计算机科学与技术学院 2016-2017 学年第 2 学期考试试卷

汇编语言程序设计 A 卷 闭卷

姓名_____班级_____学号_____考试日期 2017 – 05 – 20

题号	一	二	三	四	五	六	七	总分	核对人
题分	10	10	10	20	10	20	20	100	
得分									

得分	评卷人

一、填空题（共 10 分，每空 1 分）

1. 如果一个 DW 定义的变量在内存中的物理地址是 10000H，那么这个变量所占的 2 个字节存贮单元的物理地址分别是 10000H 和 _10001H__ 。

2. 实方式下，（DS）= 1000 H，（SS）=（SI）= 2000H，则指令 MOV AX, [SI+5] 中源操作数的物理地址是 __12005H_____。

3. CPU 执行转移指令“JMP FAR PTR L1”的主要操作是（L1 是一条指令语句的标号）：_____标号 L1 所在段地址送 CS，L1 的偏移地址送 IP_____。

4. 设在数据段 DATA 中定义了变量：BUF DW 1234H。 且代码段中有：

```
ASSUME   DS:DATA
MOV      AX,  DATA
MOV      DS,  AX
LEA      BX,  BUF
MOV      SI,  0
```

若在执行上述指令之后，需要将字变量 BUF 中的内容送入 AX 中，可使用如下几种方式：
用直接寻址方式访问源操作数，指令语句为： MOV AX, ___BUF_____
用基址加变址寻址方式访问源操作数，指令语句为： MOV AX, ___[BX+SI]____

5. 假设在执行下列程序段之前（EAX）= 1234567 H，（SP）= 1000 H：

```
PUSH     EAX
PUSH     AX
POP      EAX
```

则执行完该程序段后：（EAX）= ___45674567H_____，（SP）= ___0FFEH_____ 。

6. 已知有下列程序段：

```
MOV      AX,  90F0H
ADD      AL,  AH
```

则执行完后：（AX）= _9080H__，CF = __1___，OF = __0___。

得分	评卷人

二、判断选择题（共 10 分，每题 1 分）

1. 指令 ADD　BX , AL 的错误原因是 ___B____。

(A) BX, AL 的位置写反了　　(B) 源、目的操作数类型不匹配

(C) 源、目的操作数不能同时为存储器操作数　　(D) 源、目的操作数类型均不明确

2. 指令 ADD　[BX] , 20H 的错误原因是 ___D____。

(A) BX 不能用于寄存器间接寻址方式　　(B) 源操作数不能用立即寻址方式

(C) 源、目的操作数不能同时为存储器操作数　　(D) 源、目的操作数类型均不明确

3. 指令 MOV　WORD PTR [BX], BX+1 的错误原因是 ___B____。

(A) 源、目的操作数的顺序写反了　　(B) 源操作数的寻址方式无效

(C) 源、目的操作数不能同时为存储器操作数　　(D) 目的操作数的寻址方式无效

4. 对于指令 MOV　AX, [SI+DI]，下列说法正确的是 ___B____。

(A) 源操作数的类型不确定　　(B) 源操作数的寻址方式无效

(C) 目的操作数的寻址方式无效　　(D) 该指令没有错误

5. 对于指令 OUT　DX, 60，下列说法正确的是 ___C____。

(A) 源、目的操作数的顺序写反了　　(B) 目的操作数错误

(C) 源操作数错误　　(D) 该指令没有错误

6. 对于指令 IN　AX, 256，下列说法正确的是 ___B____。

(A) 源、目的操作数的顺序写反了　　(B) 源操作数错误

(C) 目的操作数错误　　(D) 该指令没有错误

7. 设（BX）= 1000 H，（DS:[BX]） = 1234 H，（CS:[BX]）= 5678 H，则转移指令 JMP WORD PTR [BX] 转移到的目标指令的偏移地址是 ___A____。

