**Lab practical 09**

**COA**

**Question:**

**Write a program in assembly language to take two single-digit numbers as input and display whether they are equal or not.**

**Code:**

**ORG 100h**

**; Display the message "Enter the first digit: "**

**MOV DX, OFFSET msg\_input1**

**MOV AH, 09h**

**INT 21h**

**; Read the first digit from the user**

**MOV AH, 01h**

**INT 21h**

**SUB AL, '0' ; Convert ASCII to integer**

**MOV BL, AL ; Store the first digit in BL**

**; Display the message "Enter the second digit: "**

**MOV DX, OFFSET msg\_input2**

**MOV AH, 09h**

**INT 21h**

**; Read the second digit from the user**

**MOV AH, 01h**

**INT 21h**

**SUB AL, '0' ; Convert ASCII to integer**

**MOV CL, AL ; Store the second digit in CL**

**; Compare the two digits**

**CMP BL, CL ; Compare the two digits**

**JE digits\_equal ; Jump if equal**

**; Display message for not equal**

**MOV DX, OFFSET msg\_not\_equal**

**MOV AH, 09h**

**INT 21h**

**JMP end\_program**

**digits\_equal:**

**; Display message for equal**

**MOV DX, OFFSET msg\_equal**

**MOV AH, 09h**

**INT 21h**

**end\_program:**

**; Print a new line (CRLF)**

**MOV DL, 0Dh**

**MOV AH, 02h**

**INT 21h**

**MOV DL, 0Ah**

**INT 21h**

**; Terminate the program**

**MOV AH, 4Ch**

**INT 21h**

**; Data section with messages**

**msg\_input1 DB 'Enter the first digit: $'**

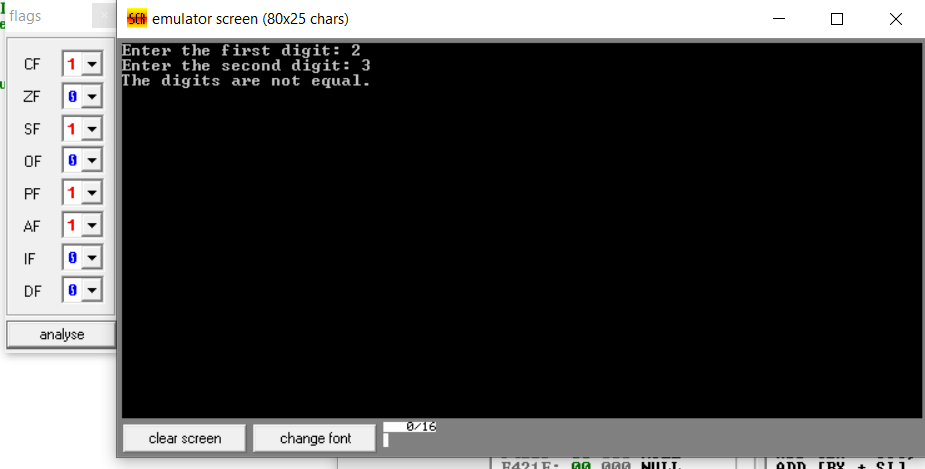
**msg\_input2 DB 0Dh, 0Ah, 'Enter the second digit: $'**

**msg\_equal DB 0Dh, 0Ah, 'The digits are equal.$'**

**msg\_not\_equal DB 0Dh, 0Ah, 'The digits are not equal.$'**

**END**

**OUTPUT:**



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

**even.**

**CODE:**

**ORG 100h**

**; Display the message "Enter a single-digit number: "**

**MOV DX, OFFSET msg\_input**

**MOV AH, 09h**

**INT 21h**

**; Read the digit from the user**

**MOV AH, 01h**

**INT 21h**

**SUB AL, '0' ; Convert ASCII to integer**

**; Check if the digit is between 0 and 9**

**CMP AL, 0 ; Check if less than 0**

**JB invalid\_input ; If below 0, jump to invalid input**

**CMP AL, 9 ; Check if greater than 9**

**JA invalid\_input ; If above 9, jump to invalid input**

**; Check if the number is odd or even**

**AND AL, 1 ; AND with 1 to check the least significant bit**

**JZ even\_number ; If result is 0, it's even**

**; Display message for odd**

**MOV DX, OFFSET msg\_odd**

**MOV AH, 09h**

**INT 21h**

**JMP end\_program**

**even\_number:**

**; Display message for even**

**MOV DX, OFFSET msg\_even**

**MOV AH, 09h**

**INT 21h**

**JMP end\_program**

**invalid\_input:**

**; Display invalid input message**

**MOV DX, OFFSET msg\_invalid**

**MOV AH, 09h**

**INT 21h**

**end\_program:**

**; Terminate the program**

**MOV AH, 4Ch**

**INT 21h**

**; Data section with messages**

**msg\_input DB 'Enter a single-digit number: $'**

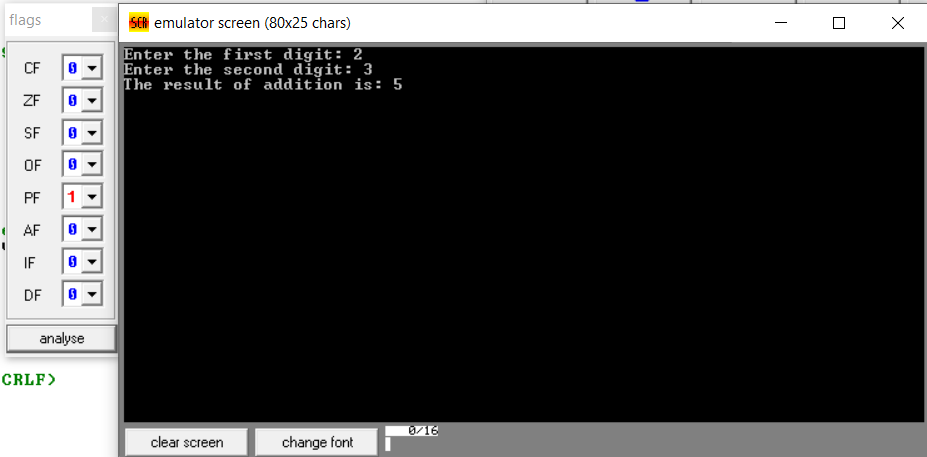
**msg\_even DB 0Dh, 0Ah, 'The number is even.$'**

**msg\_odd DB 0Dh, 0Ah, 'The number is odd.$'**

**msg\_invalid DB 0Dh, 0Ah, 'Invalid input. Please enter a single-digit number (0-9).$'**

**END**

**OUTPUT:**



**GITHUB:**

[**https://github.com/srijachakilam15/COA/blob/main/lab%20practical%2003.docx**](https://github.com/srijachakilam15/COA/blob/main/lab%20practical%2003.docx)

s