Index

Mawlana Bhashani Science and Technology University

Santosh, Tangail-1902.

Lab Report

Department of Information and Communication Technology

Course Title: Microprocessor and Assembly Language Lab

Course Code: ICT-3106

Submitted By	Submitted To
Name: **Md. Ashik Mahmud** ID: **IT-18006** Session: 2017-18 3rd Year 1st Semester Dept. of Information & Communication Technology, MBSTU.	**S.M. Shamim** **Lecturer,** Dept. of Information & Communication Technology, MBSTU.

Lab Report-01

1.Write an assembly program to print a character.

Algorithm:

1. Start the program
2. Move the character in dl register
3. Display the character
4. Stop the program

Code:

```
.model small
.stack 100h
.code main
proc    mov
ah,2    mov
dl,65
   int 21h

exit:     mov
ah,4ch
int 21h
main endp
end main
```

Output:

2. Write an assembly program to print a number.

Algorithm:

1.Start the program.
2.Move the number in ‘dl’ register.
3.Display the character.
4.Stop the Program.

Code:

```
.model small
.stack 100h
.code main
proc     mov
ah,2     mov
dl,49    int
21h     exit:
mov
```

```
ah,4ch
int 21h
main endp
end main
```

Output:

3.Write an assembly program to print several characters with new line.

Algorithm:

1.Start the program.
2.Move the Character in ‘dl’ register.
3.Display the character.
4.Display a new line.
5.Again move a character in ‘dl’ register.
6.Display the character.
7.Stop the program.

Code:

```
.model small
.stack 100h
.code main
proc    mov
ah,2    mov
dl,'a'    int
21h

   mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

   mov ah,2
mov dl,'b'
int 21h
        exit:
mov ah,4ch
int 21h
main endp
end main
```

Output:

4.Write an assembly program to print several digits with new line.

Algorithm:

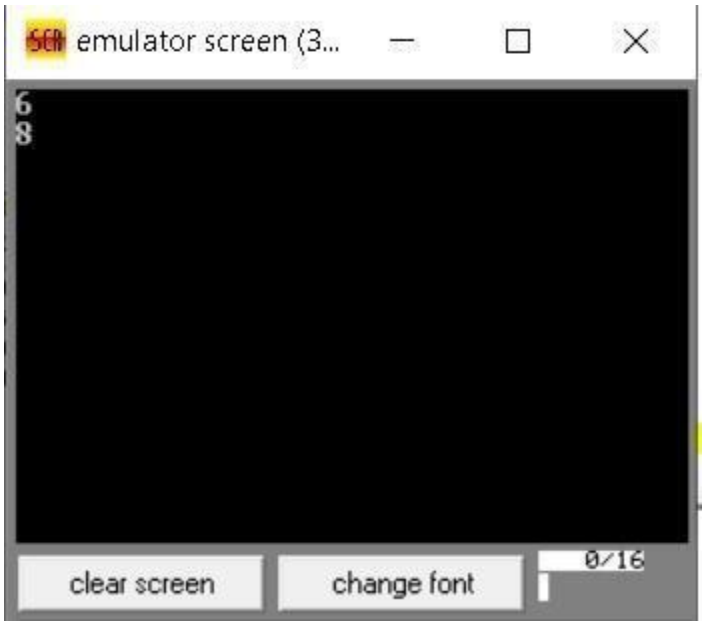
1.Start the program.
2.Move the digit in ‘dl’ register.
3.Display the digit.
4.Display a new line.
5.Again move a digit in ‘dl’ register.
6.Display the digit.
7.Stop the program.

Code:

```
.model small
.stack 100h
.code main
proc    mov
ah,2    mov
```

```
dl,'6'     int
21h

   mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

   mov ah,2
mov dl,'8'
int 21h
        exit:
mov ah,4ch
int 21h
main endp
end main
```

Output:



5.Write an assembly program to enter character or digit and display it on the screen with new line.

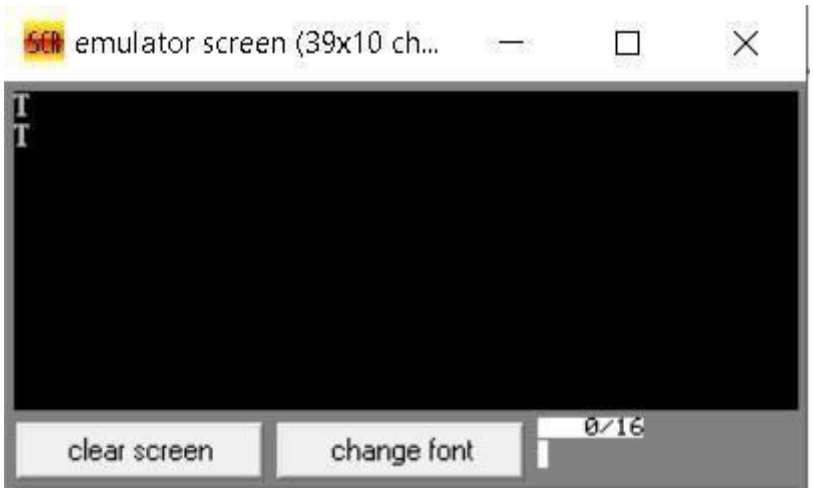
Algorithm:

1.Start the program
2.Enter a character or digit from 'al' register.
3. Move the character or digit in 'bh' register.
4.Display a new line.
5.Move the character or digit to 'dl' register.
6.Display the digit or character.
7.Stop the program.

Code:

```
.model small
.stack 100h
.code main
proc    mov
ah,1
int 21h
mov bh,al
mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

   mov ah,2
mov dl,bh
   int 21h

exit:     mov
ah,4ch    int
21h
main endp
end main
```

Output:



6. Write an assembly program to enter several character or digit and display it on the screen with new line.

Algorithm:

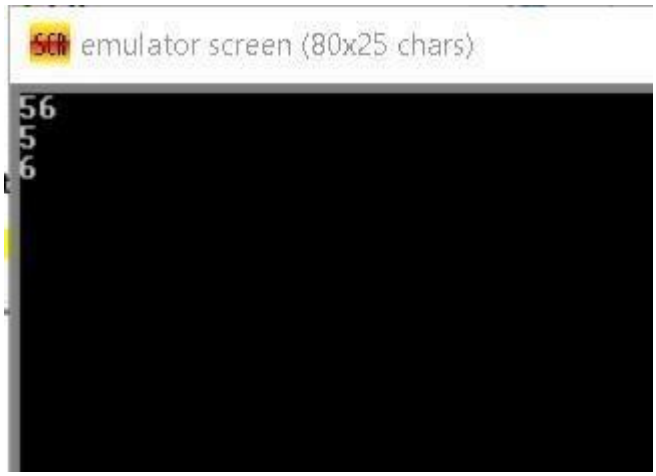
1.Start the program.
2.Enter a character from ‘al’ register.
3.Move the digit or character to ‘bh’ register.
4.Enter another character or digit form ‘al’ register.
5.Move the character or digit to ‘bl’ register.
6.Display a new line.
7.Move the character or digit stored in ‘bh’ register to ‘dl’ register.
8.Display the character or digit.
9.Display a new line.
10. the character or digit stored in ‘bl’ register to ‘dl’ register.
11. Display the character or digit.
12.Stop the program.

