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

Submitted To	Submitted By
Sadia Zaman Mishu **Assistant Professor** **RUET,CSE**	**Nazmul Haque** **Roll : 1803109** **Section : B** **Department : CSE**

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:

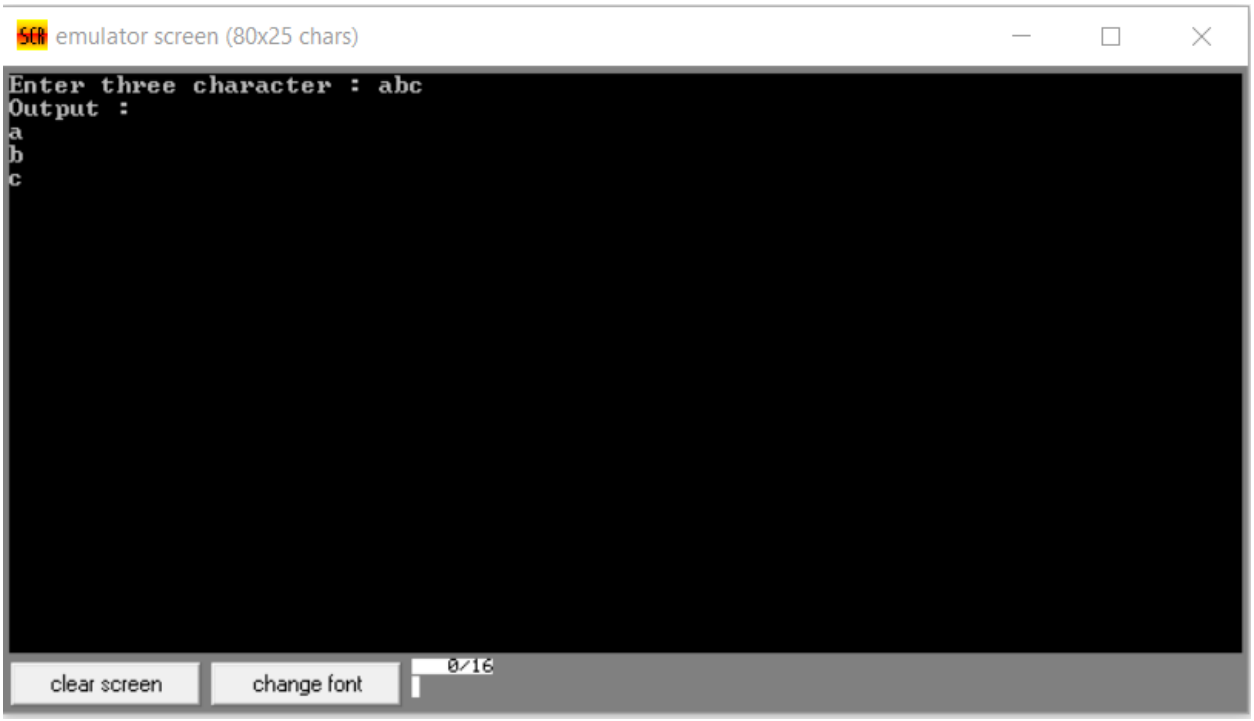
In this problem I need to take three character using **mov ah,1** function and store this value in the predeclared variable in data segment. To use that variable we need access the data in main code using **mov ax , @data mov ds,dx** . then print the stored values using mov ah,2 function. Mov operation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

