Index

Mawlana Bhashani Science and Technology University

Santosh, Tangail-1902.

Lab Report

Department of Information and Communication Technology

Report No: 01

Report Name: Assembly language Program.

Course Title: Microprocessor and Assembly Language Lab

Course Code: ICT-3106

Submitted By	Submitted To
Name: **Zafrul Hasan Khan** ID: **IT-18003** Session: 2017-18 3rd Year 1st Semester Dept. of Information & Communication Technology, MBSTU.	**S.M. Shamim** **Lecturer,** Dept. of Information & Communication Technology, MBSTU.

Program 1: Write an assembly program to print a character.

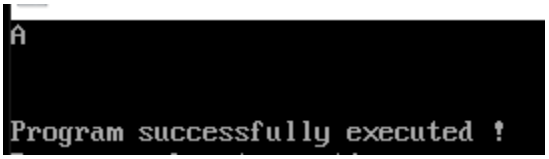
Algorithms:

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

Source Code:

```
.MODEL SMALL
.STACK 100H
.CODE
MAIN PROC
MOV AH,2
MOV DL,”A”
INT 21H
MOV AH,4CH
INT 21H
MAIN ENDP
End main
```

Ouput:



Program 2: Write an assembly program to print a number.

Algorithms:

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

Source Code:

```
.MODEL SMALL
.STACK 100H
.CODE
MAIN PROC
MOV AH,2
MOV DL,"1"
INT 21H
```

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

Output:

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

Algorithms:

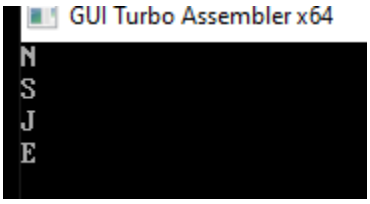
1.Start the program.
2.Move the Character in ‘dl’ register.
3.Display the character.
4.Display a new line.
5.Again move a character in ‘dl’ register.
6.Display the character.
7.Stop the program.

Source Code:

```
.MODEL SMALL
 .STACK 100H
 .CODE
   MAIN PROC
   MOV AH,2
   MOV DL,"B"
   INT 21H
   MOV AH, 2
     MOV DL, 10
     INT 21H
     MOV DL,13
     INT 21H

     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
```

```
MOV AH, 2
    MOV DL, 10
    INT 21H
    MOV DL,13
    INT 21H
      MOV AH,2
    MOV DL,"A
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.

Algorithms:

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

Source Code:

```
.MODEL SMALL
 .STACK 100H
 .CODE
   MAIN PROC
   MOV AH,2
   MOV DL,"5 "
   INT 21H
MOV AH, 2
    MOV DL, 10
    INT 21H
    MOV DL,13
    INT 21H
    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,"7"
    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.

Algorithms:

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.

Source Code:

```
 .MODEL SMALL
.STACK 100H
.CODE
```

```
MAIN PROC
  MOV AH, 1
  INT 21H
  MOV BL, AL
  MOV AH, 2
  MOV DL, 10
  INT 21H
  MOV DL,13
  INT 21H
  MOV AH, 2
  MOV DL, BL
  INT 21H
  MOV AH, 2
  MOV DL, 10
  INT 21H
  MOV DL,13
  INT 21H
  EXIT:
  MOV AH, 4CH
 INT 21H

 MAIN ENDP
END MAIN
```

Input : Y

Output:

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

Algorithms:

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.

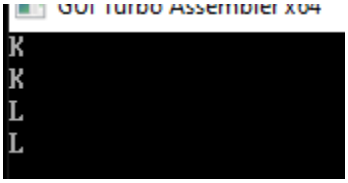
10. the character or digit stored in ‘bl’ register to ‘dl’ register.
11. Display the character or digit.
12.Stop the program.

Source Code:

```
.MODEL SMALL
 .STACK 100H
 .CODE
   MAIN PROC
     MOV AH, 1
     INT 21H
     MOV BL, AL
     MOV AH, 2
     MOV DL, 10
     INT 21H
     MOV DL,13
     INT 21H
MOV AH, 2
     MOV DL, BL
     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
```

Input:_K , L

Output :



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

Algorithms:

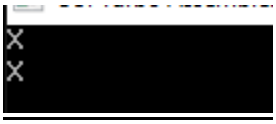
1.Start the Program.
2.Declare a variable.
3.Initialize the variable.
4.Display the variable.
5.Stop the program

Source code

```
.MODEL SMALL
 .STACK 100H
 .DATA
   VALUE_1   DB  ?
   .CODE
  MAIN PROC
    MOV AX, @DATA
    MOV DS, AX
    MOV AH,1
INT 21H
    MOV VALUE_1,AL

    MOV AH, 2
    MOV DL, 10
    INT 21H
    MOV DL,13
    INT 21H
  MOV AH,2
    MOV DL,VALUE_1
    INT 21H
    EXIT:
    MOV AH, 4CH
    INT 21H
  MAIN ENDP
 END MAIN
```

