

# 华南农业大学期末考试试卷 ( A 卷 )

2012 学年第2 学期

考试科目: 汇编语言程序设计

考试类型: 闭卷考试

考试时间: 120 分钟

学号\_\_\_\_\_ 姓名\_\_\_\_\_ 年级专业\_\_\_\_\_

## 考试注意事项:

- ① 本试卷分为试卷与答卷2部分。试卷共有9页, 三大题。
- ② 所有解答必须写在答卷上, 写在试卷上不得分。
- ③ 本试卷使用Intel 8086/8088 CPU指令系统和寻址方式可以完成答卷, 也可以使用Intel 80X86的指令系统和寻址方式来完成。
- ④ 试卷背面可以作为草稿纸用。

## 一、简答题

1. 若变量DATA在数据段中的偏移地址为24C0H, 24C0H至24C3H单元中分别存放着11H、22H、33H、44H。执行下列几条指令后, 寄存器BX (2211H)、CL (11H)、SI (24C0H)、DS (4433H) 的值分别是多少?

```
LEA    SI,DATA
MOV    CL,[SI]
LDS    BX,DATA
```

2. 假设端口地址为40H的寄存器的内容是88, 端口地址为80H的寄存器的内容是10。下列程序段执行结束后, 寄存器AX、BL、CL的值分别是多少?

START:

```
IN      AL,40H
MOV     BL,AL
IN      AL,80H
MOV     CL,AL
MOV     AX,0
```

CONTIN:

```
ADD     AL,BL
ADC     AH,0
DEC     CL
JNZ     CONTIN
```

AX:880 BL:88 CL:0

3. 下面是一个.COM结构的程序, 该程序的功能是从键盘输入1个字符串, 并在下一行将所输入字符串逆序输出, 请补充完成程序中未完成部分。

```
PROG  SEGMENT
      ORG     100H
      ASSUME  CS:PROG,DS:PROG,SS:PROG,ES:PROG
MAIN  PROC    NEAR
      MOV     AH,0AH
      LEA     DX,BUFF
      INT     21H
      LEA     DX,CRLF
      MOV     AH,09H
```

```

                INT      21H
                LEA      BX,BUFF
                INC      BX
                ;在此处补充所需要的程序段
                MOV CL, [BX]
                ADD BX, CX
L1:             MOV      DL,[BX]
                MOV      AH,2
                INT      21H
                DEC      BX
                LOOP     L1
                MOV      AH,4CH
                INT      21H
MAIN           ENDP
BUFF          DB        101,?,101 DUP(?)
CRLF          DB        0DH,0AH,24H
PROG          ENDS
                END      MAIN

```

4. 设变量A、B、C1、D、E都是16位有符号整数，下面程序将 $(A-(B \times C1 + D - 540)) \div E$ 的商保存保存在A，余数保存在B，请补充所需要的程序段。

```

CODE          SEGMENT
                ORG      100H
MAIN          PROC      NEAR
                MOV      AX,B
                IMUL     C1
                MOV      BX,DX
                MOV      CX,AX
                MOV      AX,D
                CWD
                ADD      CX,AX
                ADC      BX,DX
                SUB      CX,540
                SBB      BX,0
                ;在此处补充所需要的程序段
                MOV AX,A
                CWD
                SUB AX,CX
                SBB DX,BX
                IDIV E
                MOV A,AX
                MOV B,DX
                INT      20H
MAIN          ENDP
                A          DW      100
                B          DW      2

