

## 《汇编语言程序设计》期末复习题及答案

### 一、选择（30 题）

1. 下列是 8 位带符号二进制数的补码，其中最大的是（ B ）。

B、00000001

2. 在一段汇编语言程序中多次调用另一段程序，用宏指令比用子程序实现（ C ）

C、占内存空间大，但速度快

3. 数据定义为： EE DB 3, 4

CC DW 3456H, 6789H

执行指令“MOV CX, WORD PTR EE+1”后，(CX) = （ B ）

B、5604H

4. 有如下指令序列：

MOV AL, 95H

MOV CL, 2

SAR AL, CL

上述指令序列执行后，AL 的内容是（ D ）

D、0E5H

5. 设  $A=186_{10}$ ， $B=273_8$ ， $C=0BB_{16}$ ，把这三个数看成无符号数，则它们之间的关系是（ D ）

D、 $A < B = C$

6. 用一条指令实现寄存器 BX 和 SI 的内容相加，并将结果送入 AX 中，这条指令是（ D ）

D、LEA AX, [BX][SI]

7. 要从端口 1234H 中读入 8 位数据，其指令应是（ C ）

C、MOV DX, 1234H

IN AL, DX

8. 循环控制指令 LOOPNZ/LOOPNE 继续执行循环的条件是（ B ）

B、 $CX \neq 0$  且  $ZF=0$

9. 8088/8086 存储器分段，每个段不超过（ D ）

D、64K 个字节

10. 表示过程定义结束的伪指令是（ A ）

- A、ENDP
11. 用来存放下一条将要执行的指令代码段地址的段寄存器是 ( D )
- D、CS
12. 假定 (SS) =1000H, (SP) =0100H, (AX) =5609H, 执行指令 PUSH AX 后, 存放数据 56H 的物理地址是 ( D )
- D、100FFH
13. 执行 “SUB AX, [BP][DI]” 指令, 取源操作数时, 使用的物理地址表达式是 ( A )
- A、16\*SS+BP+DI
14. 设 AX 中有一带符号数 8520H, 执行 “SAR AX, 1” 指令后, AX 中的值是 ( C )
- C、0C290H
15. 设 (BX) =1234H, (DS) =2000H, (21234H) =5678H, 则指令 “LEA SI, [BX]” 执行后的结果是: ( B )
- B、SI=1234H
16. 测试 BL 中的数是否为负数, 若为负数则转移至 AA1 处, 横线处的指令应为 ( A )
- A、JNE
17. 设 DF=0, (SI) =20H, (CX) =10, 执行 “REP LODSW” 指令后, SI 中的内容是 ( C )
- C、34H
18. 在程序执行过程中, IP 寄存器中始终保存的是 ( B )
- B、下一条指令的首地址
19. 设 SP 初值为 2000H, 执行指令 “POP AX” 后, SP 的值是 ( A )
- A、2002H
20. 若 AX=3500H, CX=56B8H, 当 AND AX, CX 指令执行后, AX= ( A )
- A、1400H
21. 汇编源程序时, 出现语法错误的语句是 ( A )
- A、MOV DS, 1200H
22. 将数据 1234H 存放在存储单元中的伪指令是 ( C )
- C、DATA1 DB 34H, 12H
23. 下面指令执行后, 可能改变 AL 寄存器内容的指令是 ( D )
- D、AND AL, BL

24. 操作数地址存放在寄存器的寻址方式称为 ( B )

B、寄存器间接寻址

25. 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 )

B、汇编语言程序

26. 用 DOS 功能调用时，子功能调用的编号应放在 ( C )

C、AH

27. 提供串指令中目的操作数地址的是 ( C )

C、ES: [DI]

28. 在寄存器间接寻址方式下，能够提供有效地址的寄存器有 ( B )

B、BX, BP, SI, DI

29. 已知某操作数的物理地址是 2117AH，则它的段地址和偏移地址可能是 ( A )

A、2025:0F2A

30. 中断向量表中的每个中断向量所占内存空间为: ( B )