(A) 1234 H　　(B) 1000 H

(C) 5678 H　　(D) 0

8. 设 BUFW 为字变量，CON 为符号常量，下面四个语句中错误的语句是 ___B____。

(A) MOV　BUFW , AX　　(B) MOV　CON ,　BUFW

(C) LEA　AX ,　BUFW　　(D) MOV　BUFW , CON

9. 在模块化程序设计中，若模块 1 需要访问模块 2 中的字变量 SUM，则在模块 1 和模块 2 中分

别需要按下面的语句进行声明 ___B____。

(A) 模块 1：PUBLIC SUM； 模块 2：EXTRN SUM:WORD

(B) 模块 1：EXTRN SUM:WORD； 模块 2：PUBLIC SUM

(C) 模块 1：PUBLIC SUM； 模块 2：EXTRN SUM

(D) 模块 1：EXTRN SUM； 模块 2：PUBLIC SUM

10. 在汇编源程序中，关于语句“END START”（START 是一条指令语句的标号）作用的完整描述是 ___D____。

(A) 告诉汇编程序不再处理该语句后面的内容

(B) 指明 CPU 执行该程序时的第一条语句是标号 START 所指的语句

(C) 表示程序结束

(D) 同时指明(A)和(B)的内容

得分	评卷人

三、数据段定义如下，请回答下面的问题。（共 10 分）

```
DATA   SEGMENT  USE16
A1     DB     1, -1
B1     DW     12, 89AB H
C1     DW     E1
D1     EQU    $ - A1
E1     DB     2 DUP ('2', 2)
DATA   ENDS
```

(1) 以字节为单位，画出该数据段的数据存储示意图，并在存储图中标明各变量名称及其对应的偏移地址值。（7 分）

	低地址
A1:00	01H
	0FFH
B1:02	0CH
	00H
	ABH
	89H
C1:06	08H
	00H
E1:08	32H
	02H
	32H
	02H
	高地址

(2) 执行下列各指令后,各寄存器的内容。（3 分）

(a) MOV AX, D1 ; (AX) = ___8___

(b) MOV AX, B1+1 ; (AX) = _0AB00H_

(c) MOV EAX, DWORD PTR C1
; (EAX) = _02320008H__

得分	评卷人

四、简答题（共 20 分）

1. 简述基于窗口的应用程序中的窗口消息处理程序的作用。（5 分）

答：对接受到的消息进行判断，分类处理；对于未处理的消息，交给操作系统做缺省处理。

2. 设计宏指令“COMP　wA, bA”，其中 wA 和 bA 是在 DATA 数据段中分别由 DW 和 DB 定义的变量（无符号数）（DS 与 DATA 相关联且（DS）= DATA）。该宏指令的功能是比较 wA 和 bA 的大小，若 wA>=bA 则设置（AX）= 1，否则设置（AX）= 0。（5 分）

答：
```
COMP    MACRO wA, bA
        LOCAL   L1, L2
        MOVZX   BX, bA
        CMP     wA, BX
        JAE     L1
        MOV AX, 0
        JMP     L2
L1:     MOV AX, 1
L2:
        ENDM
```

3. 下面的程序实现了对 **INT 16H** 的子功能 00H 和 10H 的接管，其实现的功能是：在安装程序运行结束返回 DOS 后，所有的按键都变成了‘A’。请完成程序中新的中断服务程序 NEW16H 的相关代码（5 分）

说明：DOS 中所有读取按键的操作，最后都是通过调用 **INT 16H** 的子功能 00H 和 10H 来实现的，所以只需接管 16H 号中断的 00H 和 10H 子功能，就可实现按键的重定义。

功能号：00H 和 10H

功能：从键盘读入字符

入口参数：AH = 00H — 读键盘

= 10H — 读扩展键盘

出口参数：AH = 键盘的扫描码

AL = 字符的 ASCII 码

提示：’A’ 的扫描码为 1EH。

```
.386
CODE        SEGMENT  USE16
            ASSUME  CS:CODE, SS:STACK
OLD_INT     DW       0,  0
NEW16H      PROC  FAR                ;;请为 NEW16H 编写代码

      CMP AH,00H
      JE MY_16
      CMP AH,10H
      JE MY_16
      JMP DWORD PTR CS:OLD_INT
MY_16:
      PUSHF
      CALL DWORD PTR CS:OLD_INT
      MOV AH, 1EH
      MOV AL, 'A'
      IRET

NEW16H      ENDP
BEGIN:      MOV      AX, 0     ；以下为安装程序
             ……          ；将原中断矢量放到 OLD_INT 中，NEW16H 设置成新的中断矢量
            INT      21H       ；驻留内存, 并返回 DOS
CODE        ENDS
STACK       SEGMENT  USE16  STACK
            DB   100   DUP (0)
STACK       ENDS
            END      BEGIN
```

4. 设 X 变量是用 DW 定义的 16 位无符号数，Y 变量用 DB 定义的 8 位无符号数，编写一个程序段计算 (X * Y + 60H) / 7，分别用 DD 和 DB 定义变量 V 和 R，将商和余数分别保存在 V 和 R 中 （需要考虑每一步可能溢出的情况）。（5 分）

