

# 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 个字节存储单元的物理地址分别是 10000H 和 \_\_\_\_\_。
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 \_\_\_\_\_。
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)：\_\_\_\_\_。

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, \_\_\_\_\_

用基址加变址寻址方式访问源操作数，指令语句为：MOV AX, \_\_\_\_\_

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) = \_\_\_\_\_，(SP) = \_\_\_\_\_。

6. 已知有下列程序段：

```

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

则执行完后：(AX) = \_\_\_\_\_，CF = \_\_\_\_\_，OF = \_\_\_\_\_。

得分	评卷人

## 二、判断选择题（共 10 分，每题 1 分）

- 指令 `ADD BX, AL` 的错误原因是 \_\_\_\_。
  - `BX, AL` 的位置写反了
  - 源、目的操作数类型不匹配
  - 源、目的操作数不能同时为存储器操作数
  - 源、目的操作数类型均不明确
- 指令 `ADD [BX], 20H` 的错误原因是 \_\_\_\_。
  - `BX` 不能用于寄存器间接寻址方式
  - 源操作数不能用立即寻址方式
  - 源、目的操作数不能同时为存储器操作数
  - 源、目的操作数类型均不明确
- 指令 `MOV WORD PTR [BX], BX+1` 的错误原因是 \_\_\_\_。
  - 源、目的操作数的顺序写反了
  - 源操作数的寻址方式无效
  - 源、目的操作数不能同时为存储器操作数
  - 目的操作数的寻址方式无效
- 对于指令 `MOV AX, [SI+DI]`，下列说法正确的是 \_\_\_\_。
  - 源操作数的类型不确定
  - 源操作数的寻址方式无效
  - 目的操作数的寻址方式无效
  - 该指令没有错误
- 对于指令 `OUT DX, 60`，下列说法正确的是 \_\_\_\_。
  - 源、目的操作数的顺序写反了
  - 目的操作数错误
  - 源操作数错误
  - 该指令没有错误
- 对于指令 `IN AX, 256`，下列说法正确的是 \_\_\_\_。
  - 源、目的操作数的顺序写反了
  - 源操作数错误
  - 目的操作数错误
  - 该指令没有错误
- 设  $(BX) = 1000H$ ,  $(DS:[BX]) = 1234H$ ,  $(CS:[BX]) = 5678H$ ，则转移指令 `JMP WORD PTR [BX]` 转移到的目标指令的偏移地址是 \_\_\_\_。
  - 1234H
  - 1000H
  - 5678H
  - 0
- 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 \_\_\_\_。
  - `MOV BUFW, AX`
  - `MOV CON, BUFW`
  - `LEA AX, BUFW`
  - `MOV BUFW, CON`
- 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 \_\_\_\_\_。

- (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 \_\_\_\_\_。

- (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
```

低地址


高地址

- (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）
- (2) 执行下列各指令后,各寄存器的内容。（3 分）
  - (a) MOV AX, D1                    ; (AX) = \_\_\_\_\_
  - (b) MOV AX, B1+1                    ; (AX) = \_\_\_\_\_
  - (c) MOV EAX, DWORD PTR C1  
   ; (EAX) = \_\_\_\_\_

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）
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）

3. 下面的程序实现了对 INT 16H 的子功能 00H 和 10H 的接管，其功能是：在安装程序运行结束后返回 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
```

```

NEW16H    ENDP
BEGIN:     MOV     AX,0    ;以下为安装程序
           ... ..       ;将原 16H 号中断服务程序的入口地址保存到 OLD_INT 中,
                           ;并将 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)

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 _____
        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     AH, 1
        INT     21H
        MOV     MSG_1-1, _____
        MOV     BX, 1
L1:      AND     AL, 0FH
        CMP     AL, _____
        JB      L2
        SUB     AL, _____
        ADD     AL, 'A'
        JMP     L3

L2:      _____
L3:      MOV     MSG_2[BX], AL

        _____
        JZ      L4
        MOV     AL, MSG_1-1
        SHR     AL, 4
        DEC     BX
        JMP     L1
L4:      MOV     AH, 9
        LEA     DX, _____

        _____
        MOV     AX, 4C00H
        INT     21H
        JMP     L1

CODE     ENDS
        END     _____

```

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 RESULT ; ----- (2.4)

;;

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）

(2) 语句**(2.1~2.4)**的作用是什么？（2 分）

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 4 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）
  
- (2) 若漏写了语句(1)，程序执行到 L4 时 RESULT 的内容是什么？（3 分）
  
- (3) 若漏写了语句(2)，程序执行到 L4 时 RESULT 的内容是什么？（3 分）
  
- (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）

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