B、4 个字节

## 二、填空 (30 题)

1. 在用直接寻址方式寻址操作数时，系统对段地址的默认值在段寄存器 DS 中。

参见 P40

2. 与指令“NEG AX”功能等效的两条指令依次是 NOT AX 和 INC AX。

参见 P85、94、84

3. 语句“VAR DB 3 DUP (3, 3 DUP (3))”经过汇编后应分配 12 个字节单元。

参见 P58

4. 指令“MOV AX, SEG VAR”的执行，是把 变量 VAR 的段地址 送到 AX 中。

参见 P62

5. 设 (DS) = 1000H, (ES) = 2000H, (SS) = 3000H, (BP) = 4000H。

计算下列各指令中源操作数的物理地址:

MOV AX, [1000H]      物理地址 11000H

MOV AX, ES: [BP]      物理地址 24000H

参见 P29

6. 变量和标号均具有三种属性，即: 段属性、偏移属性、类型属性。

参见 P56、59

7. 在做乘法运算时，如果是做字运算，乘积放在 DX 和 AX 两个寄存器中，做除法运算时，若做字节运算，余数放在 AH 中，商放在 AL 中。

参见 P86、87

8. 用以访问存储器操作的地址表达式中，只要出现 BP 寄存器，系统就默认以段寄存器 SS 的内容作为存储器操作数的段地址。

参见 P40~44

9. 在数据段或代码段的最开始，常用伪指令“ORG n”来定义本段的偏移地址初值，该初值的范围是 0000H~FFFFH。

参见 P66、28

10. 指令“CWD”能实现对某寄存器中内容进行扩展，该寄存器是：AX。

参见 P89

11. 8088/8086CPU 能寻址的最大存储空间为 1MB，因此 CPU 的地址总线是 20 条。

参见 P28

12. 在应用串处理指令 MOVSB 时，源串首地址存放在寄存器 SI 中，目的串首地址存放在寄存器 DI 中，并且目的串必须在 附加 段中，执行串处理指令时，用方向标志位 DF 控制地址自动增减，数据串的长度存放在寄存器 CX 中。

参见 P125、126

13. 程序状态字 PSW 中的 ZF 位，当结果为 0 时，ZF= 1，结果为 1 时，ZF= 0。

参见 P33

14. 十六进制数 0D6H，当把它看作无符号数时，它表示的十进制正数为 214，当把它看作带符号数时，它表示的十进制负数为 -42。

参见 P22、23

15. IBM PC 机中用来存放段地址的段寄存器有四个：CS、DS、SS、ES。

参见 P31、32

### 三、判断题（15 题）

- |              |                               |
|--------------|-------------------------------|
| 1. SHL AL, 2 | ( 错 ) <u>参见 P95 2 改为 CL</u>   |
| 2. INC 2     | ( 错 ) <u>参见 P84、83 不能用立即数</u> |
| 3. ADD BX, 1 | ( 对 ) <u>参见 P83</u>           |

4. MOV [SI][DI], BX ( 对 ) 参见 P44
5. SUB BX, CL ( 错 ) 参见 P84 BX 与 CL 数据宽度不一致
6. OUT 34H, AL ( 对 ) 参见 P170
7. IN AL, 0AH ( 对 ) 参见 P169
8. MOV CX, [SP] ( 错 ) 参见 P42 SP 不能做间址寄存器
9. MUL AL ( 对 ) 参见 P86、87
10. SUB BX, BP ( 对 ) 参见 P84
11. MOV DS, SEG VARY ( 错 )  
参见 P62、46 SEG VARY 是立即数, 而 DS 不能接受立即数
12. MOV VARY[SI], ES: BL ( 错 )  
参见 P40 对寄存器不能使用段跨越前缀
13. CMP BLOCK, [100H] ( 错 )  
参见 P85 不能对两个内存单元直接进行比较
14. 宏调用时实元和哑元的个数可以不一致。 ( 对 ) 参见 P161
15. REP STOSB ( 对 ) 参见 P127

#### 四、写程序段 (9 题)

1. 计算 8\*9, 并将乘积送 SI。

(参见 P87, 例 5.10, 考查对乘法指令的掌握情况)

```
mov al,8
mov bl,9
mul bl
mov si,ax
```

2. 计算 35/6, 商存于 BL 中, 余数存于 CL 中。

(参见 P88, 例 5.13, 考查对乘法指令的掌握情况)

```
mov ax,35
mov bl,6
div bl
mov cl,ah
```

3. 输入一个字母, 然后输出它的后继字母 (不考虑输入字母是 Z 的情况)。

(参见 P102 第 6 题及 P186, 考查对 1 号、2 号功能调用的掌握)

```
mov ah,01h
```

```
int 21h
inc al
mov dl, al
mov ah, 02h
int 21h
```