```

```

C1      DW    300
D       DW    -10
E       DW     3
CODE    ENDS
        END    MAIN

```

## 二、程序填空题

下列程序实现从一组按照升序排列的无符号数数组中查找VALUME的值，如果找到，则使CF=0，并在SI中给出该单元在数组中的偏移地址；如果未找到，则使CF=1。

```

DSEG    SEGMENT
VALUME  DW    55H
LOW_IDX DW    ?
HIGH_IDX DW   ?
DSEG    ENDS
ESEG    SEGMENT
LIST    DW    12,11H,22H,33H,44H,55H,66H,77H,88H,99H,111H,222H,333H
ESEG    ENDS
CSEG    SEGMENT
        ASSUME CS:CSEG,DS:DSEG, ES:ESEG
B_SEARCH PROC FAR
        PUSH    DS
        MOV AX, 0 ①
        PUSH    AX
        MOV     AX,DSEG
        MOV DS, AX ②
        MOV     AX,ESEG
        MOV     ES,AX
        LEA DI, LIST ③
        MOV     AX,VALUME
        CMP     AX,ES:[DI+2]
        JA      CHK_LAST
        CLC
        LEA     SI,ES:[DI+2]
        JE      EXIT
        STC ④
        JMP     EXIT
CHK_LAST: MOV     SI,ES:[DI]
        SHL     SI,1
        ADD     SI,DI
        CMP     AX,ES:[SI]
        JB      SEARCH
        CLC
        JE      EXIT
        STC
        JMP EXIT ⑤
SEARCH:  MOV     LOW_IDX,1
        MOV     BX,ES:[DI]

```

```

MOV     HIGH_IDX,BX
MOV     BX,DI
MID:    MOV     CX,LOW_IDX
        MOV     DX,HIGH_IDX
        CMP     CX,DX
        JA      NO_MATCH
        ADD     CX,DX
        SHR     CX,1
        MOV     SI,CX
        SHL     SI,1
COMPARE:
        CMP     AX,ES:[BX+SI]
        JE      EXIT
        JA HIGHER                                ⑥
        DEC     CX
        MOV     HIGH_IDX,CX
        JMP     MID
HIGHER:
        INC     CX
        MOV     LOW_IDX,CX
        JMP     MID
NO_MATCH:
        STC
EXIT:
        RET                                    ⑦
B_SEARCH ENDP
CSEG     ENDS
        END B SEARCH                            ⑧

```

### 三、阅读程序

1. 仔细阅读下面程序，说明该程序的功能。

```

DATA    SEGMENT
X        DB      0CH, 9, 8, 0FH, 0EH, 0AH, 2, 3, 7, 4
DATA    ENDS
STACK   SEGMENT  STACK
        DW      20H DUP(0)
TOP     LABEL    WORD
STACK   ENDS
CODE    SEGMENT
        ASSUME   CS:CODE,DS:DATA,SS:STACK
STARTUP  PROC    FAR
        MOV      AX,STACK
        MOV      SS,AX
        LEA      SP,TOP
        MOV      AX,DATA

```

```

        MOV     DS,AX
        XOR     AX,AX
        XOR     CL,CL
        XOR     BX,BX
LOP:    TEST     X[BX],01H
        JZ      NEXT
        ADD     AL,[BX]
        INC     AH
NEXT:   INC     BX
        INC     CL
        CMP     CL,10
        JNE     LOP
EXIT:   MOV     AH,4CH
        INT     21H
STARTUP      ENDP
CODE        ENDS
        END     STARTUP

```

统计 X 数组中奇数的和，保存到 AL 中，将累加个数保存到 AH 中

2.下面程序的功能是什么？

```

        DATAS      SEGMENT
                FZ      DB 'F:\ABC.TXT',0
                FBUFFER  DB 10 DUP(?)
                ESTRING  DB "ERROR!$"
        DATAS      ENDS
        STACKS     SEGMENT  STACK
                DB      100 DUP(?)
        TOP  LABEL  WORD
        STACKS     ENDS
        CODES      SEGMENT
                ASSUME  CS:CODES,DS:DATAS,SS:STACKS
MAIN  PROC  FAR
        PUSH     DS
        XOR      AX,AX
        PUSH     AX
        MOV      AX,DATAS
        MOV      DS,AX
        MOV      AL,2
        LEA      DX,FZ
        MOV      AH,3DH