Code:

```
.model small
.stack 100h
.code main
proc
mov ah,1
int 21h
mov bh,al
```

```
int  21h
mov bl,al

   mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

   mov ah,2
mov dl,bh
int 21h

   mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

   mov ah,2
mov dl,bl
int 21h

exit:      mov
ah,4ch     int
21h
main endp
end main
```

Output:



7. Write an assembly program to print a character or digit using variable.

Algorithm:

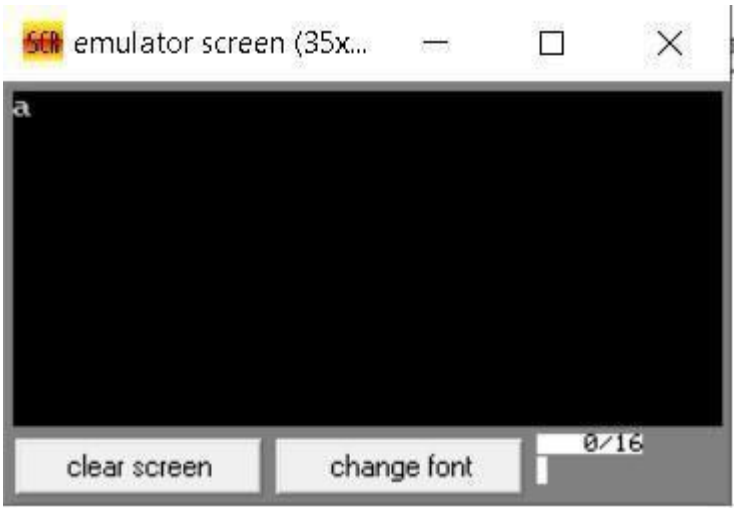
1.Start the Program. 2.Declare a variable.
3.Initialize the variable.
4.Display the variable.
5.Stop the program.

Code:

```
.model small
.stack 100h
.data var db 'a$' .code
main proc     mov
ax,@data
mov ds,ax

   mov ah,2
mov dl,var
int 21h

exit:      mov
ah,4ch     int
21h
```

```
main endp
end main
```

Output:



8. Write an assembly program to print a string.

Algorithm:

1.Start the Program. 2.Declare a variable.
3.Initialize the variable.
4.Display the variable.
5.Stop the program.

Code:

```
.model small   .stack
100h .data  var db
"Hello world $"
.code main proc
mov ax,@data
mov ds,ax

   mov ah,9
lea dx,var
int 21h
exit:     mov
ah,4ch    int
21h
```

```
main endp
end main
```

Output:

9. Write an assembly program to print a string and enter character/digit and display it.

Algorithm:

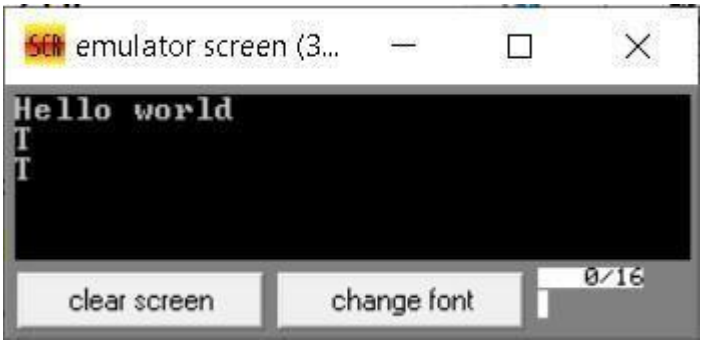
1.Start the Program. 2.Declare a variable.
3.Initialize the variable.
4.Display the variable.
5.Enter a character or digit.
6.Display it.
7.Stop the Program.

Code:

```
.model small .stack
100h .data var db
"Hello world $" .code
main proc mov
ax,@data
mov ds,ax

  mov ah,9
lea dx,var
int 21h
```

```
   mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

   mov ah,1
int 21h
mov bh,al
mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

   mov ah,2
mov dl,bh
int 21h

exit:
   mov ah,4ch
int 21h
main endp
end main
```

Output:



10. Write an assembly program to read first, middle, and last initials of a person's name, and display them in left margin.

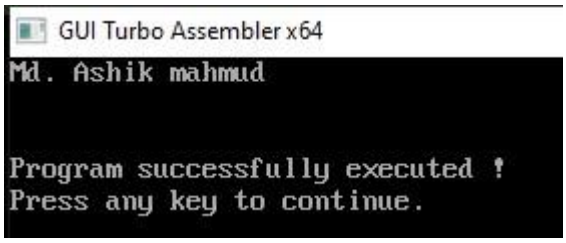
Algorithm:

1.Start the program.
2.Declare three variable.
3.Initialize those three variable.
4.Display three variable.
5.Stop the program.

Code:

```
.model small  .stack
100h .data first db
"Shakib $" middle db
"Al $" last db
"Hasan$"  .code
main proc     mov
ax,@data      mov
ds,ax

   mov    ah,9
lea      dx,first
int  21h    lea
dx,middle int
21h lea
   dx,last   int
   21h
            exit:
mov ah,4ch
int 21h
main endp
end main
```

output:



Lab Report-02

1. Write instructions to do the following.

a. Read a character and display it at the next position on the same line.

b. Read an uppercase letter and display it at the next position on the same line in lowercase.

Algorithm:

1.Start the program.

2.Read a character from ‘al’ register.

3.Move the character to ‘bh’ register.

4.Display character.

5.Read an uppercase letter from ‘al’ register.

6.Move the character to ‘bl’ register.

7.Convert uppercase to lowercase letter.

8.Displsy the lowercase letter.

