

一、假设 (AX)=1234H, (BX)=5678H, (SP)=1000H, 指出执行下面的程序段后, 各相关寄存器及堆栈段中的内容。

```
PUSH    AX    ; (AX)=?, (BX)=?, (SP)=?, 栈顶字节[SP]=?, 栈顶第二字节[SP+1]=?
PUSH    BX    ; (AX)=?, (BX)=?, (SP)=?, 栈顶字节[SP]=?, 栈顶第二字节[SP+1]=?
POP     AX    ; (AX)=?, (BX)=?, (SP)=?, 栈顶字节[SP]=?, 栈顶第二字节[SP+1]=?
```

答: PUSH AX; (AX)=1234H, (BX)=5678H, (SP)=0FFEh, [SP]=34H, [SP+1]=12H  
PUSH BX; (AX)=1234H, (BX)=5678H, (SP)=0FFCh, [SP]=78H, [SP+1]=56H  
POP AX; (AX)=5678H, (BX)=5678H, (SP)=0FFEh, [SP]=34H, [SP+1]=12H

二、设已编写好类型码为 60H 的中断服务程序为:

```
INT-ROUT  PROC  FAR
            .
            .
            .
            IRET
INT-ROUT  ENDP
```

请编写一段程序, 实现该程序的中断矢量在矢量表中的装填

答: PUSH DS  
PUSH DX  
MOV AX, SEG INT-ROUT  
MOV DS, AX  
MOV DX, OFFSET INT-ROUT  
MOV AL, 60H  
MOV AH, 25H  
INT 21H  
POP DX  
POP DS

三、选择题

(1) 已知 AX=0008H, 执行 TEST AX, 1 后, AX 寄存器的内容是 ( D )。

- A. 0005H
- B. FFFBH
- C. 0003H
- D. 0008H

(2) 在串操作指令前使用重复前缀指令 REPE, 终止串的重复操作条件是 ( C )。

- A. CX=0 且 ZF=0
- B. CX=0 且 ZF=1
- C. CX=0 或 ZF=0
- D. CX=0 或 ZF=1

(3) 若有 BUF DW 1, 2, 3, 4 , 则可将数据 02H 取到 AL 寄存器中的指令为 ( B )。

- A. MOV AL, BYTE PTR BUF+1
- B. MOV AL, BYTE PTR BUF+2
- C. MOV AL, BYTE PTR BUF+3
- D. MOV AL, BYTE BUF 2

(4) 当执行完下列指令后, 标志位 CF、OF 和 SF 的值为 ( D )。

```
MOV AL, 3AH
ADD AL, 7CH
```

- A. 1, 1, 0                      B. 1, 0, 1                      C. 0, 0, 1                      D. 0, 1, 1

(5) 执行下列指令后, 正确的结果是 ( B )。

```
MOV AL, 100
MOV BL, -2
```

- A. AL=100H  
BL=02H
- B. AL=64H  
BL=0FEH
- C. AL=64H  
BL=82H
- D. AL=100H  
BL=0FEH

(6) 实现将 DX: AX 中存放的 32 位数扩大四倍, 正确的程序段是 ( C )。

- A. SHL AX, 2  
ROL DX, 2
- B. RCL AX, 2  
SHL DX, 2
- C. MOV CX, 2  
LOP: SHL AX, 1  
RCL DX, 1  
LOOP LOP
- D. SHL AX, 1  
SHL AX, 1  
RCL DX, 1  
RCL DX, 1

