



UITs
UNIVERSITY OF INFORMATION
TECHNOLOGY AND SCIENCES

LAB Report

COURSE TITLE – Microprocessor Lab

COURSE CODE – CSE 360

Submitted To

Md. Ismail

Lecturer of UITs

Submitted By

Md. Shoyaif Rahman (0432310005101050)

Semester: Autumn-2025 (6th)

Department: CSE

Batch: 53

Section: 6A2

Date: 10/09/25

Assembly Language Lab Report

Lab report: 05

Experiment Name: print the characters in an array

Process:

Printing character in an array:

1. Start the program and set up the stack and data segment (MOV AX,@DATA and MOV DS,AX).
2. Define the string 'var' in the data segment containing 'sohan'.
3. Initialize loop registers:
 - CX = 5 → number of characters to print.
 - SI = OFFSET var → pointer to the start of the string.
4. Start the loop (l1) to print each character in the string.
5. Load the character from the string into DL using MOV DL,[SI].
6. Print the character using DOS interrupt INT 21h with AH = 2.
7. Increment SI to point to the next character in the string.
8. Loop back (LOOP l1) until CX reaches zero.
9. Exit the program with MOV AH,4Ch and INT 21h.

Printing array of strings:

1. Start the program and set up the stack and data segment (MOV AX,@DATA and MOV DS,AX).
2. Define the string 'Msg' in the data segment containing 'Hello! ', 'bollo ', 'ckoolo '.
3. Initialize SI register to point to the start of the string (MOV SI, OFFSET Msg).
4. Start the loop (PrintLoop) to print each character in the string.
5. Load the current character from memory into AL using MOV AL,[SI].
6. Compare AL with 0 (CMP AL,0) to check for the end of the string.
7. If the end is reached (JE Exit), jump to program exit.
8. Otherwise, move AL to DL (MOV DL,AL) and set AH=2 for DOS print character function.
9. Call DOS interrupt 21h (INT 21H) to print the character.
10. Increment SI (INC SI) to point to the next character.
11. Jump back to PrintLoop to repeat the process for the next character.
12. Exit the program (MOV AH,4Ch and INT 21h) when all characters are printed.

2. Implementation (Program Code – ASM)

1.Printing array of characters:

```
.model small
.stack 100
.data
var db 'sohan'
.code

main proc

    mov ax,@data
    mov ds,ax

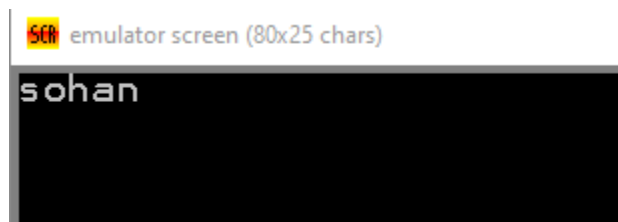
    mov cx,5
    mov si,offset var

l1:
    mov ah,2
    mov dx,[si]
    int 21h

    inc si
loop l1:
    exit:
        mov ah,4ch
        int 21h

main endp
end main
```

output:



2.Printing array of strings:

```
.MODEL SMALL
.STACK 100H
.DATA
    Msg DB 'Hello! ', 'bollo ', 'ckoolo '

.CODE
MAIN PROC
    MOV AX, @DATA
    MOV DS, AX

    MOV si, OFFSET Msg
PrintLoop:
    MOV AL, [si]
    CMP AL, 0
    JE Exit

    MOV AH, 2
    MOV DL, AL
    INT 21H
    INC si
    JMP PrintLoop

Exit:
    MOV AH, 4CH
    INT 21H
MAIN ENDP
END MAIN
```

output:

 emulator screen (80x25 chars)



```
Hello! bollo ckoolo
```

3.Result

The first program prints the string 'sohan' on the screen, displaying each character one by one using a loop and DOS interrupt 21h. The second program prints the combined string 'Hello! bollo ckoolo ' sequentially, looping through each character until it reaches the end, also using DOS interrupt 21h. Both programs demonstrate the use of loops, string traversal, and basic DOS character output to display text on the screen.

4. Conclusion

Both programs effectively demonstrate how to use loops and DOS interrupt 21h in 8086 assembly language for string output. The first program illustrates sequential character printing by looping through the string "sohan", while the second program shows how a longer combined string such as "Hello! bollo ckoolo " can be displayed character by character until completion. These examples reinforce the concepts of string traversal, loop control, and basic character output, highlighting the fundamental techniques for handling and displaying text in assembly language.