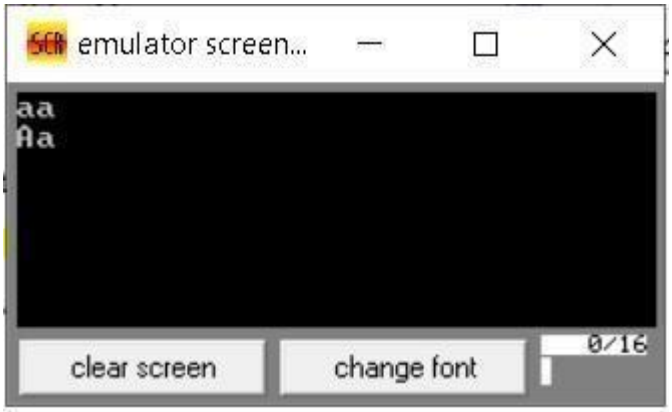
9.Stop the program.

Code:

```
.model small
.stack 100h
.code main
proc     mov
ah,1     int
21h     mov
bh,al

   mov ah,2
mov dl,bh int
21h

   mov ah,2
```

```
   mov dl,10
   int 21h
mov dl,13
int 21h

   mov ah,1
int 21h
mov bl,al

   add bl,32

   mov ah,2
mov dl,bl
int 21h
exit:      mov
ah,4ch     int
21h
main endp
end main
```

Output:



2. Write a program to

a. display a “?”

b. read two decimal digits whose sum is less than 10

c. display them and their sum in the next line with an appropriate message.

Algorithm:

1.Start the program.
2.Display “?”
3.Read two decimal digit from ‘al’ register.
4.Move them to ‘bh’ and ‘bl’ register accordingly.
5.Add those two numbers.
6.Display the sum of those two numbers.
7.Stop the program.

Code:

```
.model small
.stack 100h
.data  qus
db "?$"   sum db 10,13,"Sum of two
number is: $"
.code main proc
mov ax,@data
mov ds,ax

   mov ah,2
mov dl,qus
int 21h

   mov    ah,2
mov dl,10 int
21h        mov
dl,13  int
   21h

   mov ah,1
int 21h
mov bh,al
int 21h
mov bl,al

   add bh,bl
```

```
   sub bh,48

   mov ah,9
lea dx,sum
int 21h
mov ah,2
mov dl,bh
int 21h

exit:
   mov ah,4ch
int 21h
main endp
end main
```

Output:

3. write a program to

a. prompt the user
b. Read first middle and last initials of a person's name
c. And display them down the left margin

Algorithm:

1.Start the program.
2.Declare three variable.
3.Initialize those three variable.

4.Display three variable.
5.Stop the program.

Code:

```
.model small  .stack
100h .data first db
"Shakib $" middle db
10,13,"Al $" last db
10,13"Hasan$"  .code
main proc     mov
ax,@data
mov ds,ax

   mov ah,9
lea dx,first     int
21h     lea
dx,middle     int
21h     lea
dx,last      int
21h

exit:      mov
ah,4ch
int 21h
main endp
end main
```

output:

4. Write an assembly program to enter one of the hex digits A-F, and display it on the next line in decimal.

Algorithm:

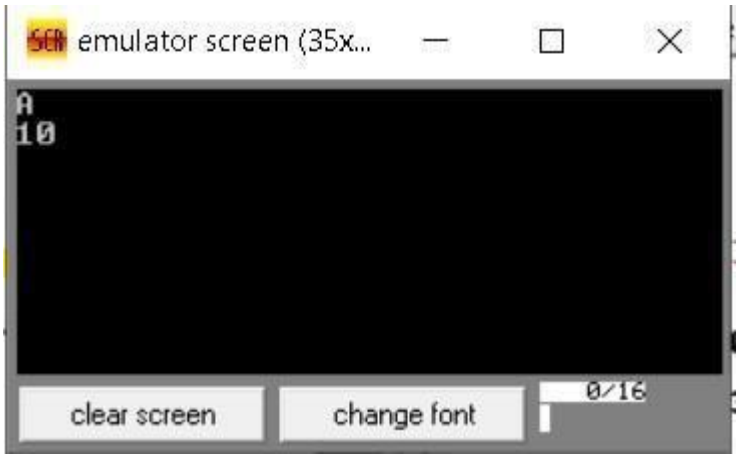
1.Start the program.
2.Read a hexadecimal number from A-F.
3.Find the decimal value for the corresponding hexadecimal number.
4.Display the decimal value.
5.Stop the program.

Code:

```
.model small
.stack 100h
.code main
proc     mov
ah,1     int
21h     mov
bh,al

   mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

   mov ah,2
mov dl,'1'
int 21h     sub
bh,17     mov
dl,bh     int
21h

exit:
   mov ah,4ch
int 21h
```

```
main endp
end main
```

Output:



5. Write an assembly program to display asterisks (**********) ten times with new line.

Algorithm:

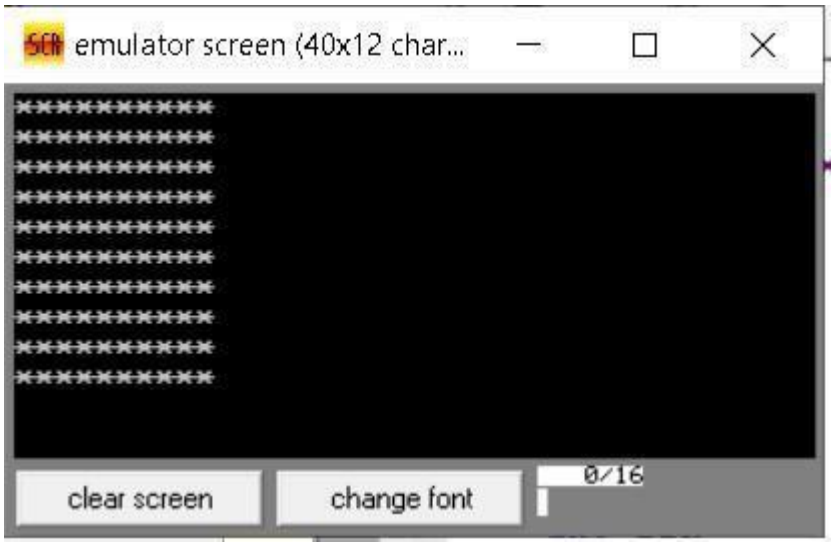
1.Start the program.
2.Display '*'.
3.Display new line.
4.Repeat step 2 and 3 for 9 times more.
5.Stop the program.

