**1. Write a program in assembly language to take two single-digit numbers as input and**

**display whether they are equal or not.**

**CODE**

.MODEL SMALL

.STACK 100H

.DATA

PROMPT1 DB 'Enter first single-digit number: $'

PROMPT2 DB 0DH, 0AH, 'Enter second single-digit number: $'

EQUAL\_MSG DB 0DH, 0AH, 'The numbers are equal.$'

NOT\_EQUAL\_MSG DB 0DH, 0AH, 'The numbers are not equal.$'

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

; Prompt and take the first number as input

LEA DX, PROMPT1

MOV AH, 09H

INT 21H ; Display prompt1

MOV AH, 01H

INT 21H ; Take first input (ASCII)

SUB AL, '0' ; Convert ASCII to integer (0-9)

MOV BL, AL ; Store the first number in BL

; Prompt and take the second number as input

LEA DX, PROMPT2

MOV AH, 09H

INT 21H ; Display prompt2

MOV AH, 01H

INT 21H ; Take second input (ASCII)

SUB AL, '0' ; Convert ASCII to integer (0-9)

; Compare the two numbers

CMP BL, AL

JE EQUAL ; If equal, jump to EQUAL label

; Numbers are not equal

LEA DX, NOT\_EQUAL\_MSG

MOV AH, 09H

INT 21H ; Display "The numbers are not equal"

JMP EXIT ; Jump to exit

EQUAL:

LEA DX, EQUAL\_MSG

MOV AH, 09H

INT 21H ; Display "The numbers are equal"

EXIT:

MOV AH, 4CH ; Terminate program

INT 21H

**OUTPUT**

A screenshot of a computer

Description automatically generated A computer screen shot of a black screen

Description automatically generated

**2. Write a program in assembly language to check whether a single-digit number is odd or**

**even**.

**//CODE**//

.MODEL SMALL

.STACK 100H

.DATA

PROMPT DB 'Enter a single-digit number: $'

EVEN\_MSG DB 0DH, 0AH, 'The number is even.$'

ODD\_MSG DB 0DH, 0AH, 'The number is odd.$'

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

; --- Input Section ---

; Display prompt to enter a number

LEA DX, PROMPT ; Load address of prompt message

MOV AH, 09H ; DOS service to display string

INT 21H ; Display "Enter a single-digit number: "

MOV AH, 01H ; DOS service to take single character input

INT 21H ; Take input (ASCII code in AL)

SUB AL, '0' ; Convert ASCII to integer (0-9)

; --- Odd/Even Check Section ---

; Check if the number is divisible by 2 (even or odd)

MOV BL, AL ; Store the input number in BL

MOV AH, 0 ; Clear AH for division

MOV AL, BL ; Move the number to AL for division

MOV CL, 2 ; Divisor is 2

DIV CL ; Divide the number by 2 (AL = AL/2, remainder in AH)

CMP AH, 0 ; Check the remainder (stored in AH)

JE EVEN ; If remainder is 0, number is even, jump to EVEN label

; --- Output Section for Odd Case ---

LEA DX, ODD\_MSG ; Load address of odd message

MOV AH, 09H ; DOS service to display string

INT 21H ; Display "The number is odd."

JMP EXIT ; Jump to exit

EVEN:

; --- Output Section for Even Case ---

LEA DX, EVEN\_MSG ; Load address of even message

MOV AH, 09H ; DOS service to display string

INT 21H ; Display "The number is even."

JMP EXIT ; Jump to exit

EXIT:

; Program termination

MOV AH, 4CH ; DOS service to terminate program

INT 21H

MAIN ENDP

END MAIN

**OUTPUT**

**A computer screen with a black screen

Description automatically generatedA computer screen with a black screen

Description automatically generated**