参考答案

2014-2015 学年第 2 学期《汇编语言程序设计》考试试卷

A 卷　　闭卷　　考试时间：20 15 年 6 月 2 4 日

专业_____　班级_____　学号_____　学生姓名_____

题号	一	二	三	四	五	六	七	八	总分	核对人
题分	10	10	10	15	10	10	15	20	100	
得分										

得分	评卷人

一、填空题（共 10 分，每空 1 分）

1. 设（DS）=2000H ，（SS）=1000H，（ESI）=3001H，（EBP）=2015H，（AX）=100H。实方式下，指令语句“ CMP AX， [EBP+ESI] ”中，源操作数的物理地址是 15016H ，目的操作数的寻址方式是 寄存器寻址 。

2. 设(AX)=9234H，执行指令语句 SHR AX, 4 之后，(AX)= 0923H ，CF= 0 。

3. 设有如下程序段：

```
BUF     DW   -9, 10, 5, -7, 2
.....
MOV     EBX, 1
MOV     CX,  BUF[EBX*2]
```

执行上述语句后，(EBX)= 1 ，（CX）= 10 。

4. Win32窗口应用程序主要由主程序、窗口主程序、窗口消息处理程序和用户处理程序四个部分组成，其中窗口消息处理程序接收由操作系统转发来的消息，分析消息类型，根据不同的消息作不同的处理。

5. 中断是计算机系统内部件 CPU 所具有的能打断当前执行的程序，转而为临时出现的事件服务，事后又能自动按要求恢复执行原来程序的一种功能，其中处理事件的程序称为中断服务程序。

得分	评卷人

二、选择题（共 10 分，每题 1 分）

1. 下列指令语句中错误的是： D 。

A. PUSH AX　　B. PUSH 1234H　　C. PUSH EAX　　D. PUSH AL

2. 指令语句 MOV AX, 2[BX+DX] 错误的原因是__B__。

A. 源操作数类型不明确　　B. DX不能作为变址寄存器

C. BX不能作为基址寄存器　　D. 源、目的操作数不能同时为存储器操作数

3. 指令语句 ADD EBX, AX 错误的原因是__C__。

A. EBX, AX 的位置写反了　　B. 源、目的操作数不能同时为存储器操作数

C. 源、目的操作数类型不匹配　　D. 源、目的操作数类型均不明确

4. 设BUF为一字类型变量，指令语句 MOV BUF, 3[SI] 错误的原因是__C__。

A. 目的操作数类型明确，但是源操作数类型不明确　　B. BUF不能作为目的操作数

C. 源、目的操作数不能同时为存储器操作数　　D. SI不能用于变址寻址方式

5. 正确读取71H端口中数据的指令语句是：__A__。

A. IN AL, 71H　　B. IN AL, [71H]　　C. MOV AL, DS:[71H]　　D. OUT 71H, AL

6.指令语句 CMP AX, BX 比较两个有符号数的大小，若希望 (AX)>(BX) 时转移到标号NEXT处执行，则紧跟在该指令语句之后的正确的转移指令语句是：__D__。

A. JNS NEXT　　B. JGE NEXT　　C. JA NEXT　　D. JG NEXT

7.设（AX）=3015H，(BX)=5015H，则执行指令语句 ADD AX，BX后，有__D__。

A. OF=0，SF=0　　B. OF=0，SF=1　　C. OF=1，SF=0　　D. OF=1，SF=1

8. 设BUF为双字类型变量，对于指令语句MOV EAX, BUF[BP+SI]，源操作数一定是在__D__中。

A. 代码段　　B. 数据段　　C. 堆栈段　　D. 变量BUF所在的段

9. 已知一完整的函数定义语句为：RADIX PROC NEAR STDCALL NUM:DWORD, FLAG:WORD，其中STDCALL指明了函数定义中的参数按从右到左的顺序入栈，则指令语句 INVOKE RADIX 1234H, 10H 经过汇编之后生成的指令语句是：__C__。

A.
```
CALL RADIX
PUSH 10
PUSH 1234H
```

B.
```
PUSH 1234H
PUSH 10H
CALL RADIX
```

C.
```
PUSH 10H
PUSH 1234H
CALL RADIX
```

D.
```
PUSH 1234H
CALL RADIX
```

10. 将 8 位有符号数(AL)扩展成 16 位有符号数并存放到 AX 中，正确的指令语句是__A__

A. CBW　　B. CWD　　C. MOVZX AX, AL　　D. MOV AX, AL

得分	评卷人

三、改错题（每改一个错误 1 分，共 10 分）

下列程序的功能是：用户输入一个字符，将输入字符的 ASCII 码按照十六进制的形式显示在屏幕上，当用户输入回车时，程序退出，否则继续循环等待输入。显示格式如下所示：

