是什么?

实现无符号数 X 乘 10, 结果存放在 AX 寄存器中。——2 分

- (2) 该程序段执行完后, (AX) = 10X, (BX) = 2X。——2 分

2. 现有一子程序段:

```

SUB1    PROC
        TEST AL, 80H
        JE PLUS
        TEST BL, 80H
        JNE EXIT0
        JMP XCHANGE
PLUS:   TEST BL, 80H
        JE EXIT0
XCHANGE: XCHG AL, BL
EXIT0:  RET
SUB1    ENDP

```

试回答：(1) 子程序的功能是什么？

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

3. 现有如下程序段：

```

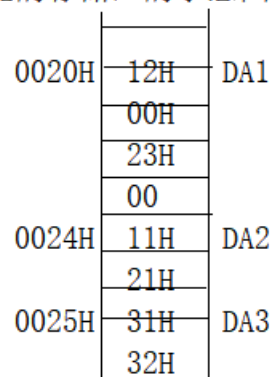
DATA    SEGMENT
        ORG 0020H
        DA1 DW 12H, 23H
        DA2 DB 11H, 21H
        DA3 DB 31H, 32H
        DA4 EQU $-DA3
        DA5 EQU 1234H
DATA    ENDS

```

请回答：

(1) 该程序段占有的存储单元字节数是多少？画出所分配的存储区的示意图。

答：8 个字节，



(2) 符号名 DA4 的值是多少？符号名 DA5 的值是多少？

答：DA4=2, DA5=1234H

4. 阅读如下程序：

```

ADRR    DB  XXH, XXH, ..
NUM      EQU  $-ADRR
RET_VLAUE DW ?
...
        MOV CX, NUM
        MOV BX, -1
NEXT:    INC BX
        CMP ADRR[BX], 0
        LOOPNZ NEXT
        JZ DONE
        MOV BX, OFFH
DONE:    MOV RET_VLAUE, BX
        HLT

```

请问：(1) 该程序完成的功能是什么？

功能：在以 ADRR 为起始地址的数组中，查找第一个是 0 的数据。

(2) 程序执行后指出 RET_VLAUE 单元中的内容是什么？

若找到，则将第一个是 0 的数据在数组中的偏移量送入 RET_VLAUE 单元中；若无 0 数据，则将 OFFH 送入 RET_VLAUE 单元中。

5. 阅读如下程序段，回答所提出的问题，假定 X_1, X_2, \dots, X_N 为无符号数。

```

ORG 100H
BLOCK DW X1, X2, ..., XN
COUNT EQU ($-BLOCK)/2
RESULT DW COUNT, ?
        LEA BX, BLOCK-2
        MOV CX, RESULT
        XOR AX, AX
GO_ON:  ADD BX, 2
        CMP AX, [BX]
        JAE NEXT
        MOV AX, [BX]
NEXT:   LOOP GO_ON
        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 = MAX (X₁, X₂, ..., X_N) ；

BX = 100H + 2 × (COUNT-1) ；

程序循环次数 = COUNT 。

五、编写程序（30 分）

得分	评阅人

1. 编写一程序，在 letter 为首地址的 26 个单元中依此存放字母 ‘A’ 、 ‘B’ 、 ‘Z’ 。（本小题 8 分）

```

data    SEGMENT
letter  DB 26 DUP(?)
count   EQU $-letter

data    ENDS

code    SEGMENT
        ASSUME cs:code, da:data
start:  mov ax, data
        mov ds, ax
        mov cx, count
        lea si, letter
        mov al, 'A'
next:   mov [si], al
        inc al
        inc si
        loop next
        mov ah, 4ch
        int 21h
code    ends
        end start

```

2. 以 Buffer 为首地址的存储区，存放有长度为 N 有符号数组，编写一程序求出其中负数的平均值（所有负数的和 > -32768），并存放在 M_average 单元中。（本小题 10 分）

```

data    SEGMENT

```



```

Buffer  DW XXXX, XXXX, ....., XXXX; N 个字
count   EQU  ($-Buffer) /2
M_average dw  ?
data    ENDS
code    SEGMENT
        ASSUME  cs:code, da:data
start:  mov ax, data
        mov ds, ax
        mov cx, count
        lea si, Buffer
        xor ax, ax ; 存负数的和
        mov bx, ax ; 存负数的个数
        next: test word ptr [si], 8000H
              je go_on
              add ax, [si]
              inc bx
        go_on: add si, 2
              loop next

```

cwd
 idiv bx
 mov M_average, ax
 ; 存负数的平均值
 mov ah, 4ch
 int 21h
 data ends
 end start

3. 从键盘接收一串不多于 99 个字符的字符串，对其进行分类统计，显示其中数字字符个数、大写英文字母个数、小写英文字母个数、以及其他字符的个数。要求使用子程序结构；数字字符的 ASCII 码为 30H……39H、大写英文字母的 ASCII 码为 41H……5AH、小写英文字母的 ASCII 码为 61H……7AH。
 (本小题 12 分)

```

data    SEGMENT
MSG      DB 0DH, 0AH, '输入一串字符: $'
MSG1     DB 0DH, 0AH, '大写字母个数: $'
MSG2     DB 0DH, 0AH, '小写字母个数: $'
MSG3     DB 0DH, 0AH, '数字符号个数: $'
MSG4     DB 0DH, 0AH, '其它符号个数: $'
BUFFER   DB 99
          DB 0
          DB 99 DUP (?)
BIG      DB 0 ; 大写字母个数
LITTLE   DB 0 ; 小写字母个数
DIG      DB 0 ; 数字符号个数
OTHER    DB 0 ; 其它符号个数
data     ENDS
code     SEGMENT
        ASSUME  cs:code, da:data
start:  mov ax, data
        mov ds, ax
        lea dx, MSG
        mov ah, 9 ; 9 号功能调用显示提示信息
        int 21h
        lea dx, BUFFER
        mov ah, 10 ; 10 号功能调用从键盘接收一串字符
        int 21h
        mov cl, BUFFER+1; 实际从键盘接收字符个数
        mov ch, 0
        lea si, BUFFER+2
again:  cmp byte ptr [si], 30H

```

```

        jb  others
        cmp byte ptr [si], 39H
        jbe digitals
        cmp byte ptr [si], 41H
        jb  others
        cmp byte ptr [si], 5AH
        jbe bigs
        cmp byte ptr [si], 61H
        jb  others
        cmp byte ptr [si], 7AH
        jbe littles
others: inc  OTHER
        jmp ok
digitals: inc DIG
        jmp ok
littles: inc LITTLE
        jmp ok
bigs:    inc BIG
ok:      inc si
        loop again
        lea dx, MSG1
        mov bh, BIG
        call dispdec
        lea dx, MSG2
        mov bh, LITTLE
        call dispdec
        lea dx, MSG3
        mov bh, DIG
        call dispdec
        lea dx, MSG4
        mov bh, OTHER
        call dispdec
        mov ah, 4ch
        int 21h
; DX 待显示数据说明, BH 待显示数据<100
dispdec PROC
        mov ah, 9
        int 21h
        mov al, bh
        xor ah, ah
        aam
        mov dl, ah
        mov dh, al
        add dl, 30H
        mov ah, 2
        int 21H
        mov dl, dh
        add dl, 30H
        mov ah, 2
        int 21H
        ret

```

```
dispec ENDP
code    ENDS
end start
```

