

计算机科学与技术学院 2017-2018 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 2018-05-26

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）

- 在实方式下，已知(DS)=1000H, (SS)=2000H, (ES)=3000H, (BP)=100H, (SI)=200H 则指令 MOV BX, ES:[BP+SI], 源操作数的物理地址是 30300H。
- 变量 VA 的定义为：VA DW 5678H, 1234H, 则指令
MOV EAX, DWORD PTR VA
执行后，AH= 56H。
- 基于窗口的 Win32 应用程序一般由 4 个部分组成，其中负责对接收到的消息进行判断、分类处理的部分称为 窗口消息处理程序。
- 已知有下列 16 位程序段：
MOV EAX, 12345H
XOR CX, CX
L: DEC EAX
LOOP L
则执行完后：(EAX) = 2345H。
- 写出一条双操作数指令语句，其中一个操作数必须是 CS 寄存器 MOV AX, CS。
- 已知有下列程序段：
MOV AL, 80H
MOV BL, AL
ADD AL, BL
则执行完后：(AL) = 0，CF = 1，OF = 1，ZF = 1。

得分	评卷人

二、判断说明题。

判断下面每个小题的语法或叙述是否有错，若有错，则在下划线上面写出错误原因；若没有错，则在下划线上面注明“对”。（共 10 分，每空 1 分）

- MOV [EAX], 10H
操作数类型不明确。
- MOVSX EAX, BL
对。
- OUT 200H, AL
端口号大于 255，应使用 DX 存放端口号。
- MOV [BX], BYTE PTR [SI]
源操作数与目的操作数同时为内存单元。
- PUSH AL
字节类型操作数不能入栈。
- 已知(AX)=1111H，(BX)=2222H，(CX)=3333H，执行下列语句后
PUSH BX
PUSH CX
POP EAX
AX、BX、CX 的值分别是 3333H、2222H、3333H
对。
- 下述指令执行后，乘积应从寄存器 EAX 取得。
MUL BX
乘积应该从 DX、AX 中取得。
- 实方式下，10H 号中断处理程序入口地址存放在 0H:64H 开始的四个字节中。
存放在 0H:40H 开始的四个字节中。
- 将一个堆栈段的源操作数送到 AX，可以用指令 MOV AX, [BP+BX]
源操作数为基址加变址寻址，不能同时使用两个基址寄存器。
- 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 A1，则在模块 1 和模块 2 中分别需要按下面的语句进行声明。
模块 1: EXTRN A1 模块 2: PUBLIC A1:WORD
EXTRN A1:WORD，PUBLIC 后不要 :WORD。

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
BUF1 DB 11H, 22H, 33H
      DB $-BUF1
BUF2 DW -1H, 1234H
BUF3 DW BUF2
      DB 2 DUP(2, 3)
CON EQU 2
DATA ENDS
```

低地址端

11H
22H
33H
03H
0FFH
0FFH
34H
12H
04H
00H
02H
03H
02H
03H

- (1) 以字节为单位，画出数据段的数据存储示意图。（6 分）
每错一格扣 0.5 分。

- (2) 变量 BUF2 和 BUF3 的偏移地址值分别是

4 和 8。（1 分）
每空 0.5 分。

高地址端

- (3) 执行下列各指令后,各寄存器的内容。（3 分）