Code:

```
.model small
.stack 100h
.data
star db "**********",10,13,"$"
.code main proc     mov
ax,@data      mov ds,ax

   mov ah,9
lea dx,star
int 21h     int
21h     int
21h     int
```

```
21h     int
21h     int
21h     int
21h     int
21h     int
21h     int
21h
exit:       mov
ah,4ch
int 21h
main endp
end main
```

Output:



6. Write an assembly program to display to (a) display"?", (b) read three initials,(a,b,c) display them in the middle of an 11 x 11 box of asterisk.

Algorithm:

1.Start the program.
2.Enter three values to bl,bh,cl register.
3.Display 11 asterisk in every first five lines.
4. Then print bl,bh,cl register value in the 5,6,7th position in 6th line.

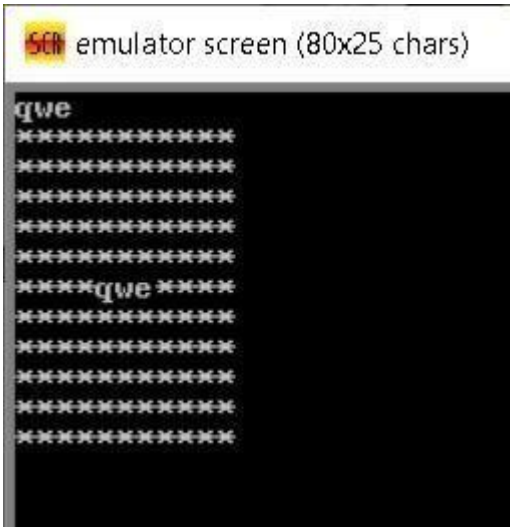
5.Then display 11 asterisk in every last five lines.
6.Stop the program.

Code:

```
.model small
.stack 100h
.data
ast db 10,13,"***********$"
ast2  db  "****$"  .code  main
proc     mov ax,@data     mov
ds,ax

   mov ah,1
int 21h
mov bl,al
int 21h
mov bh,al
int 21h
mov cl,al

   mov ah,9
lea dx,ast
int 21h     int
21h     int
21h     int
21h     int
21h

   mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h
```

```
   mov ah,9
lea dx,ast2
int 21h
mov ah,2
mov dl,bl
int 21h
mov dl,bh
int 21h
mov dl,cl
int 21h

   mov ah,9
lea dx,ast2
int 21h

   mov ah,9
lea dx,ast
int 21h     int
21h     int
21h     int
21h
```

Output:



Lab report - 03

1.Write an assembly program to display different triangle using asterisk and digit.

2.Write an assembly program to enter two 8 bit numbers and print their sum which is less than 9.

Algorithm:

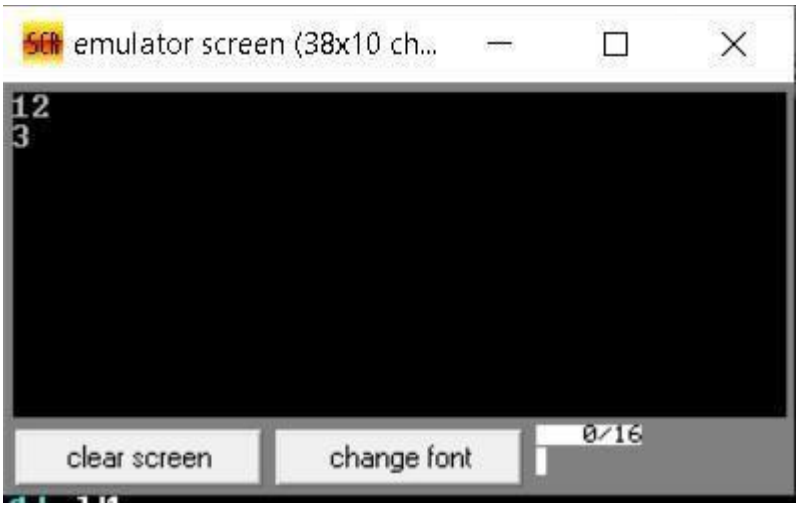
1.Start the program.
2.Enter two numbers from ‘al’ register.
3.Move those two numbers to ‘bh’,’bl’ register accordingly.
4.Add ‘bh’ & ‘bl’ and store the reault in ‘bh’ register.
5.Sub 48 from ‘bh’ register.
6.Display ‘bh’ register.
7.Stop the program.

Code:

```
.model small
.stack 100h
.code main
proc
mov ah,1
int 21h
mov bh,al

   mov ah,1
int 21h
mov bl,al

   mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h
```

```
   add bh,bl
sub bh,48

   mov ah,2
mov dl,bh
int 21h
exit:     mov
ah,4ch
int 21h
main endp
end main
```

Output:



9.Write an assembly program to enter two 8 bit numbers and print their sum which is larger than.

Algorithm:

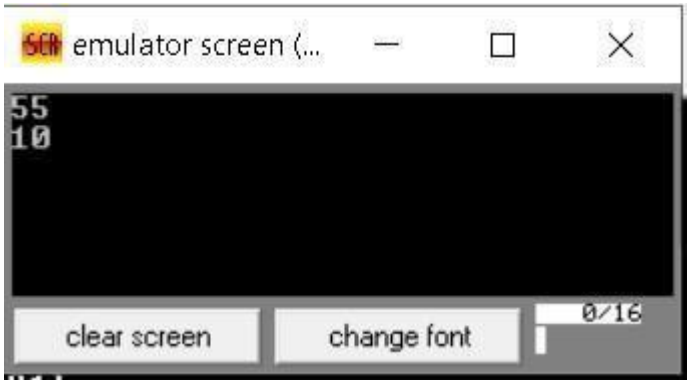
1.Start the program.
2.Enter to number from ‘al’ register,
3.Move those two numbers to ‘bh’ and ‘bl’ register accordingly.
4.Add them and sub 58 from ‘bh’ register and store the result to ‘bh’ register.
5.Display 1 first and then ‘bh’ .
6.Stop the program.