Input : X

Output :



8. Write an assembly program to print a string.

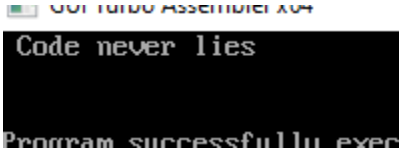
Algorithms:

1. Create a string
2. Load the effective address of the string in dx using LEA command
3. Print the sting by calling the interrupt with 9H in AH
4. The string must be terminated by '$' sign

Source Code:

```
.MODEL SMALL
.STACK 100H
.DATA
 STRING DB ' Code never lies', '$'
.CODE
MAIN PROC FAR
 MOV AX,@DATA
 MOV DS,AX
 LEA DX,STRING
 MOV AH,09H
 INT 21H
 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.

Algorithms:

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

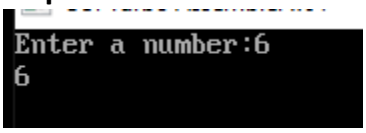
6.Display it.
7.Stop the Program.

Source Code:

```
.MODEL SMALL
.STACK 100H
.DATA
 STRING DB 'Enter a number:', '$'
.CODE
MAIN PROC FAR
 MOV AX,@DATA
 MOV DS,AX
 LEA DX,STRING
 MOV AH,09H
 INT 21H
 MOV AH, 1
    INT 21H
    MOV BL, AL
    MOV AH, 2
    MOV DL, 10
 INT 21H
MOV DL,13
    INT 21H
MOV AH, 2
    MOV DL, BL
    INT 21H
 MOV AH,4CH
 INT 21H
MAIN ENDP
END MAIN
```

Input: 6

Ouput :



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

Algorithms:

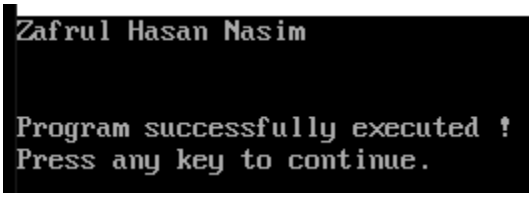
1.Start the program.
2.Declare three variable.
3.Initialize those three variable.
4.Display three variable.

5.Stop the program.

Source Code:

```
.model small
.stack 100h
.data
first db "Sajidur $"
middle db "Rahman $"
last db "Sajid$"
.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 :



Mawlana Bhashani Science and Technology University

Santosh, Tangail-1902.

Lab Report

Department of Information and Communication Technology

Report No: 02

Report Name: Assembly language Program.

Course Title: Microprocessor and Assembly Language Lab

Course Code: ICT-3106

Submitted By	Submitted To
Name: **Zafrul Hasan Khan** ID: **IT-18003** Session: 2017-18 3rd Year 1st Semester Dept. of Information & Communication Technology, MBSTU.	**S.M. Shamim** **Lecturer,** Dept. of Information & Communication Technology, MBSTU.

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

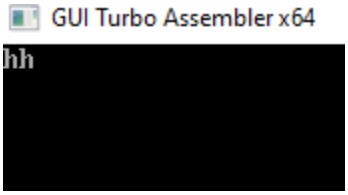
Algorithms :

1.Start the program.
2.Read a Character .
4.Display the character.
5.Stop the program.

Source Code :

```
.model small
.stack 100h
.data
.code
main proc

 mov ah,1
 int 21h

 mov ah,2
 mov dl,al ;charcter print
 int 21h

 main endp
 end main
```

Input & Output:



1(b) Read an lowercase letter and display it at the next position on the same line in upper case.

Algorithms :

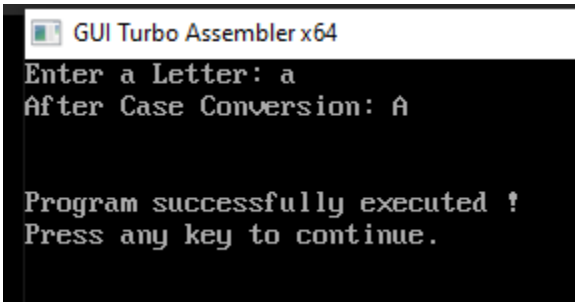
1.Start the program.
2.Read a lowercase letter.
3.add 32 .
4. Then added ascii number find uppercase
5.Stop the program.

```
.MODEL SMALL
.STACK 100H
.DATA
MSG1 DB 'Enter a Letter: $'
MSG2 DB 'After Case Conversion: $'
```

```
.CODE

MAIN PROC
   MOV AX,@DATA    ;DATA SEGMENT
   MOV DS,AX

   LEA DX,MSG1
   MOV AH,9       ;MSG1
   INT 21H

   MOV AH,1
   INT 21H       ;INPUT
   MOV BL,AL

   MOV AH,2
   MOV DL,0AH
   INT 21H         ;NEW LINE
   MOV DL,0DH
   INT 21H

   LEA DX,MSG2
   MOV AH,9        ;MSG2
   INT 21H

   CMP BL,97
   JGE L1

   ADD BL,32
   MOV AH,2        ;UPPER TO LOWER
   MOV DL,BL
   INT 21H
   JMP EXIT

   L1:
   SUB BL,32
   MOV AH,2        ;LOWER TO UPPER
   MOV DL,BL
   INT 21H

   JMP EXIT

   EXIT:
   MOV AH,4CH       ;EXIT
   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.

Algorithms :

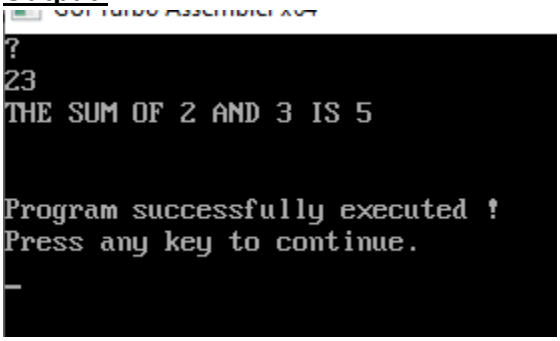
1.Start the program.
2.Firstly display '?'.
3.Read two decimel number .
4. Then sum these number and add new line
5.Stop the program.

Source Code :

