

**Heaven's light is our guide**



**Rajshahi University of Engineering And Technology**

**Dept. of Computer Science & Engineering**

**Details**

Course No : CSE 3010

Course Title : Microprocessors and Assembly Language

Date of Submission : 26-06-2022

<b>Submitted To</b>	<b>Submitted By</b>
<b>Sadia Zaman Mishu</b> <b>Assistant Professor</b> <b>RUET,CSE</b>	<b>Nazmul Haque</b> <b>Roll : 1803109</b> <b>Section : B</b> <b>Department : CSE</b>

## Lab-1

### Problem Statement :

Write a program to (a) prompt the user, (b) read first, middle, and last initials of a person's name, and (c) display them down the left margin.

### Discussion:

In this problem I need to take three character using mov ah,1 function and store this value in the predeclared variable in data segment. To use that variable we need access the data in main code using mov ax, @data mov ds,dx . then print the stored values using mov ah,2 function. Mov operation is applicable between memory location and register but not both memory location.

### Code:

1. include "emu8086.inc"
2. .model small
3. .stack 100h
4. .data
- 
5. ch1 db ?
6. ch2 db ?
7. ch3 db ?
- 
8. .code
- 
9. main proc
10. mov ax,@data
11. mov ds,ax
- 
12. print "Enter three character : "
- 
13. mov ah,1
14. int 21h
15. mov ch1,al

16. mov ah,1  
17. int 21h  
18. mov ch2,al

19. mov ah,1  
20. int 21h  
21. mov ch3,al

22. printn  
23. print "Output : "  
24. printn

25. mov ah,2  
26. mov dl,ch1  
27. int 21h

28. printn

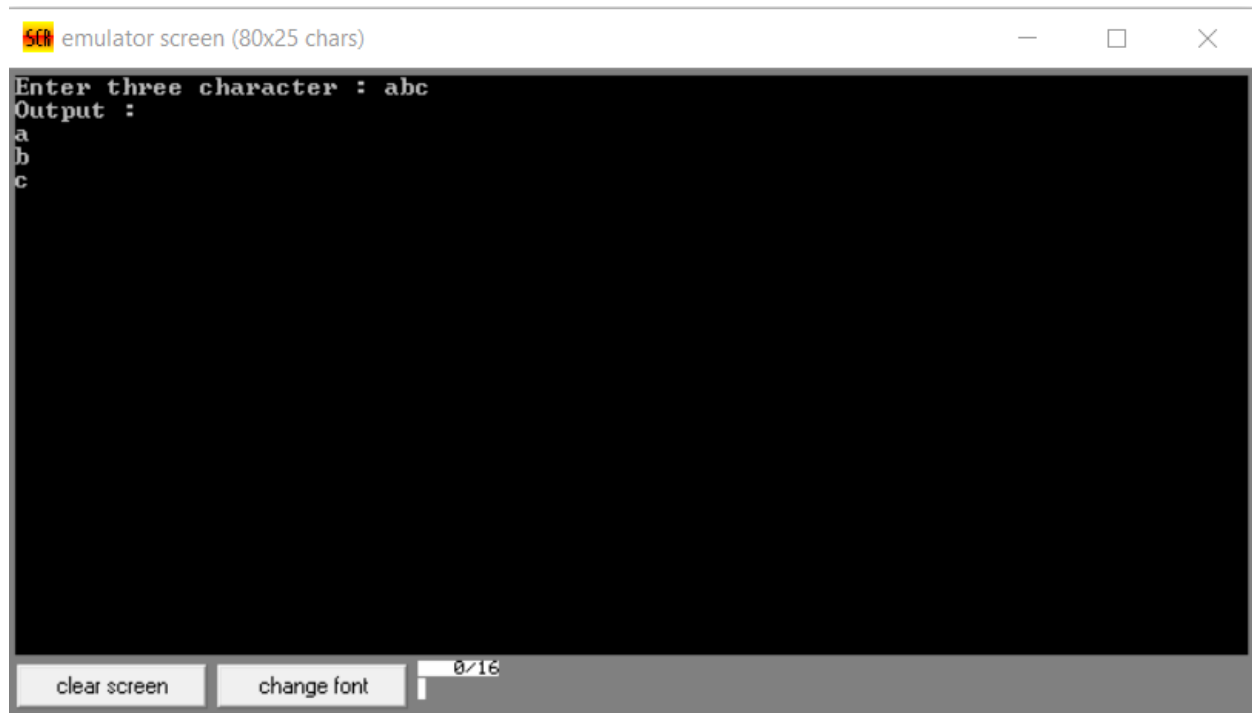
29. mov ah,2  
30. mov dl,ch2  
31. int 21h