答：

```
MOVZX   EAX, X
MOVZX   EBX, Y
MUL   EBX              ;X*Y=>EDX:EAX
ADD     EAX, 60H       ;不可能进位到 EDX
MOV EBX, 7
DIV   EBX              ;EDX:EAX/7
MOV R, DL
MOV V, EAX
```

得分	评卷人

五、程序填空题（共 10 分，每空 1 分）

下面程序的功能是：从键盘输入一个字符，将该字符的 ASCII 码按 16 进制显示出来。
例如，若输入字符是‘C’，则显示：THE ASCII OF C IS 43H

```
.386
STACK    SEGMENT USE16 _STACK_
         DB      100  DUP(0)
STACK    ENDS
DATA     SEGMENT USE16
MSG      DB      'THE ASCII OF A'
MSG_1    DB      ' IS '
MSG_2    DB      '41H', 0AH, 0DH, '$'
DATA     ENDS
CODE     SEGMENT USE16
         ASSUME CS:CODE, DS:DATA, SS:STACK
BEGIN:   MOV     AX, DATA
         __MOV   DS, AX___
         MOV     AH, 1
         INT     21H
         MOV     MSG_1－1, ___AL____
         MOV     BX, 1
L1:      AND     AL, 0FH
         CMP     AL, _10__
         JB      L2
         SUB     AL, _10__
         ADD     AL, 'A'
         JMP     L3
L2:      __ADD   AL, '0'_____
L3:      MOV     MSG_2 [BX], AL
         __CMP   BX, 0__
         JZ      L4
         MOV     AL, MSG_1－1
         SHR     AL, 4
         DEC     BX
         JMP     L1
L4:      MOV     AH, 9
         LEA     DX, _MSG_
```

```
        ___INT   21H__
        MOV     AX,  4C00H
        INT     21H
        JMP     L1
CODE    ENDS
        END     _BEGIN_
```

得分	评卷人

六、程序分析（共 20 分）

1. 阅读程序，回答问题。（共 10 分）

```
.386
STACK   SEGMENT  USE16  STACK
DB      100      DUP(0)
STACK   ENDS
DATA    SEGMENT  USE16
BUF     DB       98, 90, 76, 84, 100, 91, 75, 55
N       EQU      $ - BUF
RESULT  DB       0
DATA    ENDS
CODE    SEGMENT  USE16
        ASSUME  CS:CODE,  DS:DATA,  SS:STACK
BEGIN:  MOV      AX,  DATA
        MOV      DS,  AX
        MOV      AX, 0
        MOV      BX, 0
L1:     CMP      BX, N
        JZ       L2
        ADD      AL, BUF[BX]
        ADC      AH, 0
        INC      BX
        JMP      L1
L2:     MOV      BL, N
        DIV      BL
        MOV      RESULT, AL      ; ---------------------------------- (1)
        ;;
        ADD      AH, AH          ; ---------------------------------- (2.1)
        CMP      AH, N           ; ---------------------------------- (2.2)
        JB       L3              ; ---------------------------------- (2.3)
        INC      AL              ; ---------------------------------- (2.4)
```

```
                MOV     RESULT, AL        ; ------------------------------- (2.5)
                ;;
L3:             MOV     AX, 4C00H
                INT     21H
CODE            ENDS
                END     BEGIN
```

(1) 程序执行到语句**(1)**处所实现的功能是什么（RESULT 中的数代表什么意义）？（8 分）
答：累加 BUF 中 N 个无符号数，然后除以 N 计算平均值(向下取整)，结果放入 RESULT
。

(2) 语句**(2.1~2.5)**的作用是什么？（2 分）
答：根据余数，对上面计算的平均值进行四舍五入，最后结果仍放入 RESULT。

2. 阅读程序，回答问题。（共 10 分）

```
.386
STACK       SEGMENT  USE16  STACK
DB          100      DUP(0)
STACK       ENDS
DATA        SEGMENT  USE16
BUF         DB       'Assembly Language', 0
RESULT      DB       $-BUF  DUP(0)
DATA        ENDS
CODE        SEGMENT  USE16
            ASSUME  CS:CODE,  DS:DATA,  SS:STACK
BEGIN:      MOV      AX,  DATA
            MOV      DS,  AX
            LEA      SI,   BUF
            LEA      DI,   RESULT
            ;;
            MOV      AH,  1
            INT      21H              ;从键盘输入一个字符并将其 ASCII 码保存到（AL）中
            MOV      AH,  AL
            CMP      AL,  'A'
            JB       L1
            CMP      AL,  'Z'
            JA       L1
            SUB      AL,  'A'
            ADD      AL,  'a'
```

```
            JMP     L2
L1:         CMP     AL, ‘a’
            JB      L2
            CMP     AL, ‘z’
            JA      L2
            SUB     AL, ‘a’
            ADD     AL, ‘A’
            ;;
L2:         MOV     BL, [SI]
            CMP     BL, 0
            JZ      L4
            CMP     BL, AL
            JZ      L3
            CMP     BL, AH
            JZ      L3
            MOV     [DI], BL      ;----------------------------------------(1)
            INC     DI            ;---------------------------------------- (2)
L3:         INC     SI
            JMP     L2
            ;;
L4:         MOV     AX, 4C00H
            INT     21H
CODE        ENDS
            END     BEGIN
```