Code:

```
include "emu8086.inc"
.model small
.stack 100h
.data


ch1 db ?
ch2 db ?
ch3 db ?


.code


main proc
mov ax,@data
mov ds,ax


print "Enter three character : "


mov ah,1
int 21h
mov ch1,al
```

```
mov ah,1
int 21h
mov ch2,al

mov ah,1
int 21h
mov ch3,al

printn
print "Output : "
printn

mov ah,2
mov dl,ch1
int 21h

printn

mov ah,2
mov dl,ch2
int 21h

printn

mov ah,2
mov dl,ch3
int 21h

mov ah,4ch
int 21h
main endp
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 'l'
6. ELSE
7. output ‘0’
8. END_IF
9. END_FOR

Code:

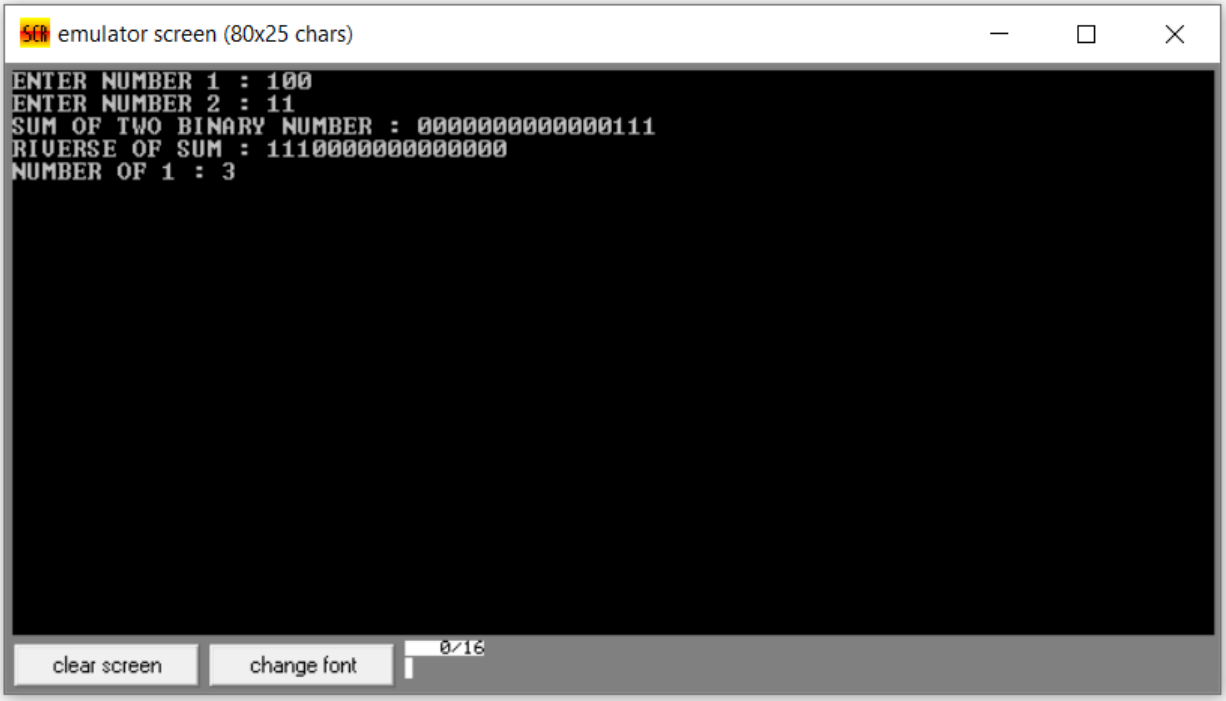
```
1.  INCLUDE "EMU8086.INC"      ; HERE ADD HEADER FILE TO USE PRINT,PRINTN FUNCTION
2.  .MODEL SMALL
3.  .STACK 100H    ;STACK SEGMENT
4.  .DATA          ; DATA SEGMENT
5.  .CODE          ; CODE SEGMENT
6.  COUNT DB 0
```

```
SUM1 DW ?
SUM2 DW ?
SUM3 DW ?
NUM_OF_1 DB 0
MAIN PROC          ; MAIN FUNCTON


MOV AX,@DATA
MOV DS,AX


XOR BX,BX
XOR CX,CX


PRINT "ENTER NUMBER 1 : "
MOV AH,1                    ; TAKE FIRST INPUT
LOOP1:
INT 21H
CMP AL,0DH
JE SECOND_INPUT
SUB AL,48
SHL BL,1
OR  BL,AL
JMP LOOP1


SECOND_INPUT:
PRINTN
PRINT "ENTER NUMBER 2 : "


MOV AH,1            ; TAKING SECOND INPUT
LOOP2:
INT 21H
CMP AL,0DH
JE SUMATION


SUB AL,48
SHL CL,1
OR  CL,AL


JMP LOOP2
SUMATION:           ; CALCULATE THE SUM OF TWO NUMBER
ADD SUM1,BX
```

```
ADD SUM1,CX
ADD SUM2,BX
ADD SUM2,CX
ADD SUM3,BX
ADD SUM3,CX
PRINTN
PRINT "SUM OF TWO BINARY NUMBER : "
SUM:
CMP COUNT,16
JE RIVERSE
SHL SUM1,1
JC ONE
JNC ZER0
ONE:
MOV AH,2
MOV DL,"1"
INT 21H
INC COUNT
JMP SUM
ZER0:
MOV AH,2
MOV DL,"0"
INT 21H
INC COUNT
JMP SUM


RIVERSE1:
PRINTN
PRINT "RIVERSE OF SUM : "     ; HERE REVERSE THE BIT STRING OF SUM
MOV COUNT,0
RIVERSE2:
CMP COUNT,16
JE FINISH
SHR SUM2,1
JC ONE1
JNC ZERO
ONE1:
MOV AH,2
MOV DL,"1"
INT 21H
INC COUNT
JMP RIVERSE2
ZERO:
MOV AH,2
MOV DL,"0"
INT 21H
INC COUNT
JMP RIVERSE2
```

```
FINISH:
PRINTN
PRINT "NUMBER OF 1 : "          ; HERE COUNT THE NUMBER OF 1
MOV CX,16
MOV AL,0


COUNT1:
ROL SUM3,1
JNC NEXT
INC AL


NEXT:
LOOP COUNT1


ADD AL,48
MOV AH,2
MOV DL,AL
INT 21H



EXIT:           ; DOS RETURN
MOV AH,4CH
INT 21H



MAIN ENDP       ; TERMINATE PROGRAM

END MAIN
```

Input & Output:



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.