```

```

        INT      21H
        JC       ERROR
        MOV      BX,AX
        MOV      CX,1
        LEA      DX,FBUFFER
L0:
        MOV      AH,3FH
        INT      21H
        JC       ERROR
        CMP      AX,0
        JZ       L2
        MOV      DI,DX
        CMP      BYTE PTR [DI],'a'
        JL       L1
        CMP      BYTE PTR [DI],'z'
        JG       L1
        SUB      BYTE PTR [DI],20H
        MOV      AX,-1
        CWD
        MOV      CX,DX
        MOV      DX,AX
        MOV      AL,01
        MOV      AH,42H
        INT      21H
        JC       ERROR
        MOV      CX,1
        LEA      DX,FBUFFER
        MOV      AH,40H
        INT      21H
        JC       ERROR
L1:
        JMP      L0
L2:
        MOV      AH,3EH
        INT      21H
        JNC      EXIT
ERROR:
        LEA      DX,ESTRING
        MOV      AH,9
        INT      21H
EXIT:
        RET
MAIN      ENDP
CODES     ENDS
        END      MAIN

```

**把 FZ 文件中的小写字母转变为大写字母**

3.下面是某程序的汇编列表文件，请将程序执行过程中堆栈最满时每个单元的内容填入下表。假设程序执行时 PSP（Program Segment Prefix）段地址为 0BDEH，STACKS 段的段地址为 1324H，CODES1 段的段地址为 0BFBH，CODES2 的段地址为 0BFEH。

地 址	机器码	汇编语言指令	
0000		DATAS	SEGMENT
0000	??	X	DB ?
0001		DATAS	ENDS
0000		STACKS	SEGMENT STACK
0000	0060[ ???? ]	DW	60H DUP(?)
00C0		TOP	LABEL WORD
00C0		STACKS	ENDS
0000		CODES1	SEGMENT
		ASSUME	CS:CODES1,DS:DATAS,SS:STACKS
0000		MAIN	PROC FAR
0000	B8 ---- R	MOV	AX,STACKS
0003	8E D0	MOV	SS,AX
0005	8D 26 00C0 R	LEA	SP, TOP
0009	06	PUSH	ES
000A	33 C0	XOR	AX,AX
000C	50	PUSH	AX
000D	B8 ---- R	MOV	AX,DATAS
0010	8E D8	MOV	DS,AX
0012	9A 0000 ---- R	CALL	FAR PTR P2
0017	C6 06 0000 R 01	MOV	X,1
001C	CB	RET	
001D		MAIN	ENDP
001D		P1	PROC FAR
001D	B8 0002	MOV	AX,2
0020	9A 0000 ---- R	CALL	FAR PTR P2
0025	CB	RET	
0026		P1	ENDP
0026		CODES1	ENDS
0000		CODES2	SEGMENT
		ASSUME	CS:CODES2
0000		P2	PROC FAR
0000	E8 0009 R	CALL	P3
0003	C6 06 0000 R 03	MOV	X,3
0008	CB	RETF	
0009		P2	ENDP
0009		P3	PROC NEAR
0009	C6 06 0000 R 04	MOV	X,4

```

000E  C3                                RET
000F                                P3    ENDP
000F                                CODES2 ENDS
                                END    MAIN

```

偏移地址	栈内容
(00B6H)	03H
(00B7H)	00H
(00B8H)	17H
(00B9H)	00H
(00BAH)	FBH
(00BBH)	0BH
(00BCH)	00H
(00BDH)	00H
(00BEH)	DEH
(00BFH)	0BH
(00C0H)	
SP 初值	00C0H

#### 四、程序设计题

1. 从键盘输入一系列字符（以回车符结束，字符的个数不超过 200 个），统计输入字符串中数字与非数字字符的个数，并将计数结果输出。

2. 有一个首地址为 LIST 的数组，其中首单元保存数组的长度。试编制程序删除数组中所有为零的项，并将后续项向前压缩，最后将数组的剩余部分补上零。

3. 一组数（不超过 200 个数）已经存入数组 Y 中，编程实现对这些数按照升序进行排列。

4. 试编写一个程序，实现判断所输入的 19 位银联卡卡号是否有误，如果正确则显示“Correct! ”，否则显示“Wrong! ”。

我们日常使用的各种银联卡卡号一般是由 19 位数字组成，前几位是发行者标识代码和个人账号标识，最后一位数字是校验位。将卡号前面的数字采用 Luhn 算法就可以计算出最后一位数字。