```
.MODEL SMALL
.STACK 100H
.DATA
   STR1 DB 0AH,0DH,'THE SUM OF '
   FIRSTNUM DB ?
   STR2 DB ' AND '
   SECONDNUM DB ?
   STR3 DB ' IS '
   ANS DB ?
   STR4 DB ' $'
.CODE
MAIN PROC

   MOV AX,@DATA
   MOV DS,AX

   MOV AH,2
   MOV DL,3FH
   INT 21H


   MOV AH,2
   MOV DL,0AH
```

```
    INT 21H
    MOV DL,0DH
    INT 21H

    INT 21H

    MOV AH,1
    INT 21H
    MOV BL,AL
    MOV FIRSTNUM,AL
    INT 21H
    MOV SECONDNUM,AL

    ADD BL,AL
    SUB BL,30H
    MOV ANS,BL

    MOV AH,9
    LEA DX,STR1
    INT 21H

    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.

Algorithms :

1.Start the program.

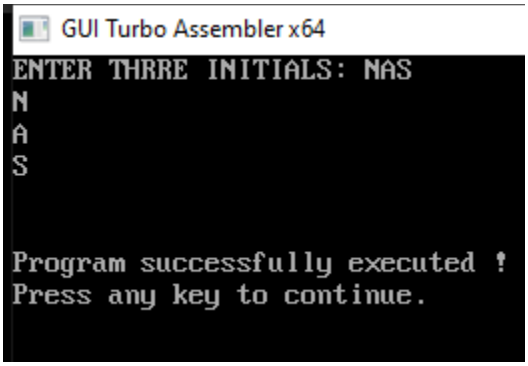
2. Read three intitials input .

3. These three initials stored into FIRST, SECOND ,THIRD register respectively.
4. Then break line and show these intital left margin
5.Stop the program.

Source Code :

```
.MODEL SMALL
.STACK 100H
.DATA
   STR DB 'ENTER THRRE INITIALS: $'
   STR1 DB '',0AH,0DH
   FIRST DB ?
   STR2 DB '',0AH,0DH
   SECOND DB ?
   STR3 DB '',0AH,0DH
   THIRD DB ?
   STR4 DB '$'
.CODE
MAIN PROC

   MOV AX,@DATA
   MOV DS,AX

   MOV AH,9
   LEA DX,STR
   INT 21H

   MOV AH,1
   INT 21H
   MOV FIRST,AL
   INT 21H
   MOV SECOND,AL
   INT 21H
   MOV THIRD,AL

   MOV AH,9
   LEA DX,STR1
   INT 21H

   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.

Algorithms :

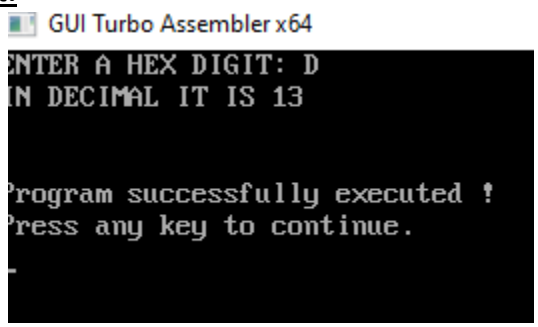
1.Start the program.
2. Input one hex digits (A-F) .
3. And subtraction 11H from these input .
4. Then its convert into binary form.
5.Stop the program.

Source Code :

```
 .MODEL SMALL
 .STACK 100H
.DATA
   STR1 DB 'ENTER A HEX DIGIT: $'
   STR2 DB 0AH,0DH,'IN DECIMAL IT IS 1'
   ANS DB ?
   STR3 DB '$'
.CODE
MAIN PROC