(7) 已知 AX=0001H, 下列四条指令执行后, 不能使 ZF=1 的语句是 ( C )。

- A. DEC AX
- B. SUB AX, 1
- C. AND AX, 1
- D. CMP AX, 1

- (8) 当一个带符号数大于 0FBH 时程序转移，需选用的条件转移指令是 ( C )。
- A. JLE  
B. JNL  
C. JNLE  
D. JL
- (9) 为了使执行串操作指令时，地址按减量方式处理应使用指令 ( A )。
- A. STD B. STI C. CLD D. CLI
- (10) 指令 JMP word PTR [BX] 转移的目标地址是 ( B )。
- A. BX 的内容  
B. BX 指出的内存单元之内容  
C. IP+[BX] 之和  
D. SP+BX 之和
- (11) 下列指令中，唯一不含有非法操作数寻址的指令是 ( C )。
- A. ADD [BX], [2000H] B. ADC [SI][DI], AX  
C. SUB [2000H], 2000H D. SBC AX, EX
- (12) 以 8086/8088 为 CPU 的微型计算机，串操作指令中，凡是目的串的地址就取自 ( D )。
- A. DS:SI B. DS:DI C. ES:SI D. ES:DI
- (13) 以 8086/8088 为 CPU 的微型计算机，串操作指令中，凡是目的源串的地址就取自 ( A )。
- A. DS:SI B. DS:DI C. ES:SI D. ES:DI
- (14) 执行下列程序后，AL 的内容为 ( B )
- ```
MOV AL, 58H
SUB AL, 75H
DAS
```
- A. 17H B. 83H C. 0E3H D. 27H
- (15) 下列指令中，错误的是 ( C )。
- A. MOV AX, 100 B. POPF  
C. ADD OFFSET[BX][SI], DX D. DEC DI

#### 四、填空题

- (1) 在串操作指令中，源操作数的段地址所在寄存器为 DS。
- (2) 软中断指令 INT 30H，其中的中断类型号为 30H，该中断的服务程序的入口地址在 0000H: [C0H]-0000H: [C3H] 的四个内存单元中。
- (3) 设 (AX) = 0304H, (BX) = 0FF00H，则执行指令 AND AX, BX 后，(AX) = 0300H。
- (4) INT 中断指令将 CS, IP, PSW 压入堆栈的次序依次为 PSW, CS, IP。中断向量共有 256 个。

(5) 条件转移指令 JNE 条件成立是 ZF= 0。

(6) 8086/8088 通用寄存器中可以作为寄存器间接寻址的寄存器是 SI、DI、BP、BX。

(7) 当 SP=4328H，执行 POPF 指令后，栈顶的有效地址是 432AH。

(8) 指令 MOV AX, [BX+DI] 的源操作数寻址属于 基址变址 寻址方式。

(9) 设双字数据 1A2B3C4DH，存放于首地址为 60000H 的数据区中，则该数据的各字节按照存储地址由低到高的顺序依次为：4DH，3CH，2BH 和 1AH。

(10) 设数据段有如下定义：

ORG 200H

BUF DB 2H, 4H

.....

代码段有下列语句：

MOV AX, WORD PTR BUF

则在代码段执行上述指令语句后，AX 中的内容是 0402H。

(11) 使用远过程调用指令 CALL 和使用中断指令 INT 均需保护现场，应保护的内容，对 CALL 指令为 CS、IP；对 INT 指令为 标志寄存器、CS、IP。

(12) DA1 EQU WORD PTR DA2 ;OCDABH

DA2 DB 0ABH, OCDH

SHL DA1, 1

SHR DA2, 1

上述两条指令执行后，DA2 中两字节单元内容分别是 2BH、9BH。

(13) 指令 JMP NEAR PROG1，在程序代码段中的偏移地址为 2013H（这是该指令第一字节的偏移地址）。组成该指令的三字节机器码为 E91234H（其中 E9H 为操作码）。执行该指令后，程序转移去的偏移地址为 5428H。

(14) 当 (AL) < 80H 时，执行 CBW 后，(AH) = 00H；当 (AL) > 80H 时，执行 CBW 后，(AH) = 0FFH。

(15) 检查 BUF 字节单元的内容是否为正偶数，完成下面程序段填空：

MOV AL, BUF

TEST AL, 81H

JNE K1

..... ; 是正偶数

K1: .....