Luhn 算法也被称作“模 10 算法”，是由 IBM 的科学家 Hans Peter Luhn 所创造，于 1954 年 1 月 6 日提出该专利的申请，并于 1960 年 8 月 23 日被授予，在美国的专利号为 2950048。它是一种简单的校验公式，目的是防止意外出现的错误。它的计算过程是：

（1）从校验位开始计数（校验位一般添加在账户的最后面），按从右向左的顺序，将偶数位的数字都乘以 2。

（2）将得到的结果相加起来（如果乘 2 的结果是两位数的话，那么就减去 9），然后再与原数字串的所有奇数位数字相加。

（3）如果加起来的和模 10 后为 0（也就是相加的结果是以 0 结尾的，10 的倍数），那么这个数字串根据 Luhn 算法来说就是有效的，反之就是无效的。假设一个字符串为“7992739871”，我们为其加上一个校验位，最后组成的数字为 7992739871x 的运算过程如下：

```

账户号码：      7  9  9  2  7  3  9  8  7  1  x
将偶数位乘以 2： 7  18 9  4  7  6  9  16 7  2  x
相加后的数字：   7  9  9  4  7  6  9  7  7  2 =67

```

校验码 x 是通过将相加后的数字乘以 9 后，再进行模 10 计算（那么就是：(67\*9) mod 10），取得的结果（3）就是校验位。



附录1：部分字符ASCII值(16进制数)

字 符	响铃	回车	换行	空格	\$	+	-	0	1	2	...	9
ASCII	07	0D	0A	20	24	2B	2D	30	31	32	...	39
字 符	A	B	C	...	Y	Z	a	b	c	...	y	z
ASCII	41	42	43	...	59	5A	61	62	63	...	79	7A

附录2：部分DOS系统功能调用参数（INT 21H,其中功能号为16进制数）

功能号	功能描述	调用参数	返回参数
1	键盘输入并回显	无	AL=输入字符
2	显示输出	DL=输出字符ASCII	无
5	打印机输出	DL=输出字符ASCII	无
9	显示字符串	DS:DX=串首址,以\$结束字符串	无
0A	键盘输入到缓冲区	DS:DX=缓冲区首地址 首字节保存缓冲区容量	(DS:DX+1)=实际输入的字符个数 (DS:DX+2)=输入的字符串开始地址
3C	建立文件	DS:DX=ASCIZ串地址 CX=文件属性(一般为00H)	CF=0时, AX=文件代号 CF=1时, 操作失败
3D	打开文件	DS:DX= ASCIZ串地址 AL=方式（0:读, 1:写）	CF=0时, AX=文件代号 CF=1时, 操作失败
3E	关闭文件	BX=文件代号	CF=1时, 操作失败
3F	读文件	DS:DX=缓冲区首地址 BX=文件代号 CX=读取字节数	CF=0时, AX=实际读字节数 当AX=0时表示文件结束 CF=1时, 操作失败
40	写文件	DS:DX=缓冲区首地址 BX=文件代号 CX=写字符个数	CF=0时, AX=实际写字节数 CF=1时, 操作失败
42	移动文件指针	CX: DX=移动指针的字节数 AL=方式码 0:从文件头往后移, 正数 1:从当前位置往前后移 2:从文件尾往前移, 正数 BX=文件代号	CF=0时, DX:AX=新指针位置 CF=1时, 操作失败
4C	程序结束	AL=返回码	

附录3：部分设备文件的的文件代号（HANDLE）

- 0:标准输入设备（即键盘）
- 1:标准输出设备（即显示器）
- 4:标准打印设备（即打印机）