Code:

```
.model small
```

```
.stack 100h
.code main
proc     mov
ah,1     int
21h     mov
bh,al     mov
ah,1     int
21h     mov
bl,al

   mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

   add bh,bl
sub bh,58

   mov ah,2
mov dl,'1'
int 21h
mov dl,bh
int 21h

exit:      mov
ah,4ch      int
21h
main endp
end main
```

Output:



4. Write an assembly program to enter a number and perform multiplication with itself which less than 9.

Algorithm:

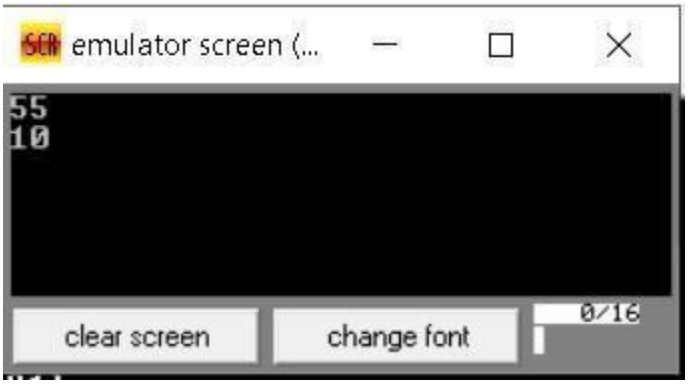
1.Start a program.
2.Enter first number in ‘bl’ register.
3.Enter second number from ‘al’ register and multiply it with ‘bl’ register.
4.Move the value in bl register,
5.Add 48 with bl register.
6.Display it.

Code:

```
.model small
.stack 100h
.data .code
main proc
mov ah,1
int 21h
mov bl,al
sub bl,48

   mov ah,1
int 21h    sub
al,48

   mul bl
mov bl,al
   add bl,48
```

```
   mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

   mov ah,2
mov dl,bl
int 21h
        exit:
mov ah,4ch
int 21h
main endp
end main
```

Output:



5.Write an assembly program to enter a number and perform multiplication with itself which larger than 9.

6. Write an assembly program to enter two numbers and perform division.

Algorithm:

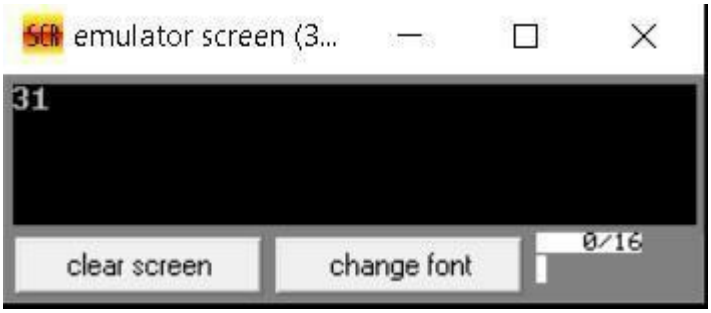
1.Start the program.
2.Enter two numbers.
3.Move them to ‘bl’ and ‘al’ register accordingly.
4.Divide ‘al’ register by ‘bl’ register.

5.Display bl register and bh register.
6.stop the program.

Code:

```
.model small
.stack 100h
.data
  .code
main
proc
mov al,7
mov bl,2

div bl
mov bx,ax
mov ah,2
mov dl,bl
add dl,48
int 21h
mov dl,bh
add dl,48
int 21h
        exit:
mov ah,4ch
int 21h
main endp
end main
```

Output:



Lab report - 04

1.Write an assembly program to find larger number between two numbers.

Algorithm:

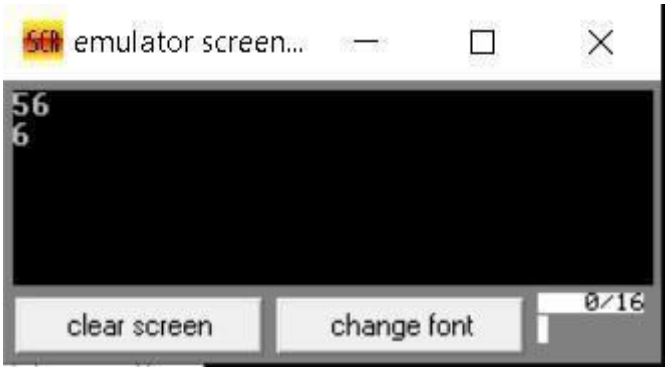
1.Start the program.
2.Enter two numbers in ‘bl’ and ‘bh’ register from ‘al’ register.
3.compare two number.
4.If ‘bl’ is greater jump to l2 else jump l1.And Display the greater number.
5.Stop the program.

Code:

```
.model small
.stack 100h
.data .code
main proc
mov ah,1
int 21h
mov bl,al
int 21h
mov bh,al

   mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

   cmp bl,bh
jg l1     jmp
l2          l2:
mov ah,2
mov dl,bh
int 21h     jmp
exit          l1:
```

```
mov ah,2
mov dl,bl
int 21h     jmp
exit        exit:
mov ah,4ch
int 21h
main endp
end main
```

Output:



02. Write an assembly program to find small number between two numbers.

Algorithm:

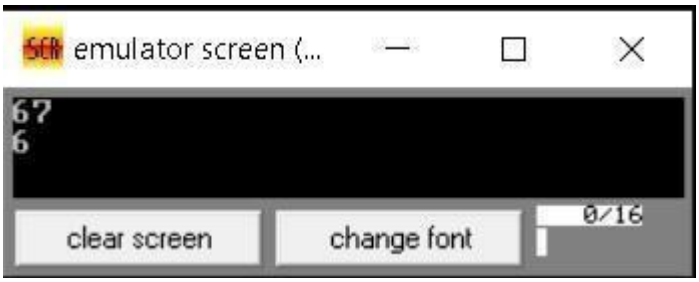
1.Start the program.
2.Enter two numbers in ‘bl’ and ‘bh’ register from ‘al’ register.
3.compare two number.
4.If ‘bl’ is small jump to l2 else jump l1.And Display the smaller number.
5.Stop the program

Code:

```
.model small
.stack 100h
.data

.code main
proc     mov
ah,1     int
21h     mov
```

```
bl,al     int
21h     mov
bh,al  mov
ah,2 mov
dl,10 int
21h mov
dl,13
int 21h

   cmp
bl,bh       jl
l1       jmp l2

l2:
   mov ah,2
mov dl,bh
int 21h     jmp
exit

l1:
   mov ah,2
mov dl,bl
int 21h
jmp exit

exit:       mov
ah,4ch      int
21h
main endp
end main
```

Output:



03.Write an assembly program to enter value of Al.If Al contains a negative number, put -1 In Bl;if Al contains 0, put O In Bl;if Al contains a positive number, put 1 In Bl.·

Algorithm:

1.Start the program.

2.Enter a number to 'bl' register from 'al' register.