   MOV AX,@DATA
   MOV DS,AX

   MOV AH,9
   LEA DX,STR1
   INT 21H

   MOV AH,1
   INT 21H
```

```
   SUB AL,11H
   MOV ANS,AL

   MOV AH,9
   LEA DX,STR2
   INT 21H

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

Output:



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

Algorithms :

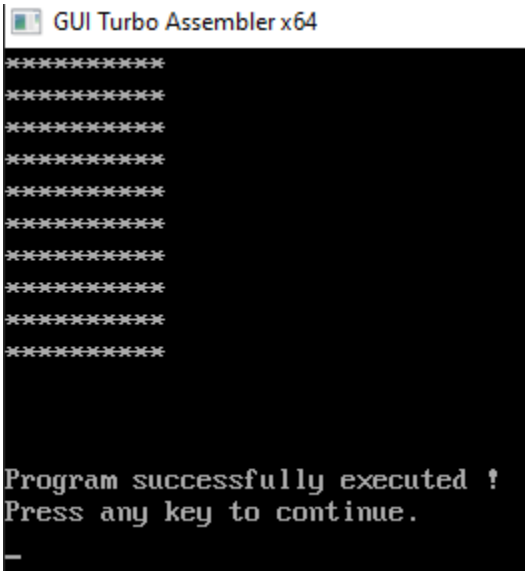
1.Start the program.
2. In data segment take a string looks like (**********).
3.Then write 'int 21h' ten times.
4.Stop the program.

Source Code:

```
.MODEL SMALL
.STACK 100H

.DATA
  SQUARE  DB  '**********',0DH,0AH,'$'

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

    LEA DX, SQUARE          ; load the string
```

```
    MOV AH, 9

    INT 21H
    INT 21H                ; display the string 10 times
    INT 21H
    INT 21H
    INT 21H
    INT 21H
    INT 21H
    INT 21H
    INT 21H
    INT 21H

    MOV AH, 4CH            ; return control to DOS
    INT 21H
  MAIN ENDP
END MAIN
```

Output :



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

Algorithms :

1. Start the program.
2. Display '?' character.
3. Read three initials (z,a,f).
4. Take asterisks in data segment
5. Load the string asterisks and Loop it 11*11 times
6.stop the program.

Source Code :

```
.MODEL SMALL
.STACK 100H

.DATA
  PROMPT     DB  0DH,0AH,'Enter three initials : $'
  ASTERISKS  DB  '***********',0DH,0AH,'$'
  NEXT_LINE  DB  0DH,0AH,"$"

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

  MOV AH, 2                  ; display "?"
  MOV DL, "?"
  INT 21H

  LEA DX, PROMPT             ; load and display the string PROMPT
  MOV AH, 9
  INT 21H

  MOV AH, 1
  INT 21H

  MOV BL, AL

  INT 21H

  MOV BH, AL

  INT 21H

  MOV CL, AL
  LEA DX, NEXT_LINE
  MOV AH, 9
  INT 21H
  INT 21H

  LEA DX, ASTERISKS          ; load the string ASTERISKS
```

```
        MOV AH, 9

        INT 21H                 ; display the string ASTERISKS 5 times
        INT 21H
    INT 21H
    INT 21H
    INT 21H

    MOV ASTERISKS+4, BL         ; place the three initials in the position
    MOV ASTERISKS+5, BH         ; of middle asterisks i.e. 4,5,6.
    MOV ASTERISKS+6, CL

    INT 21H                     ; display the modified string ASTERISKS

    MOV ASTERISKS+4, "*"        ; place the "*" back in their original
    MOV ASTERISKS+5, "*"        ; position
    MOV ASTERISKS+6, "*"

    INT 21H                     ; print the string ASTERISKS 5 times
    INT 21H
    INT 21H
    INT 21H
    INT 21H

    MOV AH, 2
    MOV DL, 7H
    INT 21H

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

Output:

```
GOTurbo Assembler x64
?
Enter three initials : zaf

************
************
************
************
************
****zaf****
************
************
************
************
************



Program successfully executed !
Press any key to continue.
```

Mawlana Bhashani Science and Technology University

Santosh, Tangail-1902.

Lab Report

Department of Information and Communication Technology

Report No: 03

Report Name: Assembly language Program.

Course Title: Microprocessor and Assembly Language Lab

Course Code: ICT-3106

Submitted By	Submitted To
Name: **Zafrul Hasan Khan** ID: **IT-18003** Session: 2017-18 3rd Year 1st Semester Dept. of Information & Communication Technology, MBSTU.	**S.M. Shamim** **Lecturer,** Dept. of Information & Communication Technology, MBSTU.

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

Source Code :