```
LEA SI, BUF1+1
MOV BX, CON
MOV CL, [BX+SI]
MOV CH, BYTE PTR BUF3
```

(SI)=1、(BX)=2、(CX)=0403H
每空 1 分。

得分	评卷人

四、简答题（共 20 分，每小题 5 分）

1. 实方式下，假设(ES)=0，简述下列两条语句能基本模拟 “INT n” 指令功能的原理。

PUSHF

CALL DWORD PTR ES:[4*n]

答：“INT n” 指令执行的主要操作是：①FLAGS 进栈；②其后下一条指令的段和偏移地址进栈，③转移到 n 号中断的处理程序，即从 0H:[4*n]开始的单元中取出段和偏移送到 CS、IP 中。而上述两条指令中的第一条指令模拟上述①的操作；第二条指令模拟了②的操作，并当(ES)=0 时，模拟了③的操作，因此，这两条指令与 “INT n” 的主要操作一样。

INT n 指令主要操作每个 1 分，两条指令模拟的操作各 1 分。

2. 设计宏指令 “WADD V1,V2”，其中 V1 和 V2 是同一个数据段中的字类型变量。该宏指令的功能是计算 V1 与 V2 之和，结果送 AX 寄存器。不考虑段寄存器初始化和溢出的问题。

答：WADD MACRO V1,V2 //宏名、宏定义伪指令、参数各 1 分
 MOV AX,V1 //宏体功能 1 分
 ADD AX,V2
 ENDM //结束标志 1 分
//以上任何一处局部不足扣 0.5 分。

3. 已知在一个数据段中定义了字类型变量 X，字节类型变量 Y 和 R，双字类型变量 V，均为有符号数，编写一个程序段计算 $(X + Y * 100) / 7$ ，将商和余数分别保存在 V 和 R 中（不考虑段寄存器初始化问题，但要考虑每一步可能溢出的情况）。

答：MOVSBX EAX,Y
 IMUL EAX,100
 MOVSBX EBX,X
 ADD EAX,EBX
 CDQ
 MOV EBX,7
 IDIV EBX
 MOV V,EAX
 MOV R,DL
乘、加、除、传输保存、考虑溢出各 1 分；任何一处局部不足扣 0.5 分。

4. 请用两种不同方法（写出指令语句）实现 SI 中的高 8 位内容与低 8 位内容的互换。

答：方法 1：ROL SI,8

方法 2：MOV AX,SI
 XCHG AH,AL
 MOV SI,AX

每种方法 2.5 分；若只是关键指令及格式基本正确 1.5 分；两种方法的指令序列类似（比如 ROL SI,8 与 ROR SI,8）则只能算是实现了同一种方法。

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：计算 $M \times N$ 矩阵 BUFA 每行元素之和 $S_i = a_{i1} + a_{i2} + a_{i3} + \dots + a_{in}$ ($i=1, 2, \dots, m$)，结果存入字存储区 BUFS 中（假定不考虑溢出的问题）。

```
.386
DATA1 SEGMENT USE16
BUFA  DB  11H,12H,13H,14H,15H      ; a11,a12,a13,a14,a15
        DB  21H,22H,23H,24H,25H      ; a21,a22,a23,a24,a25
        DB  31H,32H,33H,34H,35H      ; a31,a32,a33,a34,a35
        DB  41H,42H,43H,44H,45H      ; a41,a42,a43,a44,a45
M      = 4
N      = 5
BUFS   DW   M  DUP(0)
DATA1 ENDS
STACK2 SEGMENT USE16 STACK
        DB 200 DUP(0)