4. 从端口 35H 读入一个字节数据，将该数的第 3 位置 1 后（其余位不变），再写回端口 35H。（说明：最低位为第 0 位）

（参见 P178，第 2 道问答题，考查对 I/O 指令的掌握）

```
in al, 35h
or al, 08h
out 35h, al
```

5. 定义数据段 NUM，数据段中的数据按下列要求依次存储：

（1）变量 X，类型为字节，初值为 17 ；

（2）数组变量 M，类型为字，所占空间为 100H 个单元；

（3）变量 MSG，存储字符串“LUCKY”。

（参见 P79，问答题的第 2 小题及 P181，考查对段定义伪操作的掌握）

```
NUM segment
    X db 17
    M dw 100h dup(?)
    MSG db 'LUCKY'
NUM ends
```

6. 寄存器 BX 中存放着一个带符号数，求出其绝对值，结果还放在 BX 中。

（参见 P107，例 6.3，考查对 CMP、NEG 指令的掌握）

```
cmp bx, 0
jg L1
neg bx
L1:...
```

7. 对存放在（DX，AX）中的双字数据作求补运算。

（参见 P83，例 5.2 及 P85，例 5.7，考查对双精度数的操作）

```
not dx
not ax
add ax, 1
adc dx, 0
```

8. 计算 5!，结果存于寄存器 SI 中。

（参见 P107，例 6.4，考查对循环指令的掌握）

```
mov cx, 5
mov al, 1
mov bl, 1
```

```

L1:mul bl
    inc bl
    loop L1
    mov si, ax

```

9. 定义一个名字为 ADDXYZ 的宏，实现  $W \leftarrow X+Y+Z$ 。（注：所有数据均为字类型）

（参见 P166，编程题第 4 题及 P201，考查对宏定义的掌握）

```

ADDXYZ MACRO X, Y, Z, W
    mov ax, x
    add ax, y
    add ax, z
    mov w, ax
endm

```

## 五、写程序（9 题）

1. 从键盘输入一个字符串（长度不超过 30），统计字符串中非数字的个数，并将统计的结果存入字单元 COUNT 中，用 EXE 格式实现。

（参见 P152~P154，考查字符数组、比较指令、循环等）

<pre> data segment     str db 30, ?, 30 dup(?)     count db 0 data ends code segment     assume cs:code, ds:data     main proc far start:     push ds     mov ax, 0     push ax      mov ax, data     mov ds, ax     lea dx, str     mov ah, 0ah     int 21h     mov cl, str+1 </pre>	<pre>     mov ch, 0     mov si, 2 L3:     cmp str[si], 30h     jae L1     inc count     jmp L2 L1:     cmp str[si], 39h     jbe L2     inc count L2:     inc si     loop L3     ret     main endp code ends end start </pre>
---	--

2. 统计一个 16 位二进制数中 1 的个数，并将结果输出，用 COM 格式实现。

（参见 P119，考查 COM 格式文件、移位指令、跳转指令）

<pre> code segment     org 100h     assume cs:code     main proc near start:     mov bx, 0a39h     mov si, 0     mov cx, 16     next:     shr bx, 1     jnc l1     inc si     l1:     loop next </pre>	<pre>     mov dx, si     add dx, 30h     cmp dl, 3ah     jb l2     add dl, 7     l2:     mov ah, 02h     int 21h     mov ax, 4c00h     int 21h     main endp     code ends end start </pre>
--	---

3. 从键盘输入两个一位十进制数，求它们的和，并将结果以十进制形式输出。

（只考虑和小于 10 的情况）

（参见 P156 第 6 题及 P200，考查 1 号功能调用、2 号功能调用、字符的输出）

<pre> code segment     assume cs:code     main proc far start:     push ds     mov ax, 0     push ax     mov ah, 01     int 21h     mov bl, al     mov ah, 01     int 21h </pre>	<pre>     mov bh, al     sub bx, 3030h     add bl, bh     mov dl, bl     add dl, 30h     mov ah, 02     int 21h     ret     main endp     code ends end start </pre>
--	--