```
.model small
.stack 1024h
.code

start:
   mov cx, 5
first:
   mov bl, 2ah
   mov bh, 1
   call    drawall
   loop    first

   mov dx, 5
second:
   mov bl, 20h
   mov bh, 0
   mov cx, dx
   call    drawall
   mov cx, 6
   sub cx, dx
   mov bl, 2ah
   mov bh, 1
   call    drawall
   dec dx
   jnz second
   mov ax, 4c00h
   int 21h


drawall:
   push    ax
   push    bx
   push    cx
   push    dx
drawone:
   mov     ah,2h
   mov     dl,bl
   int     21h
   loop    drawone
```

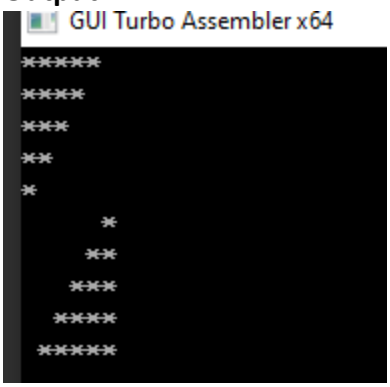
```
    or  bh, bh
    jz  retorn
    mov     dl,0Ah
    int     21h
    mov dl,0Dh
    int 21h
retorn:
    pop dx
    pop cx
    pop bx
    pop ax
    ret

    end start
```

Another starts triangle :

```
.MODEL SMALL
.STACK  50H
.DATA
    NL  DB  0DH, 0AH, '$'        ; NL = NEXT LINE
.CODE
MAIN PROC
    MOV AX, @DATA
    MOV DS, AX

    MOV CX, 5
    MOV BX, 1

    FOR_1:
        PUSH    CX
        MOV DL, 20H    ; 20H IS ASCII CODE FOR SPACE
        MOV AH, 2

        FOR_2:
            INT 21H    ; PRINTING SPACES
        LOOP    FOR_2

        MOV CX, BX
        MOV DL, '*'
        MOV AH, 2
```

```
        FOR_3:
          INT     21H    ; PRINTING STARS
        LOOP    FOR_3

        LEA DX, NL
        MOV AH, 9
        INT     21H
        INC BX
        POP CX

      LOOP    FOR_1

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

Output :



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:

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

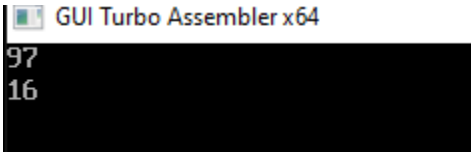
<u>Algorithm:</u>

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.

Source 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:

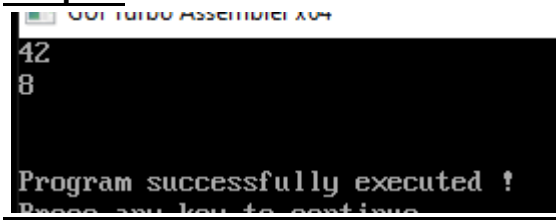
1.Start a program.
2.Enter first number in 'bl' register.
3.Enter second number from 'al' register and multiply it with 'bl' register.
4.Move the value in bl register,
5.Add 48 with bl register.
6.Display it.

Source 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:



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

Algorithm:

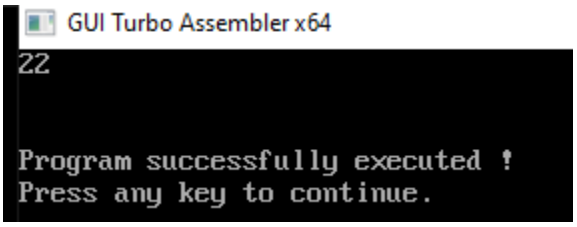
1.Start the program.
2.Enter two numbers.
3.Move them to ‘bl’ and ‘al’ register accordingly.
4.Divide ‘al’ register by ‘bl’ register.
5.Display bl register and bh register.
6.stop the program.

Source Code:

```
.model small
.stack 100h
.data
.code
main proc
 mov al,8
 mov bl,3

 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 :



Mawlana Bhashani Science and Technology University

Santosh, Tangail-1902.

Lab Report

Department of Information and Communication Technology

Report No: 04

Report Name: Assembly language Program.

Course Title: Microprocessor and Assembly Language Lab

Course Code: ICT-3106

Submitted By	Submitted To
Name: **Zafrul Hasan Khan** ID: **IT-18003** Session: 2017-18 3rd Year 1st Semester Dept. of Information & Communication Technology, MBSTU.	**S.M. Shamim** **Lecturer,** Dept. of Information & Communication Technology, MBSTU.

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

Algorithms :

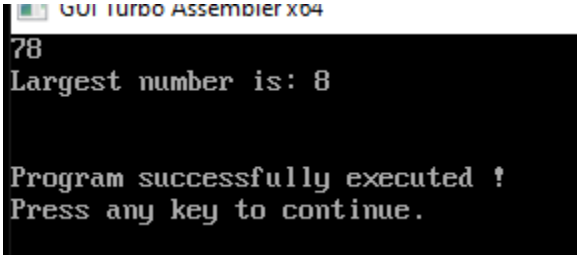
1.Start the program.
2. Read the two integer number
3.Then use CMP using compare two numbers .
4. use JG for jumping to greatest number.
5. Stop the program.

Source Code:

```
.model small
.stack 100h
.DATA
MSG1 DB 'Largest number is: $'

.code
main proc
   MOV AX,@DATA    ;DATA SEGMENT
   MOV DS,AX

   mov ah,1
   int 21h
   mov bl,al

   mov ah,1
   int 21h
   mov bh,al

   lp:
   cmp bl,bh
   MOV AH,2
   MOV DL,0AH
   INT 21H
   MOV DL,0DH
   INT 21H

   LEA DX,MSG1
   MOV AH,9       ;MSG1
```

```
    INT 21H
    jg greater

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

    greater:

    mov ah,2
    mov dl,bl
    int 21h

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

Output:



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

Algorithms :

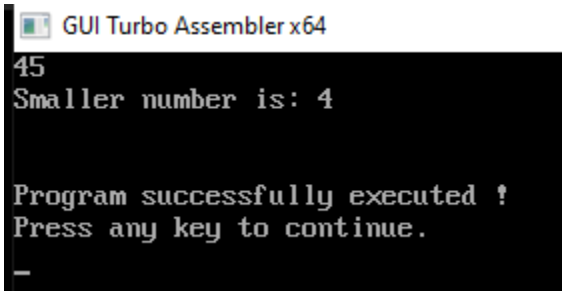
1.Start the program.
2. Read the two integer number
3.Then use CMP using compare two numbers .
4. use JL for jumping to smallest number.
5. Stop the program.

Source Code:

```
.model small
.stack 100h
.DATA
MSG1 DB 'Smaller number is: $'

.code
main proc
   MOV AX,@DATA    ;DATA SEGMENT
   MOV DS,AX

   mov ah,1
   int 21h
   mov bl,al

   mov ah,1
   int 21h
   mov bh,al

   lp:
   cmp bl,bh
  MOV AH,2
   MOV DL,0AH
   INT 21H
   MOV DL,0DH
   INT 21H

   LEA DX,MSG1
   MOV AH,9       ;MSG1
   INT 21H


   jg smaller

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

   smaller:
   mov ah,2
   mov dl,bh
```

```
    int 21h

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

Output :



3.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.·

Algorithms :

1.Start the program.
2. Input the one number
3.Then use CMP using check numbers by zero .
4. if it is JG, then its positive and put 1 in BL.
4. if it is JL, then its negative and put -1 in Al.
4. otherwise its zero.
5. Stop the program.

Source Code :

```
.model small
.stack 100h

.code
main proc
    mov ah,1
    int 21h

    cmp ax,0
```

```
    jl negative
    je zero
    jg positive

    negative:
    mov bx,-1
    jmp exit

    positive:
    mov bx,1
    jmp exit

    zero:
    mov bx,0
    jmp exit

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

3. **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".**

Algorithms :

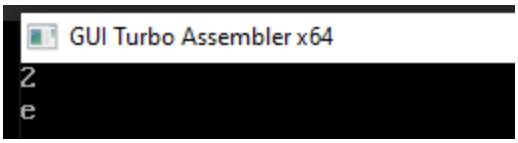
1.Start the program.
2. Input the one value that is entered into Al
3. If the value match with 1 and 3 then its display ‘o’ .
4. If the value match with 2 and 4 then its display ‘e’ .
5. Stop the program.

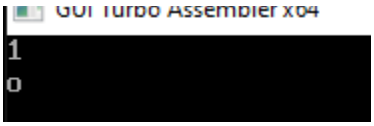
<u>Source Code :</u>

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

```
        int 21h
        cmp al,"1"
        je odd
        cmp al,"3"
        je odd
        cmp al,"2"
        je even
        cmp al,"4"
        je even
         odd:
        MOV AH,2
        MOV DL,0AH
        INT 21H
        MOV DL,0DH
        INT 21H

        mov ah,2
        mov dl,"o"
        int 21h
        jmp exit;

        even:
        MOV AH,2
        MOV DL,0AH
        INT 21H
        MOV DL,0DH
        INT 21H

        mov ah,2
        mov dl,"e"
        int 21h
        jmp exit;

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

Input & Output :





4. **Write an assembly program to enter a character if it's an uppercase latter, display it otherwise terminate.**

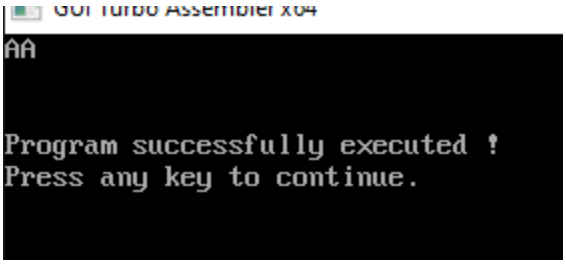
Algorithms :

1.Start the program.
2. Input a character
3. Then check the character inside the uppercase letters .
4. if it is yes , then show that .
5. Stop the program.

Source Code :

```
.model small
.stack 100h

.code
main proc
   mov ah,1
   int 21h

   lp:
   cmp al,"A"
   jnge exit
   cmp al,"Z"
   jnle exit

   mov ah,2
   mov dl,al
   int 21h

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

Output:



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

Algorithms :

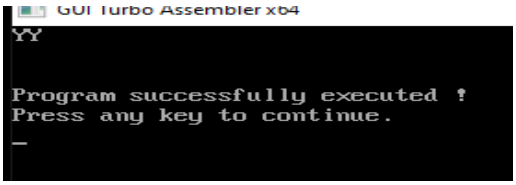
1.Start the program.
2. Input a character
3. Then check the character , if it is 'y' or 'Y' .
4. if it is yes , then show that .
4. Otherwise terminate the program
5. Stop the program.

