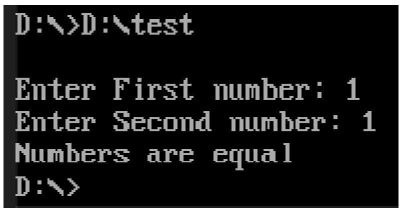
COAL LAB 5
ABDUL SAMI QASIM
22I-1725
CY-D

TASK1:

```
;code to compare two numbers
.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
    ; Display message to enter the first number
    mov dx, offset msg1
    mov ah, 09h
    int 21h
    ; Read the first number
    mov ah, 01h      ; Function to read a character from STDIN
    int 21h          ; Call DOS interrupt
    sub al, 30h      ; Convert ASCII to numeric value
    mov cl, al       ; Store the first number
    ; Display message to enter the second number
    mov dx, offset msg2
    mov ah, 09h
    int 21h
    ; Read the second number
    mov ah, 01h      ; Function to read a character from STDIN
    int 21h          ; Call DOS interrupt
    sub al, 30h      ; Convert ASCII to numeric value
    mov dl, al       ; Store the second number
    ; Compare the two numbers
    cmp dl, cl
    je equal        ; If equal, jump to label1
    ; If not equal, print the message
    mov dx, offset msg4
    mov ah, 09h
    int 21h
    jmp end_prog
equal:
    ; If equal, print the message
    mov dx, offset msg3
    mov ah, 09h
    int 21h
end_prog:
;end_prog: This is the end of the program. It terminates the program using DOS interrupt 4Ch.
    mov ah, 4ch
    int 21h
main endp
end main
```

OUTPUT 1:



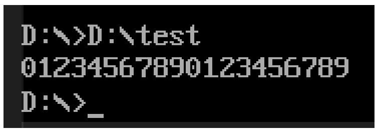
```
D:\>D:\test

Enter First number: 1
Enter Second number: 1
Numbers are equal
D:\>
```

TASK2:

```
.model small
.stack 100h
.data
array1 db 10 dup(?),0
val1 db 0
.code
main proc
    mov ax, @data
    mov ds, ax
    mov si, 0
    mov cx,10
loop1:
mov al,val1
add al,048d
mov array1[si], al
inc val1
inc si
int 21h
loop loop1

mov si, offset array1
mov cx, 10
```

```
; loop
l1:
mov dx, [si]
;add dx, 30h
mov ah,2
int 21h
;mov dx, [si+1]
inc si
loop l1

mov ah, 4ch
int 21h
main endp
end main
```

OUTPUT2:



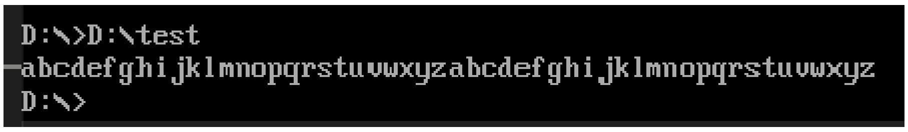
```
D:\>D:\test
01234567890123456789
D:\>_
```

TASK3:

```
.model small
.stack 100h
.data
array1 db 26 dup(?),0
val1 db 0
.code
main proc
mov ax, @data
mov ds, ax
mov si, 0
mov cx,26
loop1:
mov al,val1
add al,030h
mov array1[si], al
inc val1
inc si
int 21h
loop loop1

mov si, offset array1
mov cx, 26



; loop
l1:
mov dx, [si]
add dx, 31h
mov ah,2
int 21h
```

```
;mov dx, [si+1]
inc si
loop l1

mov ah, 4ch
int 21h
main endp
end main
```

OUTPUT3:

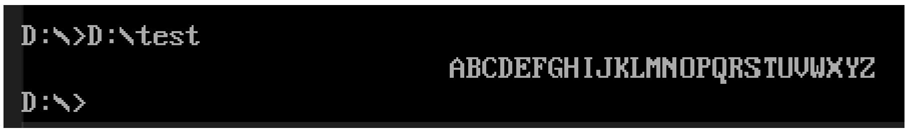


```
D:\>D:\test
abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz
D:\>
```