STACK2 ENDS
CODE SEGMENT USE16 'CODE'
        ASSUME CS:CODE,  SS: STACK2, DS: DATA1 
BEGIN: MOV  AX, DATA1
        MOV  DS, AX
         LEA  DI, BUFS 
        MOV  CX, M
        MOV  BX,  0 
LOPI:   MOV  DX, 0
        MOV  SI, 1
LOPJ:   MOV  AL, BUFA-1[BX][SI]
         CBW 
        ADD  DX, AX
        INC  SI
        CMP  SI, N
         JBE  LOPJ 
        MOV  [DI], DX
         ADD  DI, 2 
        ADD  BX, SI
        DEC  BX
         LOOP LOPJ 
EXIT:   MOV  AH, 4CH
        INT  21H
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
DATA SEGMENT USE16
BUF      DW  1, 2, 3, 5, 30, 20
NUM      EQU  ($-BUF)/2
BUF1     DW  NUM DUP(0)
BUF2     DW  NUM DUP(0)
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  BX,BUF
        LEA  SI,BUF1
        LEA  DI,BUF2
        MOV  CX, NUM  ;----- ①
L1:     MOV  BP, [BX]  ;----- ②
        MOV  AX, BP
        MOV  DL, 2    ;----- ③
        DIV  DL
        CMP  AH, 0    ;----- ④
        JZ   EV
        MOV  [SI], BP
        ADD  SI, 2
        JMP  NXT
EV:     MOV  [DI], BP
        ADD  DI, 2
NXT:
        ADD  BX, 2    ;----- ⑤
        LOOP L1
        MOV  AH,4CH   ;----- ⑥
        INT  21H
CODE ENDS
        END  BEGIN

```

(1) 上述程序的功能是什么？（5 分）

答：将数组 BUF 中的奇数送到数组 BUF1 中，偶数送到数组 BUF2 中。

全对给 5 分；奇数偶数写反给 3 分；未看出区分奇偶数（如写成大于、小于的）酌情给 1~2 分。

(2) 若将语句②处的标号 L1 上移一行，误写到语句①处，则程序是否还会执行语句⑥？为什么？（2 分）

答：不会，程序会发生死循环。

全对给 2 分；回答“会”，给 0 分，只写“不会”，未写原因，给 1 分。

(3) 若漏写了语句⑤，程序执行到语句⑥时，BUF1 和 BUF2 中的内容是什么？（3 分，卓越工程师班 1 分）

答：BUF1 中的内容为：1, 1, 1, 1, 1, 1。

BUF2 中的内容为：0, 0, 0, 0, 0, 0。

全对，给 3 分（卓越工程师班 1 分）；数值对，但数量不对（BUF1: 1,1,1 和 BUF2: 0, 0,0），给 2 分（卓越工程师班 0.5 分）；数量对，但数值不对（BUF1: 1,0,0,0,0,0 和 BUF2: 0, 0,0,0,0,0）给 2 分（卓越工程师班 0.5 分）。其他，酌情给 0~1 分（卓越工程师班 0~0.5 分）。

(4) 从语句③到语句④，共 3 条指令，可以优化为 1 条指令，请写出这条指令（2 分，本题仅卓越工程师班需要做）

答：TEST AX,1 （或 AND AX,1）

合理答案均给 2 分。不合理答案给 0 分。

2. 阅读程序，回答问题。（共 10 分）

```
.386
DATA SEGMENT USE16
STR1 DB 'Hello World !', 0
STR2 DB 'Hello', 0
STR3 DB 'HELLO', 0
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 SI, STR1
    LEA DI, STR2
    CALL FUN1
    LEA SI, STR3 ;----- ③
    LEA DI, STR1
    CALL FUN1
```

