## Lab05

### Task#1

.model small

.stack 100h

.data

msg1 db 10,13, "Enter First number: $"

msg2 db 10,13, "Enter Second number: $"

msg3 db 10,13, "Numbers are equal $"

msg4 db 10,13, "Numbers are not equal $"

.code

main proc

mov ax, @data

mov ds, ax

mov dx, offset msg1

mov ah, 09h

mov ah, 01h

int 21h

sub al, 30h

mov cl, al

mov dx, offset msg2

mov ah, 09h

int 21h

mov ah, 01h

int 21h

sub al, 30h

mov dl, al

cmp dl, cl

je equal

mov dx, offset msg4

mov ah, 09h

int 21h

jmp end\_prog

equal:

mov dx, offset msg3

mov ah, 09h

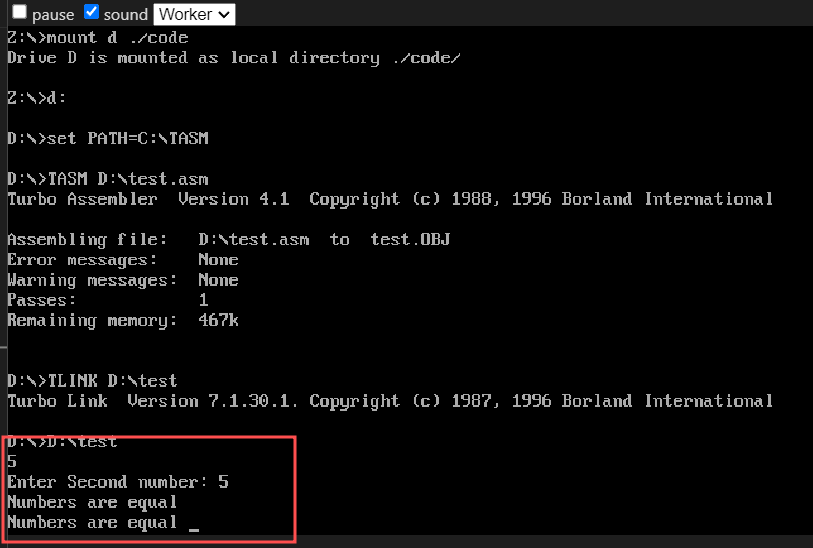
int 21h

end\_prog:

int 21h

main endp

end main



### Task#2

.model small

.stack 100h

.data

    var01 db 1,3,5,9,0,2,4,6,7,8

.code

main proc

    mov ax, @data

    mov ds, ax

    mov cx, 10

    mov si, offset var01

print\_loop:

    mov dl, [si]

    add dl, '0'

    mov ah, 02h

    int 21h

    inc si

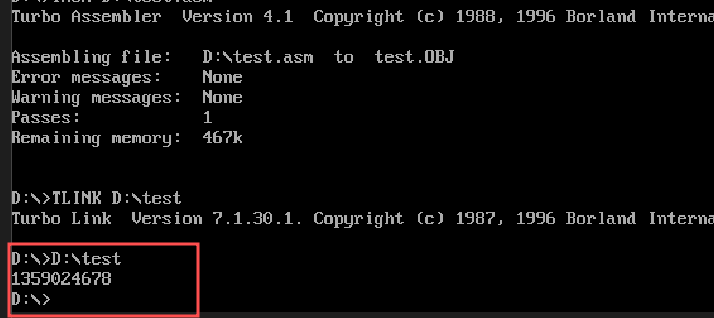
    loop print\_loop

    mov ax, 4c00h

    int 21h

main endp

end main



### Task#3

.model small

.stack 100h

.data

    var01 db 'l','t','x','k','w','s','i','q','d','y','g','n','z','m','b','e','h','u','a','r','c','v','p','j','f','o'

.code

main proc

    mov ax, @data

    mov ds, ax

    mov cx, 26

    mov si, offset var01

print\_loop:

    mov dl, [si]

    mov ah, 02h

    int 21h

    inc si

    loop print\_loop

    mov ax, 4c00h

    int 21h

main endp

end main

A screenshot of a computer

Description automatically generated

### Task#4

.model small

.stack 100h

.data

    var01 db 'N','B','V','G','Y','F','X','P','Z','R','W','U','C','I','Q','J','H','T','O','A','K','L','S','D','E','M'

.code

main proc

    mov ax, @data

    mov ds, ax

    mov cx, 26

    mov si, offset var01

print\_loop:

    mov dl, [si]

    mov ah, 02h

    int 21h

    inc si

    loop print\_loop

    mov ax, 4c00h

    int 21h

main endp

end main

A screenshot of a computer

Description automatically generated

### Task#5

.model small

.stack 100h

.data

msg1 db 10,13,"It is Even $"

msg2 db 10,13,"It is Odd $"

var01 db 6

.code

main proc

mov ax,@data

mov ds,ax

mov al,var01

and al,1

jnz odd

even:

mov ah,09h

lea dx,msg1

int 21h

jmp exit

odd:

mov ah,09h

lea dx,msg2

int 21h

exit:

mov ah,4ch

int 21h

main endp

end main

A screenshot of a computer

Description automatically generated

### Task#7

.model small

.stack 100h

.data

    var01 db 1,1,1,1,1

.code

main proc

    mov ax, @data

    mov ds, ax

    mov cx, 5

    mov si, offset var01

    mov al, 0

l1:

    add al, [si]

    inc si

    loop l1

    add al,30h

    mov dl, al

    mov ah, 02h

    int 21h

    mov ax, 4c00h

    int 21h

main endp

end main

A screenshot of a computer

Description automatically generated

### Task#6

.model small

.stack 100h

.data

    var0 db 6,9,3,6,7,2

    arr db 6 dup(?)

    newline db 0Dh, 0Ah, '$'

.code

main proc

    mov ax, @data

    mov ds, ax

    mov cx, 6

    lea si, var0

    lea di, arr

    call process\_numbers

    lea si, arr

    mov cx, 6

    call print\_array

    mov ah, 4ch

    int 21h

main endp

process\_numbers proc

    mov cx, 6

    l1:

        mov al, [si]

        and al, 1

        jnz odd

        mov byte ptr [di], 'e'

        jmp next

    odd:

        mov byte ptr [di], 'o'

    next:

        inc di

        inc si

        loop l1

    ret

process\_numbers endp

print\_array proc

    l2:

        mov dl, [si]

        mov ah, 02h

        int 21h

        inc si

        loop l2

    mov dx, offset newline

    mov ah, 09h

    int 21h

    ret

print\_array endp

end main

A screenshot of a computer

Description automatically generated