THE ASCII CODE OF A IS 41H
THE ASCII CODE OF 7 IS 37H
THE ASCII CODE OF （当用户输入回车时，程序退出）

请将程序中的语法错误和逻辑错误圈出来并在其右侧写出正确的形式。

```
.386
DATA SEGMENT USE16
   BUF    D B 'THE ASCII CODE OF'
   BUF1  D  B ' IS $'
   BUF2  D  B ' ', 0, 0, 0A, 0DH, '$'
   TAB    DB '0123456789ABCDEF'
DATA ENDS
STACK SEGMENT USE16 STACK
   100 DUP(0)
STACK ENDS
CODE SEGMENT
ASSUME CS:CODE, DS:DATA, SS:STACK
BEGIN: MOV   DS,   DATA
       MOV   DX,   OFFSET  BUF
       MOV   AH,   9
       INT   21H
NEXT:  M OV   AH,   1
       INT   21H
       CMP   AL,   0DH
       JE    EXIT
       PUSH  AX
       LEA   DX,   BUF1
       MOV   AH,   9
       INT   21H
       POP   CL
       MOV   AL,   CL
       AND   AL,   0FH
       MOV   B X,  AL
       MOV   AL,   TAB[BX]
       MOV   B UF2+2,  AL
       MOV   AL,   CL
       AND   AL,   F0H
       SHR   AL,   4
```

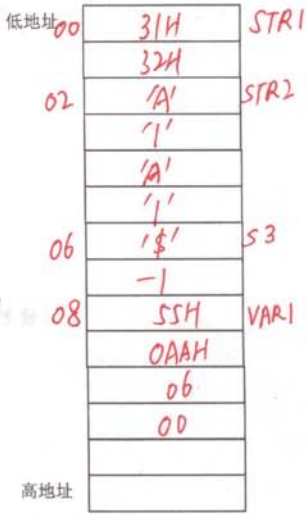
```
        MOVZX  B X, AL
        MOV    AL, TAB[BX]
        MOV    B UF2+1, AL
        MOV    DX, OFFSET BUF2
        MOV    AH, 9
        INT    21H
        JMP    NEXT
EXIT: MOV    AX, 4CH        MOV AH,4CH
        INT    21H
CODE ENDS
END   B EGIN
```

得分	评卷人

四、简答题（共 15 分）

1. 数据段定义如下，回答下列问题。(10 分)

```
DATA   SEGMENT   USE16
STR1   DB   '12'
STR2   DB   2 DUP('A', '1')
S3     DB   '$',  -1
COUNT = $ - STR1
VAR1   DW   0AA55H, S3
DATA   ENDS
```



(1) 以字节为单位，画出它们在存储单元中的存放形式。并给出 STR1、STR2、S3 和 VAR1 的偏移地址和 COUNT 的值（7 分）

11
8 STR1, STR2, S3, VAR1 的偏移地址分别是 0000, 0002H, 0006H, 0008H

(2) 执行下列语句后屏幕显示的结果是什么？（3 分）

```
MOV   AX,   DATA
MOV   DS,   AX
LEA   BX,   STR2
ADD    BX,  2
MOV   BYTE PTR [BX], '$'
LEA   DX,   STR1
MOV   AH,  9
INT   21H
```

12A1

2. 写出在实方式下，用 TD 下查看中断号是 10H 的中断向量表的步骤。(5 分)

① 双击 TD.EXE，启动 TD；② 用鼠标单击数据窗口区并将鼠标移
③ 按鼠标右键，在弹出的菜单上，选择 Goto.
④ 在出现的对话框中，输入：0000:0040↵
⑤ 0000:0040 处的第1个字内容为10号中断处理程序的偏移地址和段地址.