Source Code :

```
.model small
.stack 100h

.code
main proc
   mov ah,1
   int 21h

   lp:
   cmp al,"y"
   je eq
   cmp al,"Y"
   je eq
   jmp exit

   eq:
   mov ah,2
   mov dl,al
   int 21h

   exit:
   mov ah,4ch
```

```
    int 21h
    main endp
end main
```

Output:



Mawlana Bhashani Science and Technology University

Santosh, Tangail-1902.

Lab Report

Department of Information and Communication Technology

Report No: 05

Report Name: Assembly language Program.

Course Title: Microprocessor and Assembly Language Lab

Course Code: ICT-3106

Submitted By	Submitted To
Name: **Zafrul Hasan Khan** ID: **IT-18003** Session: 2017-18 3rd Year 1st Semester Dept. of Information & Communication Technology, MBSTU.	**S.M. Shamim** **Lecturer,** Dept. of Information & Communication Technology, MBSTU.

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

Algorithms :

1.Start the program.
2. use Loop and print asterisks 80 times .
3.Stop the program.

Source Code :

```
.model small
.stack 100h
.data
mg1 db ?
mg2 db ?
.code
main proc

 mov ax,@data
 mov ds,ax

 mov cx,80
 jcxz skip
 top:
  mov ah,2
  mov dl,'*'
  int 21h
  loop top
  jmp skip

  skip:
 mov ah,4ch
 int 21h

 main endp
 end main
```

Output:

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

Algorithms :

1.Start the program.
2. Initialize ‘cx’ register with the value 9.

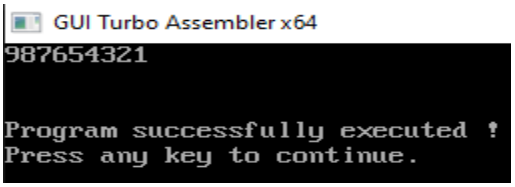
3.create a level named top, print 57,decrement the value of ‘dl’ register.Loop the level.

4.Stop the program.

Source Code :

```
.stack 100h
.data
mg1 db ?
mg2 db ?
.code
main proc
mov ax,@data
mov ds,ax
 mov cx,9
 jcxz skip
 top:
  mov ah,2
  mov bx,cx
  add bx,30h
  mov dl,' '
  mov dx,bx
  int 21h
  loop top
  jmp skip

  skip:
 mov ah,4ch
 int 21h

 main endp
 end main
```

Output:



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

Algorithm:

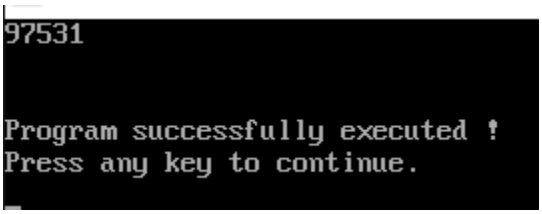
1.Start the program.
2. Initialize 'cx' register with the value 5.
3.Create a level named top, print 57 ascii character .
4. Decrement the value of 'dl' register by 2. Loop the level.
4.Stop the program.

Source Code :

```
.model small
.stack 100h
.data
mg1 db ?
mg2 db ?
.code
main proc

 mov ax,@data
 mov ds,ax

 mov cx,10
 jcxz skip
 top:
  mov ah,2
  sub cx,1
  mov bx,cx
  add bx,30h
  mov dx,bx
  int 21h
  loop top
  jmp skip
```

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

main endp
end main
```

Output:



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

Algorithm:

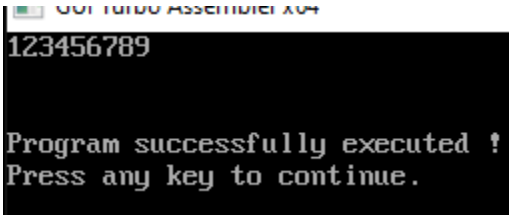
1.Start the program.
2. Initialize ‘cx’ register with the value 9.
3.Create a level named top, print 49 ascii values character.
4. Increment the value of ‘dl’ register, Loop the level.
5. Stop the program.

Source Code :

```
.model small
.stack 100h

.code
main proc


mov cx,9
mov ah,2
mov dl,49
top:
int 21h


 inc dl
 loop top
```

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

main endp
end main
```

Output :



11. W rite an assembly program to print the following series (for) 8 6 4 2

Algorithm:

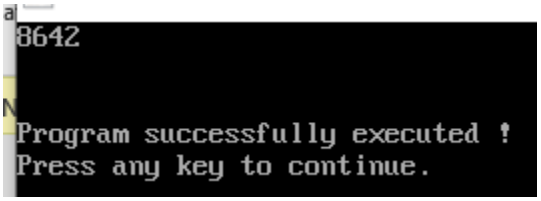
1.Start the program.
2. Initialize ‘cx’ register with the value 4.
3. create a level named top .
4. Print 56 ascii values , decrement the value of ‘dl’ register . Loop the level.
5. Stop the program.