五、从键盘输入两个数，计算他们相加的结果，并显示到屏幕上。需要检查输入的是否为数字，结果是否溢出。

DOSBox 0.74a, Cpu speed: 3000 cycles, Frameskip 0, Program: EDIT

File Edit Search View Options Help

C:\ADD1.ASM

```

;-----
;data segment
;buff db 6, ?, 6 dup (?)
;count db 2
;result dw ?
;msg1 db 'invalid number', 0dh, 0ah, '$'
;msg2 db 'overflow', 0dh, 0ah, '$'
;data ends
;
;code segment
;    assume cs:code, ds:data
;main proc far
;    push ds
;    mov ax, 0
;    push ax
;    mov ax, data
;    mov ds, ax
;in1:
;    lea dx, buff
;    mov ah, 0ah
;    int 21h
;    mov dl, 0ah
;    mov ah, 02h
;
;F1=Help

```

Line:1 Col:1

```

DOS
BOX
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: EDIT
File Edit Search View Options Help
C:\ADD1.ASM
:
: mov ah, 02h
: int 21h
: mov bx, 0
: lea si, buff + 2
: mov ah, 0
: cld
:
: 1:
: lodsb
: cmp al, 30h
: jb err1
: cmp al, 39h
: ja err1
: and al, 0fh
: add bx, bx
: jc err2
: mov cx, bx
: add bx, bx
: jc err2
: add bx, bx
: jc err2
: add bx, cx
: jc err2
:
F1=Help
Line:43 Col:1

```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: EDIT

File Edit Search View Options Help

C:\ADD1.ASM

```
add bx, cx
jc err2
add bx, ax
jc err2
dec buff + 1
jnz l
dec count
jz in2
mov result, bx
jmp in1
in2:
add bx, result
jc err2
mov ax, bx
mov cx, 0h
mov bx, 000ah
mov dx, 0h
out1:
div bx
or dx, 30h
push dx
mov dx, 0h
```

F1=Help Line:63 Col:1

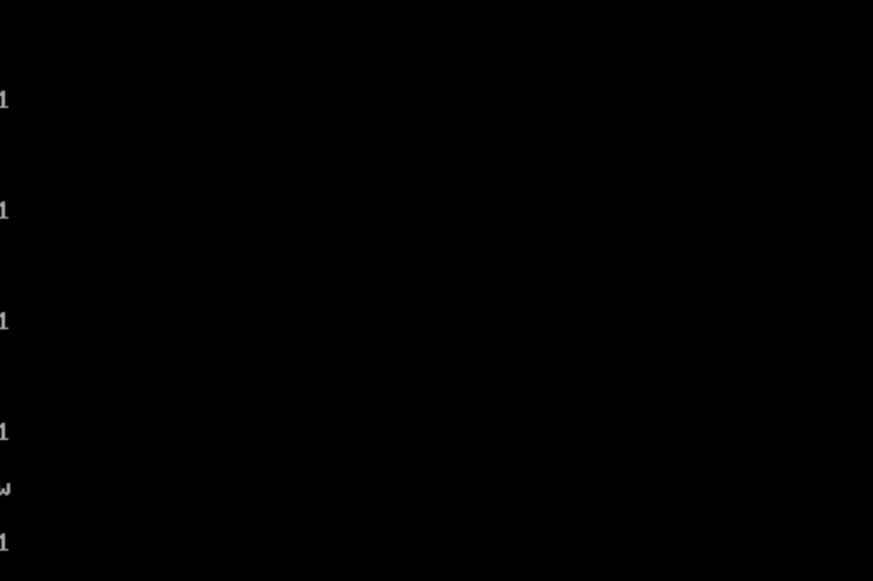
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: EDIT

File Edit Search View Options Help

C:\ADD1.ASM

```
mov dx, 0h
inc cx
cmp ax, 0h
jne out1
out2:
pop dx
mov ah, 02h
int 21h
dec cx
jnz out2
done:
ret
err1:
lea dx, msg1
jmp output
err2:
lea dx, msg2
output:
mov ah, 09h
int 21h
jmp done
main endp
```