假设从键盘上输入的是字母’A’，请回答下面的 3 个问题：

1) 程序执行到 L4 时，缓冲区 RESULT 的内容是什么？（4 分，卓越工程师班 3 分 W）
   答：前面的字节是’ssembly Lnguge’，最后三个字节为 0。

2) 若漏写了语句**(1)**，程序执行到 L4 时 RESULT 的内容是什么？（3 分）
   答：内容不变，仍是全 0。

3) 若漏写了语句**(2)**，程序执行到 L4 时 RESULT 的内容是什么？（3 分）
   答：第一个字节为’e’，后面的字节全 0。

4) 已知字母’A’的 ASCII 码是 41H，’z’的 ASCII 码是 7AH，请只修改一行代码，优化该程序。(1 分)(本题仅卓越工程师班需要做)

   答：CMP AL, ‘A’

   JB L1 => 这一行可改为 JB L2

   因为 ASCII 小于’A’的字符也不可能是小写字母，不需要执行 L1 下面的小写字母处理。

得分	评卷人

七、程序设计（20 分）

编写一个完整的实方式下程序，实现如下功能：先在屏幕显示：“Please input a string: ”，然后从键盘输入一个字符串（可以是任意字符），对输入字符串中的字符按照从小到大次序排序（根据字符的 ASCII 码的大小），排序结果保存在原来的缓冲区中，最后在屏幕上输出排序后的结果。（ACM 班要求小写字母按照大写字母的值来判断；相同字母的大小写同时出现时，按照先大写后小写的次序排序，例如：3AaaDeGGgrr）

要求：(1) 画出程序流程图；

(2) 程序完整（包括堆栈段、数据段、代码段定义等），至少给出 2 条必要的注释；

(3) 用子程序 SORT 实现排序，采用堆栈传递参数，主程序调用 SORT 的方式如下：

```
PUSH  字符串缓冲区的偏移地址
PUSH  字符串缓冲区中字符的个数
CALL  NEAR PTR SORT
ADD   SP, 4
```

```
DATA SEGMENT USE16
MSG DB 'Please input a string: $'
BUF   DB   50
      DB   ?
      DB 50 DUP(0)
CRLF DB 0DH, 0AH, '$'
DATA ENDS

STACK SEGMENT USE16 STACK
       DB 200 DUP(0)
STACK ENDS

CODE SEGMENT USE16
      ASSUME CS:CODE,DS:DATA,SS:STACK
BEGIN:
      MOV AX,DATA
      MOV DS,AX

      LEA DX, MSG
```

```
        MOV AH, 9
        INT 21H

        LEA DX, BUF
        MOV AH, 10
        INT 21H

        LEA DX, CRLF
        MOV AH, 9
        INT 21H

        MOV BL, BUF+1
        MOV BH, 0
        MOV BYTE    PTR    BUF+2[BX], '$'

        PUSH OFFSET BUF+2          ;压入字符串起始地址
        PUSH BX                                ;压入字符串长度
        CALL NEAR PTR SORT
        ADD    SP, 4

        LEA DX, BUF+2
        MOV AH, 9
        INT 21H

        MOV AH,4CH
        INT 21H

SORT    PROC
      PUSH    BP
      MOV    BP, SP
      MOV    BX, [BP+6]
      MOV    CX, [BP+4]
      MOV    DX, BX
      ADD    DX, CX
      DEC    DX                    ;计算字符串结束的偏移地址
      MOV    SI, BX
L1:     CMP    SI, DX
```

```
        JAE     EXT
        MOV DI, SI
        INC DI
L2:   CMP   DI, DX
        JA   L3
        MOV AL, [DI]
        CMP [SI], AL
        JBE   OK
        XCHG [SI], AL       ;前面字符较大则交换两个字符
        XCHG [DI], AL
OK: INC DI
          JMP L2
L3: INC SI
         JMP   L1
EXT:     POP   BP
      RET
SORT   ENDP
CODE ENDS
        END BEGIN
```