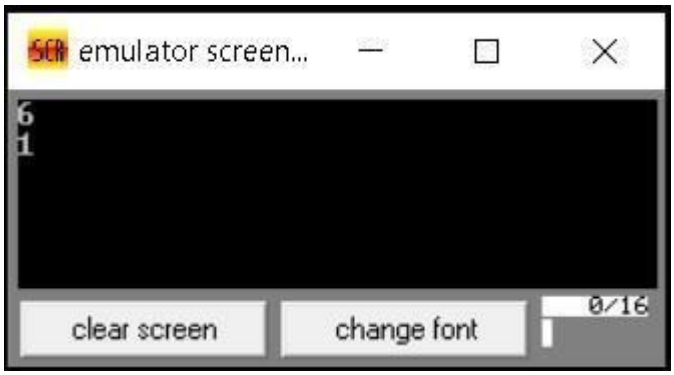
3.Compare the number with 0.if it is greater than 0 jump to level 2 and print 1.If it is less than 0 jump to level 2 and print -1.If it is equal to 0 than jump to level 3 and print 0.

Code:

```
.model small
.stack 100h
.data

.code main
proc
mov ah,1
int 21h
mov bl,al

   mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

   cmp bl,0
jl l1     jg l2
```

```
je l3

l2:
mov
ah,2 mov
dl,"1
" int
21h jmp
exit
        l1:
mov ah,9
mov dl,'-'
int 21h
mov dl,'1'
int 21h
jmp exit
l3:
mov ah,2
mov dl,"0"
int 21h     jmp
exit

exit:       mov
ah,4ch      int
21h
main endp
end main
```

Output:



04. Write an assembly program to enter value of Al;If AL contains 1 or 3, display "o"; if AL contains 2 or 4, display "e".

Algorithm:

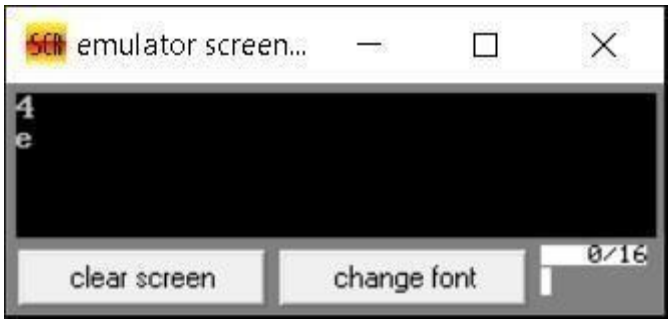
1.Start the program.
2.Enter a number to bl register.
3.Compare this number with 1 and 3 . If it is equal to 1 or 3 jump to level and print ‘o’ else jump level 2 and print ‘e’
4.Stop the program.

Code:

```
.model small
.stack 100h
.data

.code main
proc
mov ah,1
int 21h
mov bl,al

   mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h
```

```
   cmp bl,49
je l1     jmp  l2
cmp
bl,51
   je l1      jmp
l2
      l1:
   mov ah,9
   mov dl,'o'
   int 21h
   jmp exit
   l2:
   mov ah,2
mov dl,"e"
int 21h     jmp
exit     exit:
mov ah,4ch
int 21h
main endp
end main
```

Output:



05. Write an assembly program to enter a character;if it's an uppercase letter, display it.Otherwise terminate.

Algorithm:

1.Start the program.

2.Take input in ‘bl’ register.

3.Compare the character whether it is between ‘A’-‘Z’.If yes then print it. Otherwise terminate.
4.Stop the program.

Code:

```
.model small
.stack 100h
.data

.code main
proc
mov ah,1
int 21h
mov bl,al

   mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

   cmp
bl,65     jge
l1      jmp
exit

l1:
cmp
bl,90
jle l2
   jmp exit
       l2:
mov ah,2
mov dl,bl
```

```
int 21h     jmp
exit

exit:
   mov
   ah,4ch int
   21h main
   endp
end main
```

Output:

06. Write an assembly program to enter a character;if it's y or Y, display it.Otherwise terminate.

Algorithm:

1.Start the program.
2.Enter a character in bl register.
3.Compare bl register with ‘y’ or ‘Y’.if yes then print it otherwise terminate the program.

Code:

```
.model small
.stack 100h
.data

.code main
proc     mov
ah,1     int
```

```
21h     mov
bl,al

   mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

   cmp bl,89
   je l1
jmp l2
        l1:
mov ah,2
mov dl,bl
int 21h
jmp exit
        l2:
cmp bl,121
je l1     jmp
exit
exit:     mov
ah,4ch     int
21h    main
endp end
main
```

output:

Lab report -05

01. Write an assembly count-controlled loop program to display a row of 80 stars.

Algorithm:

1.Start the program.
2.Initialize ‘cx’ register with the value 80.
3.Create a level named l1.then loop the level and print ‘*’
4.Stop the program.

Code:

```
.model small
.stack 100h
.data

.code  main
proc     mov
cx,80
mov     ah,2
mov    dl,'*'
l1:
   int 21h
loop l1
exit:      mov
ah,4ch     int
21h
main endp
end main
```

Output:

02. Write an assembly program to print the following series (for)9 8 7 6 5 4 3 2 1.

Algorithm:

1.Start the program.

2. Initialize ‘cx’ register with the value 9.

3.create a level named l1,print 57,decrement the value of ‘dl’ register.Loop the level.

4.Stop the program.

Code:

```
.model small
.stack 100h
.code main
proc     mov
cx,9     mov
ah,2     mov
dl,57     l1:
int 21h
dec dl
   loop l1

exit:       mov
ah,4ch      int
21h
main endp
end main
```

Output:

03. Write an assembly program to print the following series (for)9 7 5 3 1.

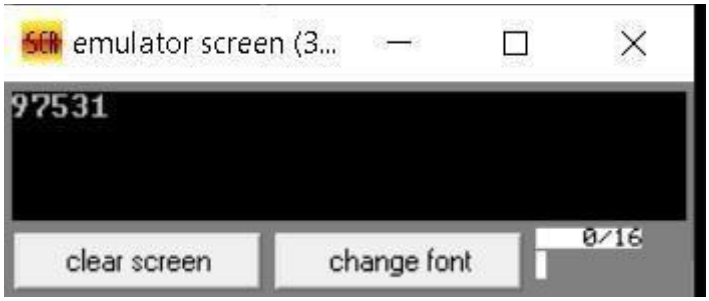
Algorithm:

1.Start the program.
2. Initialize ‘cx’ register with the value 5.
3.create a level named l1,print 57,decrement the value of ‘dl’ register by 2.Loop the level.
4.Stop the program.