```

LEA SI, STR1          ;----- ④
LEA DI, STR1
CALL FUN1
MOV AH, 4CH           ;----- ⑤
INT 21H

FUN1 PROC
LOOP1: MOV AL, [SI]
      MOV BL, [DI]
      INC SI
      INC DI
      CMP AL, 0
      JZ RET1
      CMP AL, BL
      JZ LOOP1
RET1:  SUB AL, BL
      JA RET2          ;----- ①
      JZ RET3
      MOV AL, -1
      JMP RET3
RET2:  MOV AL, 1        ;----- ②
RET3:  RET
FUN1 ENDP
CODE ENDS
END BEGIN

```

(1) 上述程序中，子程序 FUN1 的功能、入口参数、出口参数各是什么？出口参数不同取值的含义是什么？（5 分）

答：FUN1 的功能是比较两个字符串的大小。入口参数是 SI、DI，出口参数是 AL。

SI 和 DI 分别存放两个字符串的首地址，字符串以字节 0 结束。

若两字符串相等，(AL)=0；若 SI 所指字符串小于 DI 所指字符串，(AL)=-1；若 SI 所指字符串大于 DI 所指字符串，(AL)=1。

子程序功能 1 分，入口参数 2 分，出口参数 1 分，出口参数含义 1 分。

(2) 执行到语句③、④、⑤时，AL 寄存器的值分别为多少？（3 分）

答：执行到语句③时，(AL)=1；

执行到语句④时，(AL)=-1；

执行到语句⑤时，(AL)=0；

每个值正确给 1 分。

(3) 某同学提出，从语句①到语句②，共 5 条指令太过繁琐，可以优化为 1 条指令：

SAR AL, 7

这种说法是否有道理？为什么？（2 分）

答：不能等价替换。

若替换，则 SI 所指字符串大于等于 DI 所指字符串时，(AL)=0，和原返回参数不同。

全对给 2 分；只写“不能”，未写原因，给 1 分。写了“不能”，原因不完全正确，给 1.5 分。

得分	评卷人

七、完整程序设计（20 分）

从键盘输入任意字符串（最大长度 50），统计子串“SUM”在该字符串中出现的次数（假设不超过 9 次），并在下一行显示该次数。

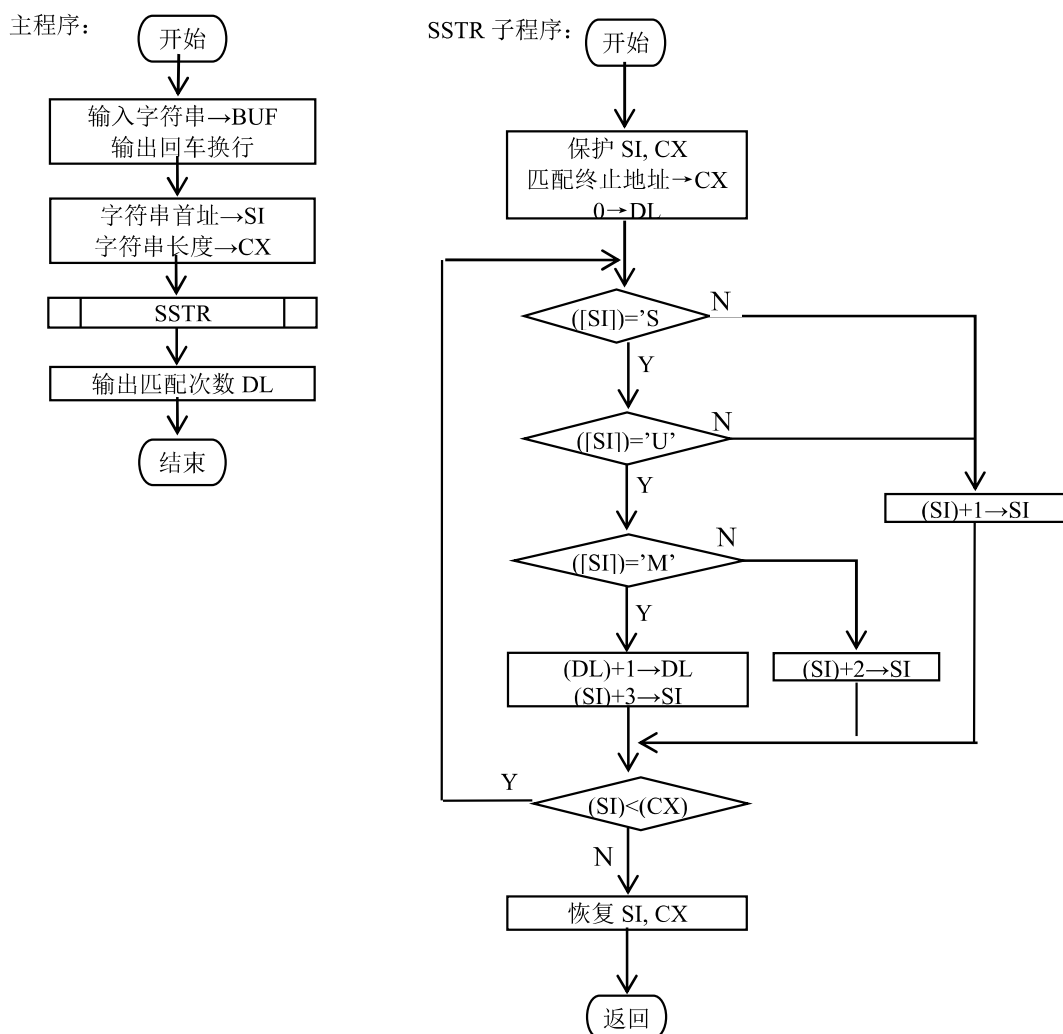
要求：（1）画出程序流程图；

（2）程序完整（包括堆栈段、数据段、代码段定义等），至少给出 4 条必要的注释；

（3）用子程序 SSTR 实现在字符串中搜索“SUM”的功能，入口参数、出口参数自定。

（对 ACM 班还要求：在搜索子串“SUM”时，对这三个字符必须同时进行比较）

流程图：



