

2、

data segment

str1 db 'Happy '

str2 db 'Love '

str3 db 'Marriage'

str4 db 'Hate '

data ends

code segment

assume cs: code, ds: data

main proc far

start:

push ds

sub ax, ax

push ax

mov ax, data

mov ds, ax

mov es, ax ; 1)

mov ah, 01h ; 2)

int 21h

mov bx, 0

mov dl, 0

search:

mov cl, 3 ; 3)

push bx

shl bx, cl

lea di, [bx] ; 4)

mov cx, 8 ; 5)

cld

repnz scasb ; 6)

jne next

inc dl

next:

pop bx ; 7)

inc bx

cmp bx, 4

jnz search

add dl, '0' ; 8)

mov ah, 02h ; 9)

int 21h

ret

main endp

code ends

```
end start ; 10)
```

3、

```
data segment
```

```
    A dw 5
```

```
    B dw 10
```

```
    C dw 100
```

```
    D dw 5
```

```
data ends
```

```
code segment
```

```
    assume cs: code, ds: data
```

```
main proc far
```

```
start:
```

```
    push ds
```

```
    sub ax, ax
```

```
    push ax
```

```
    mov ax, data
```

```
    mov ds, ax
```

```
    mov ax, B
```

```
    mov cx, 10
```

```
    imul cx
```

```
    add ax, C
```

```
    adc dx, 0
```

```
    mov cx, D
```

```
    idiv cx
```

```
    add ax, A
```

```
    ret
```

```
main endp
```

```
code ends
```

```
end start
```

4、

```
data segment
```

```
    arrayA db 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
```

```
    arrayB db 20, 18, 16, 14, 12, 10, 8, 6, 4, 2, 0, 21, 23, 25, 27, 29, 31, 33, 34,
```

```
    35
```

```
data ends
```

```
code segment
```

```
    assume cs: code, ds: data
```

```
main proc far
```

```
start:
```

```
    push ds
```

```

        sub ax, ax
        push ax

        mov ax, data
        mov ds, ax

        mov bx, 0
        mov si, 0
loopA:   mov di, 0
loopB:   mov al, arrayB[di]
        cmp arrayA[si], al
        jnz next
        inc bx
next:    inc di
        cmp di, 20
        jl loopB
        inc si
        cmp si, 15
        jl loopA
        call bin2dec
        call crlf
        ret
main endp

bin2dec proc near
        push cx
        mov cx, 10d
        call decdiv
        mov cx, 1d
        call decdiv
        pop cx
        ret
bin2dec endp

decdiv proc near
        mov ax, bx
        mov dx, 0
        div cx
        mov dx, bx
        mov dl, al
        add dl, '0'

```

```
        mov ah, 02h
        int 21h
        ret
decdiv endp
```

```
crlf proc near
        mov dl, 13
        mov ah, 02h
        int 21h
        mov dl, 10
        mov ah, 02h
        int 21h
        ret
crlf endp
code ends
        end start
```

```
5、
data segment
        count db 1
        str db 'busy', 13, 10, '$'
data ends
```

```
code segment
        assume cs: code, ds: data
main proc far
start:
        push ds
        sub ax, ax
        push ax

        mov ax, data
        mov ds, ax

        mov al, 1ch
        mov ah, 35h
        int 21h
        push es
        push bx

        push ds
        mov ax, seg requery
        mov ds, ax
        mov dx, offset requery
        mov al, 1ch
```

```
mov ah, 25h
int 21h
pop ds
```

```
in al, 21h
and al, 11111110b
out 21h, al
sti
```

```
mov si, 10000
delay: mov di, 10000
```

```
delay1: dec di
jnz delay1
dec si
jnz delay
```

```
cli
pop dx
pop ds
mov al, 1ch
mov ah, 25h
int 21h
ret
```

```
main endp
```

```
requery proc near
push ds
push ax
push dx
```

```
mov ax, data
mov ds, ax
sti
dec count
jnz exit
mov count, 18
mov dx, 397h
in al, dx
test al, 0ffh
jns exit
lea dx, str
mov ah, 09h
```

```
        int 21h
exit:    cli
        pop dx
        pop ax
        pop ds
        iret
requery endp
code ends
        end start
```