32. printn

33. mov ah,2  
34. mov dl,ch3  
35. int 21h

36. mov ah,4ch  
37. int 21h  
38. main endp  
39. end main

## **Input & Output:**



**Conclusion:** This program gives write output according to the given input. Here include header file to print easily.

## Lab-2

### Problem Statement :

Write a program that prompts the user to enter two binary numbers of up to 8 digits each, and prints their sum on the next line in binary , reverse the sum and total number of '1' in the sum. Each input ends with a carriage return.

### Discussion:

In this problem I need to take two binary string. When we take input at first we need to confirm a register is empty. Using SHL , take user input and find sum using ADD operation.

To reverse the sum I use SHR operation similarly. To find the number of 1 I use ROL operation and increment the value of al lower byte register.

Algorithm:

Binary Input:

1. Clear a register which hold binary value
2. Input a character '0' or '1'
3. WHILE character <> CR DO
4. Convert character to binary value
5. Left shift the register Insert value into lsb of BX register
6. Input a character
7. END\_WHILE

Binary Output:

1. FOR 16 times DO
2. Rotate left the BX register which holds output value, put ·msb into CF
3. IF CF := 1
4. THEN
5. output '1'
6. ELSE
7. output '0'
8. END\_IF
9. END\_FOR

### Code:

1. INCLUDE "EMU8086.INC" ; HERE ADD HEADER FILE TO USE PRINT,PRINTN FUNCTION
2. .MODEL SMALL
3. .STACK 100H ;STACK SEGMENT
4. .DATA ; DATA SEGMENT
5. .CODE ; CODE SEGMENT
6. COUNT DB 0

```

7.  SUM1 DW ?
8.  SUM2 DW ?
9.  SUM3 DW ?
10. NUM_OF_1 DB 0
11. MAIN PROC      ; MAIN FUNCTION

12. MOV AX,@DATA
13. MOV DS,AX

14. XOR BX,BX
15. XOR CX,CX

16. PRINT "ENTER NUMBER 1 : "
17. MOV AH,1      ; TAKE FIRST INPUT
18. LOOP1:
19. INT 21H
20. CMP AL,0DH
21. JE SECOND_INPUT
22. SUB AL,48
23. SHL BL,1
24. OR  BL,AL
25. JMP LOOP1

26. SECOND_INPUT:
27. PRINTN
28. PRINT "ENTER NUMBER 2 : "

29. MOV AH,1      ; TAKING SECOND INPUT
30. LOOP2:
31. INT 21H
32. CMP AL,0DH
33. JE SUMATION

34. SUB AL,48
35. SHL CL,1
36. OR  CL,AL

37. JMP LOOP2
38. SUMATION:      ; CALCULATE THE SUM OF TWO NUMBER
39. ADD SUM1,BX

```

```

40. ADD SUM1,CX
41. ADD SUM2,BX
42. ADD SUM2,CX
43. ADD SUM3,BX
44. ADD SUM3,CX
45. PRINTN
46. PRINT "SUM OF TWO BINARY NUMBER : "
47. SUM:
48. CMP COUNT,16
49. JE RIVERSE
50. SHL SUM1,1
51. JC ONE
52. JNC ZERO
53. ONE:
54. MOV AH,2
55. MOV DL,"1"
56. INT 21H
57. INC COUNT
58. JMP SUM
59. ZERO:
60. MOV AH,2
61. MOV DL,"0"
62. INT 21H
63. INC COUNT
64. JMP SUM

65. RIVERSE1:
66. PRINTN
67. PRINT "RIVERSE OF SUM : " ; HERE REVERSE THE BIT STRING OF SUM
68. MOV COUNT,0
69. RIVERSE2:
70. CMP COUNT,16
71. JE FINISH
72. SHR SUM2,1
73. JC ONE1
74. JNC ZERO
75. ONE1:
76. MOV AH,2
77. MOV DL,"1"
78. INT 21H
79. INC COUNT
80. JMP RIVERSE2
81. ZERO:
82. MOV AH,2
83. MOV DL,"0"
84. INT 21H
85. INC COUNT
86. JMP RIVERSE2