```
.386
STACK  SEGMENT USE16 STACK
        DB 200 DUP(0)
STACK  ENDS
DATA    SEGMENT USE16
        BUF DB 11
        DB 0
        DB 11 DUP(0)
CLRF    DB 0DH, 0AH, '$'
DATA    ENDS
CODE    SEGMENT USE16
        ASSUME  CS:CODE, SS:STACK, DS:DATA, ES:DATA
START:  MOV  AX, DATA
        MOV  DS, AX
        MOV  DX, OFFSET BUF ;输入字符串
        MOV  AH, 10
        INT  21H
        MOV  DX, OFFSET CLRF
        MOV  AH, 9
        INT  21H
        LEA  SI, BUF + 2
        MOVZX CX, BUF + 1 ;获得字符串长度
        CALL SSTR
        ADD DL, '0' ;输出匹配次数
        MOV AH, 2
        INT 21H
        MOV AH, 4CH
        INT 21H
;子程序 SSTR: 统计字符串中“SUM”子串出现的次数
;入口参数: SI: 字符串首址, CX: 字符串长度
;出口参数: DL: “SUM”子串出现次数
SSTR    PROC
        PUSH SI
        PUSH CX
        ADD CX, SI
        SUB CX, 2 ;CX 保存终止匹配位置
        XOR DL, DL
L0:     CMP [SI], BYTE PTR 'S' ;比较 S
        JNE L1
        CMP [SI+1], BYTE PTR 'U' ;比较 U
        JNE L1
        CMP [SI+2], BYTE PTR 'M' ;比较 M
        JNE L2
        INC DL ;DL 保存匹配次数
        ADD SI, 3 ;当前匹配上, 则从第四个字母重新比较
```

```

L3:    CMP SI, CX
        JB L0                ;若到达终止匹配位置, 则退出循环
        JMP EXIT
L1:    INC SI                ;当前第一个字母不匹配,或第二个字母不匹配, 则从第二个字母重新比较
        JMP L3
L2:    ADD SI, 2            ;当前第三个字母不匹配, 则从第三个字母重新比较
        JMP L3
EXIT:   POP CX
        POP SI
        RET
SSTR    ENDP
CODE    ENDS
        END    START

```

ACM 要求:

定义一个 4 字节字符串, SUBSUM DB 'SUM', 0;再用一个双字寄存器取待比较的输入串的 4 个字符, 把高 4 位屏蔽后进行比较。具体程序段如下:

```

L0:    MOV    ECX, DWORD PTR [SI]
        CMP    CL, 0DH
        JE     EXIT
        AND    ECX, 00FFFFFFH
        CMP    ECX, DWORD PTR SUBSUM
        JNE    L1
        INC    DL
        ADD    SI, 3
        JMP    L0
L1:    INC    SI
        JMP    L0

```

第七题评分标准:

- (1) 程序流程图 4 分 (主程序、子程序各 2 分)。
- (2) 整个程序框架正确 4 分, 含:
 - 所有段定义正确完整 2 分;
 - ASSUME 和 END 语句正确 1 分;
 - 子程序定义正确完整 1 分。
- (3) 正确实现主程序功能 (含必要的注释) 7 分。
- (4) 正确实现子程序功能 (含必要的注释) 5 分 (ACM 班所规定要求未达到则扣 2 分)。

注: 主程序和子程序中至少要含有 4 条必要的注释, 每条 0.5 分