得分	评卷人

五、指令填空（共 10 分，每空 1 分）

1.在以 BUF 为首址的存储区中，存放着以'$'为结束符的字符串，下列程序段完成的功能是将 BUF 中小写字母转换成大写字母，并且将 BUF 中的字符显示在屏幕上。

```
.386
DATA   SEGMENT   USE16
BUF    DB 'AD D AX, BX', 0AH, 0DH
       DB 'MOV cx, 10', 0AH, 0DH, '$'
DATA   ENDS
.....
CODE   SEGMENT      USE16
START: MOV      AX,  DATA
       MOV      DS,  AX
       LEA BX, BUF
NEXT:  M OV     DL, [BX]
       CMP DL, '$'
       JE       EXIT
       CMP      DL,  'a'
       JB   OUT1
       CMP      DL，'z'
       JA       OUT1
       SUB      DL，20H
OUT1:  MOV  AH, 2
       INT  21H
       INC   BX
       JMP      NEXT
EXIT：MOV       AH，4CH
       INT      21H
CODE   ENDS
       END START
```

2. 在以 BUF 为首址的存储区中，存放着若干有符号双字类型数据，下面的程序段完成将 BUF 中的数按照从大到小的顺序排序，采用冒泡排序方法。

```
.386
DATA S EGMENT   USE16
 BUF    DD   -5, 200H, 300H, 50H
 N=($-BUF)/4
DATA   ENDS
STACK S EGMENT   USE16  ST ACK
```

```
D B   200 DUP(0)
STACK ENDS
CODE   SEGMENT   USE16
 ASSUME  CS: CODE,  D S:DATA,   SS:STACK
START:  MOV   AX, DATA
        MOV  DS, AX
        MOV   CX,  N-1  ；外循环用 CX 计数
QU1:    MOV   DX,   CX；内循环用 DX 计数
        MOV  E SI, 0
QU2:    MOV  EAX, BUF[ESI*4]
        CMP  E AX,  B UF[ESI*4+4]；比较相邻的两个数的大小
        JGE  NOXCH
XCH:    XCHG   BUF[ESI*4+4], E  AX
        MOV  B UF[ESI*4], E  AX
NOXCH: INC   ESI
        DEC   DX
        JNE   QU2
        LOOP   QU1
EX1:    MOV   AH, 4CH
        INT   21H
CODE   ENDS
END   S TART
```

得分	评卷人

六、编写程序段（10 分）

1. 设学生结构和数据段定义如下：（5 分）

```
STUDENT   STRUCT
        XUEHAO        DW   0
        STU_NAME      DB   'ZHANG'
        YUWEN         DB   0；语文成绩
        SHUXUE        DB   0；数学成绩
        ENGLISH       DB    0；英语成绩
        ZONGFEN       DW   0；总分
  STUDENT   ENDS
.DATA
   S1   STUDENT   <50, , 74,    95 , 8 0, >
```

编写程序段计算变量 S1 中三门课程的总分，并保存在 ZONGFEN 字段中。

```
MOV  AL, S1.YUWEN
ADD  AL, S1.SHUXUE
MOVZX AX, AL
MOV  CL, S1.ENGLISH
MOV  CH, 0
ADD  AX, CX
MOV  S1.ZONGFEN, AX
```

2.定义一个宏 EXPA，其调用形式是 EXPA A1, B1, C1(其中 A1，B1，C1 均为有符号字类型变量)，该宏实现计算(A1*5+B1)/8 的值，并保存在 C1 中，不考虑溢出。(5 分)

```
EXPA MACRO A1, B1, C1
MOV AX, A1
IMUL AX, 5
ADD AX, B1
CWD
MOV BX, 8
IDIV BX
MOV C1, AX
ENDM
```

得分	评卷人

七、阅读程序并分析（15 分）

阅读程序，解答下列问题。

```
.386
DATA  S EGMENT  USE16
    BUF1  DW  100H, 20 0H,  300H,  400H,  500 H
    M=($-BUF1)/2
    BUF2  DW  100H,  22 2H,  300H,  444 H,  500 H, 600  H
    N=($-BUF2)/2
    BUF3  DW  M DUP(?)
DATA  ENDS
STACK  S  EGMENT  USE16  ST ACK
DB  200 DUP(0)
STACK ENDS
CODE  SEGMENT  USE16
ASSUME  CS: CODE,  D S:DATA,  SS:STACK
START:  MOV  AX,  DATA
        MOV  DS,  A X
        LEA  SI,  BUF1
        LEA  B X,  BUF3
        MOV  DI，0
        MOV  CX, M    ----------①
NEXT1:  MOV  AX,  [SI]
        CALL  FIND_ONE
        ADD  SI, 2
        LOOP  NEXT1
        MOV   AH,  4CH
        INT  21 H
FIND_ONE  PROC
        PUSH  DI
        PUSH  CX
        LEA   DI,  B UF2
```

```
        MOV    CX,  N
LOPA1:  CMP    AX, [ DI]
        JNE    LOPA2
        MOV    [BX],  A X
        ADD    BX, 2
LOPA2:  ADD    DI,  2
        LOOP   LOPA1
        POP    CX
        POP    DI----------②
        RET
FIND_ONE  E NDP
CODE   ENDS
END   S TART
```