```

```
87. FINISH:
88. PRINTN
89. PRINT "NUMBER OF 1 : "      ; HERE COUNT THE NUMBER OF 1
90. MOV CX,16
91. MOV AL,0
```

```
92. COUNT1:
93. ROL SUM3,1
94. JNC NEXT
95. INC AL
```

```
96. NEXT:
97. LOOP COUNT1
```

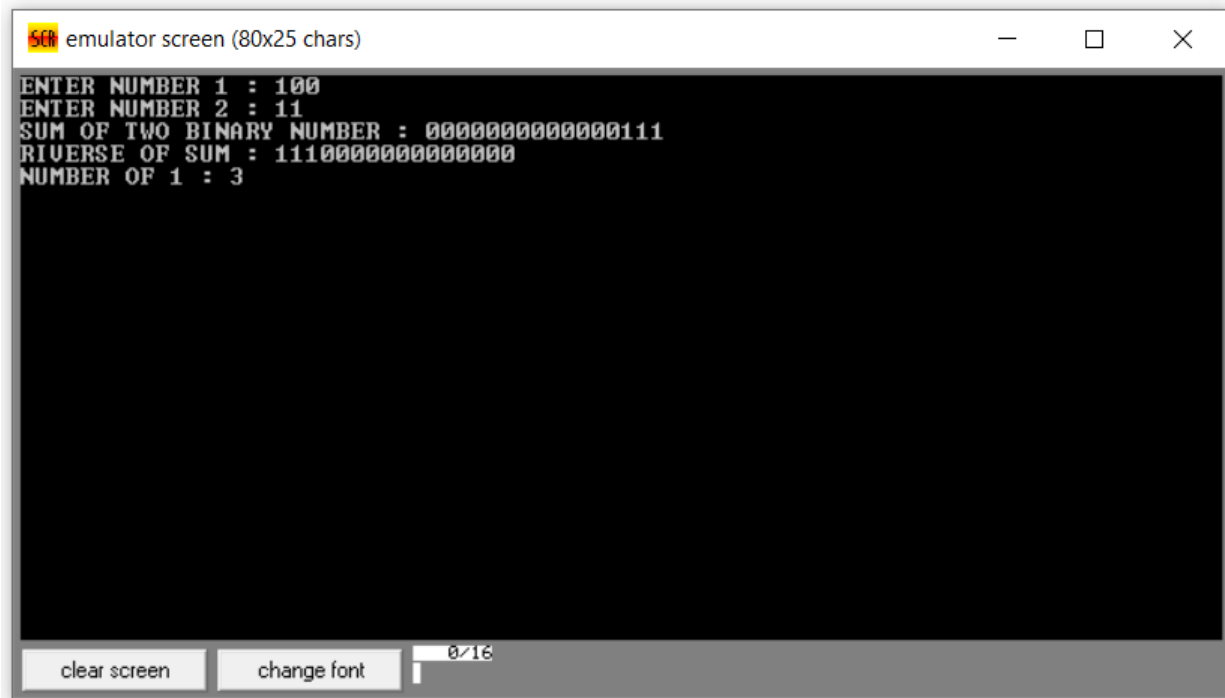
```
98. ADD AL,48
99. MOV AH,2
100.MOV DL,AL
101.INT 21H
```

```
102.EXIT:      ; DOS RETURN
103.MOV AH,4CH
104.INT 21H
```

```
MAIN ENDP      ; TERMINATE PROGRAM
END MAIN
```

## **Input & Output:**





The image shows a window titled "emulator screen (80x25 chars)". The window contains a black area with white text. The text displays the results of a program: "ENTER NUMBER 1 : 100", "ENTER NUMBER 2 : 11", "SUM OF TWO BINARY NUMBER : 0000000000000111", "RIVERSE OF SUM : 1110000000000000", and "NUMBER OF 1 : 3". At the bottom of the window, there is a grey bar with two buttons labeled "clear screen" and "change font", and a small display showing "0/16".

```
emulator screen (80x25 chars)
ENTER NUMBER 1 : 100
ENTER NUMBER 2 : 11
SUM OF TWO BINARY NUMBER : 0000000000000111
RIVERSE OF SUM : 1110000000000000
NUMBER OF 1 : 3
clear screen change font 0/16
```

**Conclusion:** This program gives write output according to the given input. I faces at first difficulties of using 1 byte and 2 byte variable in same mov operation.