Source Code :

```
.model small
.stack 100h
.data
mg1 db ?
mg2 db ?
.code
main proc

 mov ax,@data
 mov ds,ax

 mov cx,8
 jcxz skip
 top:
  mov ah,2

  mov bx,cx
  add bx,30h
```

```
mov dx,bx
sub cx,1
int 21h
loop top
jmp skip

skip:
mov ah,4ch
int 21h

main endp
end main
```

Output :



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

Algorithm:

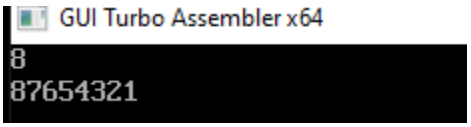
1.Start the program.
2. Initialize 'dl' register with the value 57.
3. create a level named while__, print 57,decrement the value of 'dl' register.
4. Compare the value of 'dl' register with the value 49.
5. If 'dl' register's value is less then 49 then jump to exit level otherwise jump to while_ level.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

Source Code:

```
.model small
.stack 100h
.data
mg1 db ?
mg2 db ?
.code
main proc

mov ax,@data
```

```
mov ds,ax

mov ah,1
int 21h
sub al,30h
mov bl,al

MOV AH,2
MOV DL,0AH
INT 21H
MOV DL,0DH
INT 21H

while_:
cmp bl,0
je exit
mov ah,2
mov cl,bl
add cl,30h
mov dl,cl
int 21h
dec bl
jmp while_

exit:
mov ah,4ch
int 21h

main endp
end main
```

Output :



Mawlana Bhashani Science and Technology University

Santosh, Tangail-1902.

Lab Report

Department of Information and Communication Technology

Report No: 06

Report Name: Assembly language Program.

Course Title: Microprocessor and Assembly Language Lab

Course Code: ICT-3106

Submitted By	Submitted To
Name: **Zafrul Hasan Khan** ID: **IT-18003** Session: 2017-18 3rd Year 1st Semester Dept. of Information & Communication Technology, MBSTU.	**S.M. Shamim** **Lecturer,** Dept. of Information & Communication Technology, MBSTU.

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.

Source 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:

```
8Even


Program successfully executed !
Press any key to continue.
```

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.

Source 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:

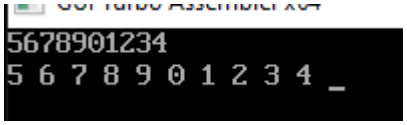
04. Write Assembly program to read ten (10) characters from console.

<u>Source code:</u>

```
.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

   main endp
end main
```

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.
5. And Display the smaller number.
5.Stop the program

Source 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 :

```
89
8

Program successfully executed !
```

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

Algorithm:

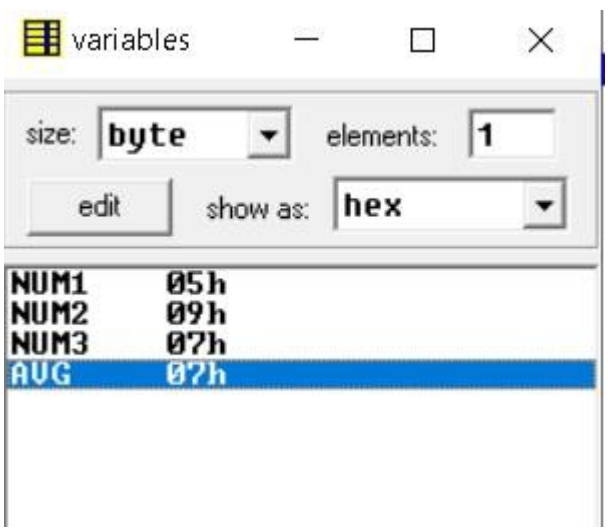
1.Start the program.
2.Define three variables.
3.Initialize those variables.
4.Move num1 to al register.add num2 and num3 to al register.
5.set the value of ah register value as 0
6.Set the value of dl register as 3.
7.perform div operation.
8.Stop the program.

Source 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.

Source 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
main proc
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
   main endp
end main
```

Output:

8. 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.

Source 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:

```
ENTER TEMPERATURE IN CELSIUS (ONLY IN 2 DIGITS): 23
CONVERTED IS FAHRENHEIT (TEMPERATURE): 73
```