1. 该程序的功能是什么？（3 分）

将缓冲区BUF1中的数，若在BUF2中出现，则将其存放到BUF3缓冲区中。重复1次，就要存放1次。依此存放。

2. 程序运行完毕，BUF3 存放的结果是什么？（3 分）

100H 300H 500H

3. 子程序 FIND_ONE 的功能和入口参数各是什么？（3 分）

判断找出BUF2中，出现和值为(AX)的数，并将其存放到(BX)所指向的内存单元。(BX)最后指向下一次要引以存放数的位置。

4. 若①所在的指令语句误写成 MOV CX, 0, 则 主程序循环体执行多少次？（3 分）

10000H次

5. 若漏写②处的指令语句，则子程序 FIND_ONE 会返回到什么地方？（3 分）

返回到 START 处执行

得分	评卷人

八、编程题（20 分）

设以 BUF 为首址的字存储区中，存放了若干有符号数，试编写一完整的程序，查找其中的最大和最小值，并以 16 进制形式显示这两个值。

(1)用子程序实现一个数的显示，并且用堆栈传递参数；

(2)写出寄存器的使用分配情况，画出程序流程图；（2分）

(3)程序完整（包括数据段定义、堆栈段定义、代码段定义等），给出 4 条以上注释；（2分）

(4)BUF 的内容自己设定。

```
.386
DATA SEGMENT USE16
TAB DB '0123456789ABCDEF'
BUF DW 123,-456,789,25
N = ($-BUF)/2
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
      MOV AX,BUF ;(AX)用于存放当前找到的最小的数
      MOV DX,BUF ;(DX)用于存放当前找到的最大的数
      MOV CX,N-1 ;循环次数存在(CX)中
      MOV EBX,1
L1:   CMP AX,BUF[EBX*2]
      JLE L2
      MOV AX,BUF[EBX*2]
L2:   CMP DX,BUF[EBX*2]
      JGE L3
      MOV DX,BUF[EBX*2]
L3:   INC EBX
      LOOP L1
```

数据段定义正确 2分
堆栈段 1分
代码段结构 2分
给DS赋值； 1分
返回 DOS 1分
找到最大最小数 3分（循环、转移、函数）
子程序结构、传参 2分
正确显示两个数 3分（循环，转换，显示）
堆栈使用 1分

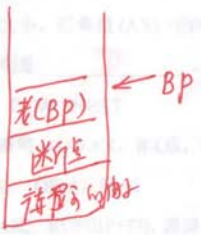
简化一下：PUSH

可优化一下：先取出两个数，放在DX中，后面用(AX)与(DX)进行比较。

循环，找到最小的数在(AX)中，最大的数在(DX)中。

和最

```
PUSH  AX
CALL  DISPLAY      ; 显示最小的数
PUSH  DX
CALL  DISPLAY      ; 显示最大的数
MOV   AH, 4CH
INT   21H
; 显示一个数 (以16进制) 的子程序 DISPLAY
; 栈中压入一个要显示的数
DISPLAY  PROC
        PUSH  BP
        MOV   BP, SP
        PUSH  AX
        PUSH  DX
        PUSH  CX
        PUSH  SI
        PUSH  BX
        MOV   CX, 4
        MOV   BX, [BP+4]   ; 取出要转换的数在BX中
DLP1:   ROL   BX, 4
        MOV   SI, BX
        AND   SI, 0FH
        MOV   DL, TAB[SI]
        MOV   AH, 2
        INT   21H
        LOOP  DLP1
        POP   BX
        POP   SI
        POP   CX
        POP   DX
        POP   AX
        POP   BP
        RET   2
DISPLAY  ENDP
CODE  ENDS
      END  BEGIN
```



等价一下:
PUSHA ⇐ PUSH AX, PUSH DX, PUSH CX, PUSH SI, PUSH BX

POPA ⇐ POP BX, POP SI, POP CX, POP DX, POP AX, POP BP