F1=Help Line:84 Col:1



DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
C:\>add1
2345
32662
35007
C:\>add1
24563
8888
33451
C:\>add1
11111
111
11222
C:\>add1
9999
9999
19998
C:\>add1
99999
overf low
C:\>add1
we23
invalid number
C:\>
```

六、数据段直接定义 10 个数 x, 10 个数 y, x 与 y 按照以下结果产生 z

$$z_1=x_1+y_1 \quad z_2=x_2-y_2 \quad z_3=x_3+y_3 \quad z_4=x_4+y_4 \quad z_5=x_5-y_5 \quad z_6=x_6+y_6 \quad z_7=x_7-y_7 \quad z_8=x_8-y_8$$
$$z_9 = x_9 + y_9 \quad z_{10} = x_{10} - y_{10}$$

在 debug 下运行程序，检查  $z$  是否正确

```

DOS
Box
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: EDIT
File Edit Search View Options Help
----- C:\XYZ.ASM -----
;data segment
;da1 db 1h,2h,3h,4h,5h,6h,7h,8h,9h,0ah
;da2 db 0ah,9h,8h,7h,6h,5h,4h,3h,2h,1h
;op db 1,0,1,1,0,1,0,0,1,0
;result db 10 dup (?)
;data ends
;
;code segment
;    assume cs:code, ds:data, es:data
;begin:
;    mov ax, data
;    mov ds, ax
;    mov es, ax
;    lea si, da1
;    lea di, result
;    lea bx, da2
;    mov cx, 0ah
;L:
;    lodsb
;    mov ah, [bx + 0ah]
;    cmp ah, 1h
;    jne sub1

```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: EDIT

File Edit Search View Options Help

C:\XYZ.ASM

```

: lea si, da1
: lea di, result
: lea bx, da2
: mov cx, 0ah
:L:
: lodsb
: mov ah, [bx + 0ah]
: cmp ah, 1h
: jne sub1
: add al, [bx]
: jmp next
:sub1:
: sub al, [bx]
:next:
: add bx, 1
: stosb
: loop L
: mov ah, 4ch
: int 21h
:code ends
: end begin

```

F1=Help Line:35 Col:1

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

LINK : warning L4021: no stack segment

C:\>debug xyz.exe

-u

|           |          |       |            |
|-----------|----------|-------|------------|
| 06CD:0000 | B8CA06   | MOV   | AX,06CA    |
| 06CD:0003 | 8ED8     | MOV   | DS,AX      |
| 06CD:0005 | 8EC0     | MOV   | ES,AX      |
| 06CD:0007 | 8D360000 | LEA   | SI,[0000]  |
| 06CD:000B | 8D3E1E00 | LEA   | DI,[001E]  |
| 06CD:000F | 8D1E0A00 | LEA   | BX,[000A]  |
| 06CD:0013 | B90A00   | MOV   | CX,000A    |
| 06CD:0016 | AC       | LODSB |            |
| 06CD:0017 | 8A670A   | MOV   | AH,[BX+0A] |
| 06CD:001A | 80FC01   | CMP   | AH,01      |
| 06CD:001D | 7505     | JNZ   | 0024       |
| 06CD:001F | 0207     | ADD   | AL,[BX]    |

-g

Program terminated (0009)

-d06ca:0 1 40

|           |                                                 |               |
|-----------|-------------------------------------------------|---------------|
| 06CA:0000 | 01 02 03 04 05 06 07 08-09 0A 0A 09 08 07 06 05 | .....         |
| 06CA:0010 | 04 03 02 01 01 00 01 01-00 01 00 00 01 00 0B F9 | .....         |
| 06CA:0020 | 0B 0B FF 0B 03 05 0B 09-00 00 00 00 00 00 00    | .....         |
| 06CA:0030 | BB CA 06 8E D8 8E C0 8D-36 00 00 8D 3E 1E 00 8D | .....6...>... |