TASK4:

```
.model small
.stack 100h
.data
array1 db 26 dup(?),0
val1 db 0
.code
main proc
    mov ax, @data
    mov ds, ax
    mov si, 0
    mov cx,26
loop1:
mov al,val1
;add al,030h
mov array1[si], al
inc val1
inc si
int 21h
loop loop1

mov si, offset array1
mov cx, 26



; loop
l1:
mov dx, [si]
add dx, 065d
mov ah,2
int 21h
;mov dx, [si+1]
inc si
loop l1

mov ah, 4ch
int 21h
main endp
end main
```

OUTPUT4:

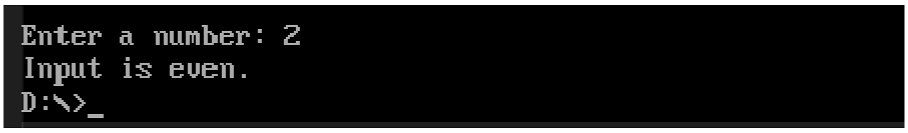


```
D:\>D:\test
                             ABCDEFGHIJKLMNOPQRSTUVWXYZ
D:\>
```

TASK5:

```
.model small
.stack 100h
.data
    prompt db 10, 13, "Enter a number: $"
    even_msg db 10, 13, "Input is even. $"
    odd_msg db 10, 13, "Input is odd. $"
.code
main proc
    mov ax, @data
    mov ds, ax

    mov dx, offset prompt
    mov ah, 09h
    int 21h

    mov ah, 01h
    int 21h

    sub al, 30h  ; Convert ASCII digit to numeric value

    mov ah, 02h
    test al, 01b
    jz even_l

    mov dx, offset odd_msg
    mov ah, 09h
    int 21h
    jmp end_prog

even_l:
    mov dx, offset even_msg
    mov ah, 09h

end_prog:
    int 21h

    mov ah, 4ch
    int 21h
main endp
end main
```

OUTPUT5:

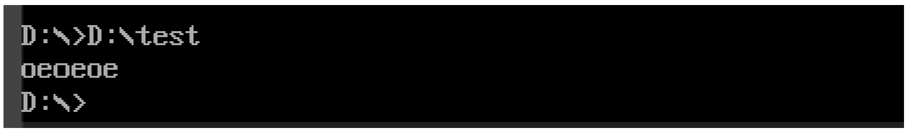


```
Enter a number: 2
Input is even.
D:\>_
```

TASK6:

```
.model small
.stack 100h
.data
    array1 db 1, 2, 3, 4, 5, 6
    even_char db 'e'
.code
main proc
    mov ax, @data
    mov ds, ax

    mov cx, 6
    mov si, 0

loop1:
    test byte ptr array1[si], 01b
    jnz odd_l

    mov byte ptr array1[si], 'e'
    jmp next_element

odd_l:
    mov byte ptr array1[si], 'o'

next_element:
    inc si
    loop loop1

    ; Print the result
    mov cx, 6
    mov si, offset array1

print_loop:
    mov dl, [si]

    mov ah, 02h
    int 21h

    inc si
    loop print_loop

    mov ah, 4ch
    int 21h
main endp
end main
```

OUTPUT6:

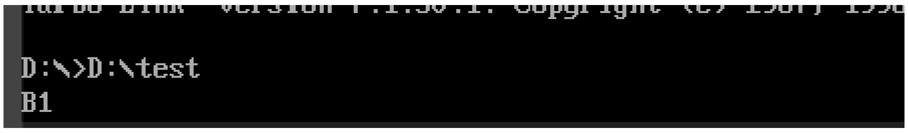


```
D:\>D:\test
oeoeoe
D:\>
```

TASK7:

```
.model small
.stack 100h
.data
    array db 1, 2, 3, 4, 5
    sum db ?
.code
main proc
    mov ax, @data
    mov ds, ax

    mov cx, 5 ; Number of elements in the array
    mov si, offset array ; Point SI to the array
    ; Clear AX to store the sum

sum_loop:
    add ax, [si] ; Add the current element to the sum
    inc si ; Move to the next element (2 bytes for each element)

    loop sum_loop ; Repeat the loop for the remaining elements

AAM
mov ch,ah
mov cl,al

mov dl,ch
add dl, 48
mov ah,2
int 21h
mov dl,cl
add dl, 48
mov ah, 2
int 21h
main endp
end main
```

OUTPUT7:



```
Turbo Link  Version 7.1.30.1. Copyright (c) 1987, 199

D:\>D:\test
B1
```