Code:

```
.model small
.stack 100h
.data

.code main
proc    mov
cx,5    mov
ah,2    mov
dl,57    l1:
int 21h
dec dl
dec dl
loop l1

exit:     mov
ah,4ch     int
21h
main endp
end main
```

Output:



04. Write an assembly program to print the following series (for)1 2 3 4 5 6 7 8 9.

Algorithm:

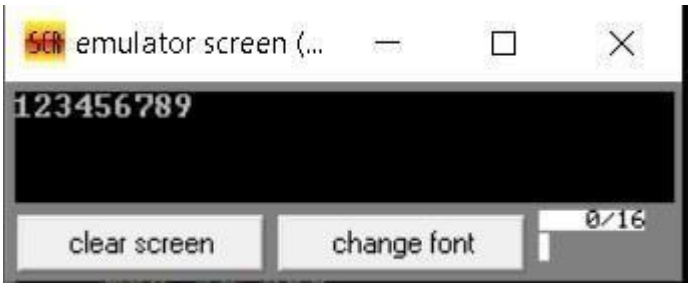
1.Start the program.
2. Initialize ‘cx’ register with the value 9.
3.create a level named l1,print 49,increment the value of ‘dl’ register.Loop the level.
4.Stop the program.

Code:

```
.model small
.stack 100h
.data

.code main
proc
mov cx,9
mov ah,2
mov dl,49
l1:     int
21h      inc
dl      loop
l1

exit:
   mov ah,4ch
int 21h
```

main endp
end main

Output:



05.Write an assembly program to print the following series (for)8 6 4 2.

Algorithm:

1.Start the program.
2. Initialize ‘cx’ register with the value 4.
3.create a level named l1,print 56,decrement the value of ‘dl’ register.Loop the level.
4.Stop the program.

Code:

```
.model small
.stack 100h
.data

.code main
proc     mov
cx,4     mov
ah,2     mov
dl,56     l1:
int 21h      dec
dl     dec dl
loop l1     exit:
mov ah,4ch
int 21h
main endp
end main
```

Output:

06. Write an assembly program to print the following series (while) 9 8 7 6 5 4 3 2 1.

Algorithm:

1.Start the program.
2. Initialize ‘dl’ register with the value 57.
3.create a level named while_,print 57,decrement the value of ‘dl’ register. Compare the value of ‘dl’ register with the value 49.If ‘dl’ register’s value is less then 49 then jump to exit level otherwise jump to while_ level.
4.Stop the program

Code:

```
.model small
.stack 100h
.data

.code   main
proc    mov
ah,2    mov
dl,57
while_:    int
21h     dec dl
cmp dl,49
jge while_
jmp exit
```

```
exit:      mov
ah,4ch     int
21h
main endp
end main
```

Output:

Lab report -06

01.Write a program in assembly language to check whether a number is even or odd.

Algorithm:

1.Start the program.
2.Take one input.
3.check whether it is even or odd.
4.If even print “Even” otherwise print “Odd”.
5.Stop the program.

Code:

```
.model small
.stack 100h
.data even
db
'Even$' odde  db 'Odd$'
          .code
main proc
   mov ax,@data
mov ds, ax
      mov
ah,1
int 21h
   mov bl,al
   test bl,01h
jne odd

mov ah, 9
lea dx,even     int 21h
jmp exit

odd:     mov
ah,9     lea
```

```
dx,odde     int
21h
    exit:
mov ah,4ch
int 21h
main endp
end main
```

Output:

02.Write a program in assembly language to load a byte in memory location 8000H and increment the contents of the memory location.

Code:

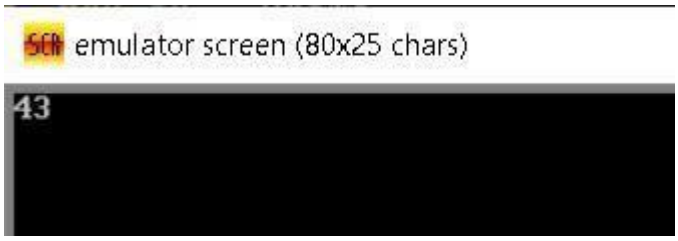
```
DATA SEGMENT NUM1
DB 7H NUM2
DB ?
ENDS
CODE SEGMENT ASSUME DS:DATA
CS:CODE START:
MOV AX,DATA
MOV DS,AX
MOV AL,NUM1
MOV [8000H],AL
INC [8000H]
MOV AL,[8000H]
MOV NUM2,AL
MOV AH,4CH
INT 21H
ENDS
END START
```

3. Write a program in assembly language to swap two numbers.

Code:

```
.MODEL SMALL
.STACK 100H
.DATA
  NUM1 DB
'3'
  NUM2 DB
'4'
.CODE
  MOV AX , @DATA
  MOV DS , AX

  MOV BL , NUM1
  MOV CL , NUM2

  MOV NUM2 , BL
  MOV NUM1 , CL

  MOV AH,2
  MOV DL,NUM1
  INT 21H
  MOV DL,NUM2
  INT 21H
EXIT:
  MOV AH , 4CH

INT 21H END
```

Output:



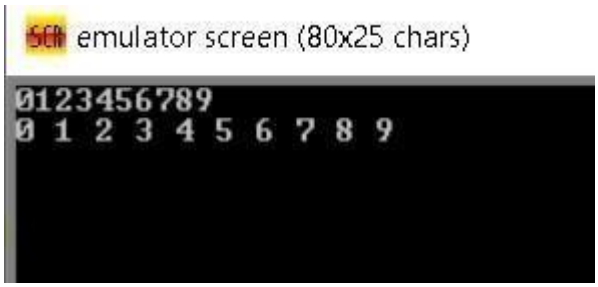
04. Write Assembly program to read ten (10) characters from console.

