**1.题目：输入1，输出9；输入2，输出8，输入3，输出7；……输入9，输出1。输入其他字符则重新输入。用地址表实现。**

代码：

.model small

.stack 100h

.data

    table   dw t1, t2, t3, t4, t5, t6, t7, t8, t9

    crlf    db 0dh, 0ah, '$'

    s0      db 'please input number between 1 to 9$'

    s1      db '9$'

    s2      db '8$'

    s3      db '7$'

    s4      db '6$'

    s5      db '5$'

    s6      db '4$'

    s7      db '3$'

    s8      db '2$'

    s9      db '1$'

.code

printStr MACRO strAddr

    push dx

    push ax

    lea dx, strAddr

    mov ah, 9

    int 21h

    pop ax

    pop dx

ENDM

start:

    mov ax, @data

    mov ds, ax

input:

    lea dx, s0

    mov ah, 9

    int 21h

    printStr crlf

    mov ah, 1

    int 21h

    ; store the input char in bx

    mov bx, ax

    printStr crlf

    cmp bl, '1'

    jb input

    cmp bl, '9'

    ja input

    and bx, 000fh

    dec bx

    shl bx, 1

    jmp table[bx]

display:

    mov ah, 9

    int 21h

    printStr crlf

    mov ah, 4ch

    int 21h

t1:

    lea dx, s1

    jmp display

t2:

    lea dx, s2

    jmp display

t3:

    lea dx, s3

    jmp display

t4:

    lea dx, s4

    jmp display

t5:

    lea dx, s5

    jmp display

t6:

    lea dx, s6

    jmp display

t7:

    lea dx, s7

    jmp display

t8:

    lea dx, s8

    jmp display

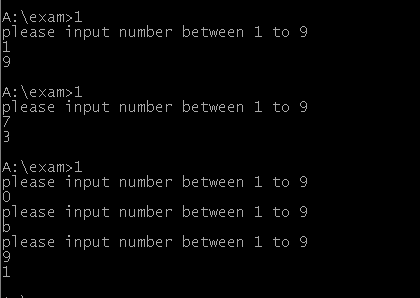
t9:

    lea dx, s9

    jmp display

end start

运行结果：



**4.找出与我的学号（2018211991）后两位最近的质数。输入输出+堆栈实现。**

代码：

.model small

.stack 100h

.data

    xueHao  dw 0

    up      dw 0

    down    dw 0

    crlf    db 0dh, 0ah, '$'

.code

printStr MACRO strAddr

    push dx

    push ax

    lea dx, strAddr

    mov ah, 9

    int 21h

    pop ax

    pop dx

ENDM

judge proc

    ; param:  xueHao: [bp + 4]

    ; return: flag: [bp + 6]

    push bp

    mov bp, sp

    push ax

    push bx

    push cx

    push dx

    mov bx, [bp + 4]

    mov cx, [bp + 4]

    sub cx, 2

    cmp bx, 2

    jle isPrime

    mov dl, 1

Prime:

    mov ax,bx

    inc dl

    div dl

    cmp ah, 0

    jz notPrime

    loop prime

    jmp isPrime

notPrime:

    mov word ptr [bp + 6], 0

    jmp endOfJudge

isPrime:

    mov word ptr [bp + 6], 1

endOfJudge:

    pop dx

    pop cx

    pop bx

    pop ax

    pop bp

    ret 2

judge endp

printUNum proc

    ; print unsigned number

    ; param al: num

    push cx

    push dx

    mov cx, 0

    pushRemainders:

    and ah, 0

    mov dl, 10

    div dl

    mov dl, ah

    and dh, 0

    push dx

    inc cx

    cmp al, 0

    jne pushRemainders

    popRemainders:

    pop dx

    or dl, 30h

    mov ah, 2

    int 21h

    loop popRemainders

    pop dx

    pop cx

    ret

printUNum endp

start:

    mov ax, @data

    mov ds, ax

    mov es, ax

    ; input xueHao

    mov ah, 1

    int 21h

    mov dh, al

    and dh, 0fh

    mov ah, 1

    int 21h

    mov dl, al

    and dl, 0fh

    mov al, dh

    mov dh, 10

    mul dh

    add al, dl

    cbw

    mov xueHao, ax

    printStr crlf

    mov ax, xueHao

    mov cx, 0

    cmp cx, 0

    ja endUp

        sub sp, 2

        push ax

        call judge

        pop dx

        cmp dx, 1

        je notFindUp

        mov cx, 1

        mov up, ax

        notFindUp:

        inc ax

    endUp:

    ; .WHILE flag == 0

    ;     sub sp, 2

    ;     push ax

    ;     call judge

    ;     pop dx

    ;     .IF dx == 1

    ;         mov flag, 1

    ;         mov up, ax

    ;     .ENDIF

    ;     inc ax

    ; .ENDW

    mov ax, xueHao

    mov cx, 0

    cmp cx, 0

    ja endDown

        sub sp, 2

        push ax

        call judge

        pop dx

        cmp dx, 1

        jne notFindDown

        mov cx, 1

        mov up, ax

        notFindDown:

        dec ax

    endDown:

    ; .WHILE flag == 0

    ;     sub sp, 2

    ;     push ax

    ;     call judge

    ;     pop dx

    ;     .IF dx == 1

    ;         mov flag, 1

    ;         mov up, ax

    ;     .ENDIF

    ;     dec ax

    ; .ENDW

    ; find which one is closer to xueHao

    mov di, up

    mov ax, xueHao

    sub di, ax

    mov ax, down

    mov si, xueHao

    sub si, ax

    cmp di, si

    ; di < si: printUp

    jb printUp

    ja printDown

    printUp:

    mov ax, up

    call printUNum

    jmp endOfMain

    printDown:

    mov ax, down

    call printUNum

    endOfMain:

    mov ah, 4ch

    int 21h

end start

结果：