4. 从键盘输入一个十进制个位数，在屏幕上显示相应数量的该数。

例如，输入 3，屏幕上将显示“333”。

（参见 P123 第 6 题及 P193，考查 1 号功能调用、字符如何变成数值、LOOP 指令）

<pre> code segment     assume cs:code     main proc far         push ds         mov ax, 0 </pre>	<pre>         push ax         mov ah, 01h         int 21h         mov bl, al         sub al, 30h </pre>
--	---



cbw	loop L1
mov cx, ax	ret
L1:	main endp
mov ah, 02h	code ends
mov dl, bl	end main
int 21h	

5. 求 100 以内所有奇数的和，存于字变量 X 中。

（参见 P143，例 8.6，考查循环指令）

data segment	mov ax, 0
x dw ?	mov bx, 1
data ends	mov cx, 50
code segment	L1: add ax, bx
assume cs:code, ds:data	add bx, 2
main proc far	loop L1
start:	mov x, ax
push ds	ret
mov ax, 0	main endp
push ax	code ends
mov ax, data	end start
mov ds, ax	

6. 将 BX 中的数以二进制形式在屏幕上显示出来。

（参见 **P123** 第 **2** 题及 **P189**，考查移位指令、跳转指令、数值的输出、**2** 号功能调用）

code segment	L2: mov ah, 02h
assume cs:code	int 21h
main proc far	loop L1
start:	mov ax, 4c00h
mov bx, 1234h	int 21h
mov cx, 16	main endp
L1: mov dl, 30h	code ends
shl bx, 1	end start
jnc L2	
inc dl	

7. 字节数组 X 中存放着 0~F 共 16 个十六进制数，请将这些数以十六进制形式显示在屏幕上。

（参见 P115，例 6.9，考查对数组的使用、数值的输出、2 号功能调用、循环指令）

data segment	x	db
--------------	---	----

```

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0ah, 0bh, 0ch, 0dh, 0eh, 0fh
data ends
code segment
    assume cs:code, ds:data
    main proc far
start:
    mov ax, data
    mov ds, ax
    mov cx, 16
    mov si, 0
L1: mov dl, x[si]
    add dl, 30h
    cmp dl, 39h
    jbe L2
    add dl, 7
L2: mov ah, 02h

```

```

int 21h
inc si
loop L1
mov ax, 4c00h
int 21h
main endp
code ends
end start

```

8. 从包含 10 个无符号数的字节数组 array 中选出最小的一个数存于变量 MIN 中，并将该数以十进制形式显示出来。

（参见 P123，第 3 题及 P190～191，考查对数组的使用、循环指令、比较指令、除法指令、数值的输出）

```

data segment
    array db 45, 67, 75, 82, 90, 29, 80, 26, 63, 59
    min db ?
data ends
stack segment stack
    dw 10 dup(?)
stack ends
code segment
    assume cs:code, ds:data, ss:stack
    main proc far
start:
    mov ax, data
    mov ds, ax
    mov cx, 10
    mov al, 255
    mov si, 0
    mov di, 0
L1: cmp al, array[si]
    jbe L2

```

```

    mov
    al, array[si]
L2: inc si
    loop L1
    mov min, al
    mov ah, 0
    mov cx, 3
L3: mov bl, 10
    div bl
    mov dl, ah
    push dx
    inc di
    mov ah, 0
    cmp ax, 0
    jz L4
    loop L3
L4: mov cx, di
L5: pop dx
    add dl, 30h

```

mov ah, 02h		main endp
int 21h		code ends
loop L5		end start
mov ax, 4c00h		
int 21h		

9. 设在起始地址为 STRING 的存储空间存放了一个字符串，统计字符串中字符“A”的个数，并将结果存放到 RESULT 字节单元中。

（参见 P134 第 5 题及 P196，考查串处理指令、比较指令）

data segment		lea si, string
string db 'assembly'		L1: lodsb
len dw \$-string		cmp al, 'a'
result db 0		jnz L2
data ends		inc result
code segment		L2: loop L1
assume cs:code, ds:data		mov ax, 4c00h
main proc far		int 21h
start:		main endp
mov ax, data		code ends
mov ds, ax		end start
mov cx, len		