Code:

```
.model small
.stack        100h
.data   arr db 10
dup(?)      .code
main          proc
mov   ax,@data
mov ds,ax

   mov cx,10      mov
si,offset arr     loop1:
mov ah,1
int 21h
mov
[si],al     inc si
loop loop1

   mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

   mov si,offset arr
mov cx,10

   loop2:
mov dl,[si]
```

```
mov ah,2
int 21h

   mov dl,32
mov ah,2
int 21h

inc si
   loop loop2
```

Output:



05. Write an Assembly program to read in two decimal inputs and print out the smaller of the two, in decimal.

Algorithm:

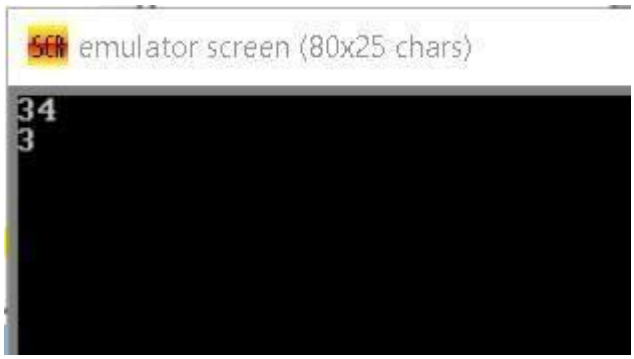
1.Start the program.
2.Enter two numbers in ‘bl’ and ‘bh’ register from ‘al’ register.
3.compare two number.
4.If ‘bl’ is small jump to l2 else jump l1.And Display the smaller number.
5.Stop the program

Code:

```
.model small
.stack 100h
.data
.code main
proc     mov
ah,1
int 21h
mov
```

```
bl,al     int
21h     mov
bh,al

   mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

   cmp bl,bh
   jl l1
jmp l2
l2:
mov ah,2
mov dl,bh
int 21h
jmp exit
l1:
mov ah,2
mov dl,bl
int 21h
   jmp exit

exit:
   mov ah,4ch      int 21h
main      endp      end      main
```

Output:



06. Write an Assembly program to calculate the average of three given numbers stored in memory.

Algorithm:

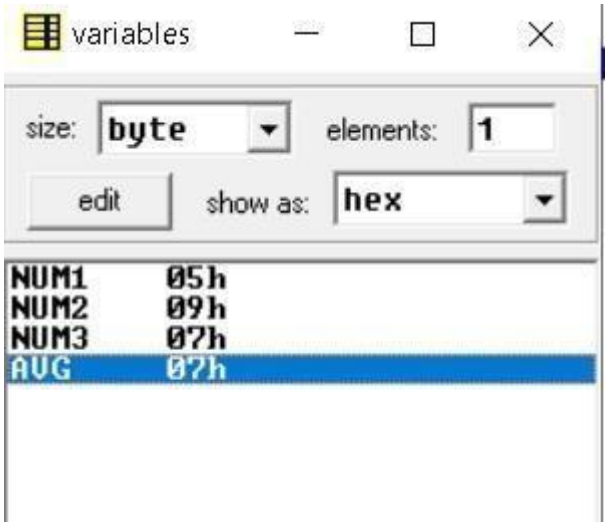
1.Start the program.
2.Define three variables.
3.Initialize those variables.
4.Move num1 to al register.add num2 and num3 to al register.
5.set the value of ah register value as 0
6.Set the value of dl register as 3.
7.perform div operation.
8.Stop the program.

Code:

```
.model small
.stack 100h
.data num1
db 5 num2
db 9 num3
db 7 avg db
? .code main
proc

   mov ax,@data
mov ds,ax

   mov al,num1
add al,num2     add
```

```
al,num3     mov  ah,0
mov dl,3      div dl

   mov avg,al

exit:      mov
ah,4ch     int
21h
main endp
end main
```

Output:



07. Write an Assembly program in which a procedure converts Hexadecimal value to print its Decimal form on Screen.

Algorithm:

1.start the program.
2.Enter a hex digit.
3.Compare the digit .if it is greater than 9 then jump to hex level else jump to num level.
4.In num level just print the number.
5.in hex level print the decimal value of the hex digit.
6.Stop the program.

Code:

```
.model small
```

```
.stack 100h .data msg1 db
10,13,'ENTER A HEX DIGIT:$' msg2 db
10,13,'IN DECIMAL IS IT:$'  msg4 db 10,13,'ILLEGAL
CHARACTER- ENTER 0-9 OR A-F:$'
   .code

again:

   mov ax,@data
mov ds,ax    lea dx,msg1
mov
ah,9
int 21h

   mov ah,1
int 21h
mov bl,al

   jmp go

go:

   cmp bl,'9'
ja hex    jb
num    je num
hex:

    cmp bl,'F'
ja illegal

   lea dx,msg2
mov ah,9
int 21h
```

```
mov dl,49d
mov ah,2
int 21h

   sub bl,17d
mov dl,bl
mov ah,2
int 21h

   jmp exit

 num:

   cmp bl,'0'
jb illegal

   lea dx,msg2
mov ah,9
int 21h

   mov dl,bl
mov ah,2
int 21h

   jmp exit

 illegal:

      lea dx,msg4
mov ah,9
      int 21h
```

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

    jmp go

exit: end
```

Output:

08. Write an Assembly program to convert Centigrade (Celsius) to Fahrenheit temperature measuring scales.

Algorithm:

1.Start the program.
2.Enter a value to al register and sub 30h from this.
3.Store 0 to ah register and 10 to bl register.
4.Multiply bl register with al register.
5.Move the value of al register to bl register.
6.Move al register value to T.
7.Store 9 to dl register.
8.Multiply dl register with al register and divide with 5.
9.Display the value.
10.Stop the program.

Code:

```
DATA SEGMENT
T    DB ?
RES  DB 10 DUP ('$')
MSG1 DB "ENTER TEMPERATURE IN CELSIUS (ONLY IN 2 DIGITS) : $"
MSG2 DB 10,13,"CONVERTED IS FAHRENHEIT (TEMPERATURE) : $"
```

```
DATA ENDS
CODE SEGMENT ASSUME DS:DATA,CS:CODE
START:
MOV AX,DATA
MOV DS,AX
LEA DX,MSG1
MOV AH,9
INT 21H
MOV AH,1
INT 21H
SUB AL,30H
MOV AH,0
MOV BL,10
MUL BL
MOV BL,AL
MOV AH,1
INT 21H
SUB AL,30H
MOV AH,0
ADD AL,BL
MOV T,AL
MOV DL,9
MUL DL
MOV BL,5
DIV BL
MOV AH,0
ADD AL,32
LEA SI,RES
CALL HEX2DEC
LEA DX,MSG2
MOV AH,9
INT 21H
LEA DX,RES
MOV AH,9
```

```
INT 21H
MOV AH,4CH
INT 21H
CODE ENDS
HEX2DEC PROC NEAR
MOV CX,0
MOV BX,10
LOOP1: MOV DX,0
DIV BX
ADD DL,30H
PUSH DX
INC CX
CMP AX,9
JG LOOP1
ADD AL,30H
MOV [SI],AL
LOOP2: POP AX
INC SI
MOV [SI],AL
LOOP LOOP2
RET
HEX2DEC ENDP  END
START
```

Output: