

【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：

```

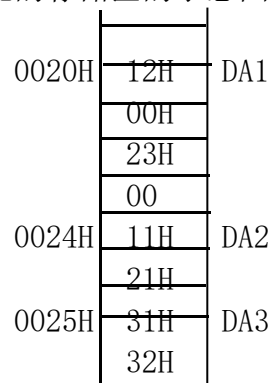
DATA    SEGMENT
        ORG 0020H
        DA1 DW 12H, 23H
        DA2 DB 11H, 21H
        DA3 DB 31H, 32H
        DA4 EQU $-DA3
        DA5 EQU 1234H
DATA    ENDS

```

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, 0FFH
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 0FFH 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
        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

```

```
dispdec ENDP
code    ENDS
        end start
```