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.

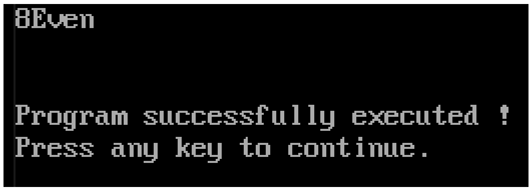
4.If even print “Even” otherwise print “Odd”.

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:

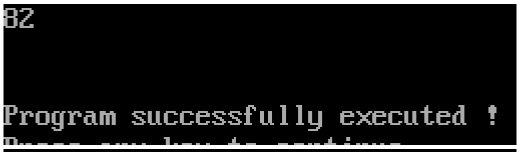
```
DATA SEGMENT
NUM1 DB 7H
NUM2 DB ?

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:



```
82


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

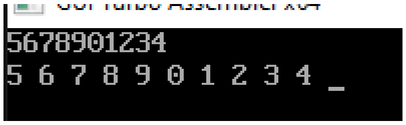
04. Write Assembly program to read ten (10) characters from console.

Source 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



   main endp

end main
```

Output :



```
5678901234
5 6 7 8 9 0 1 2 3 4 _
```

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

<u>Source Code :</u>

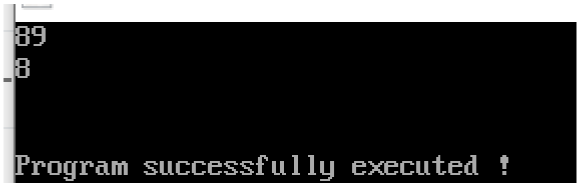
```
.model small

.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.

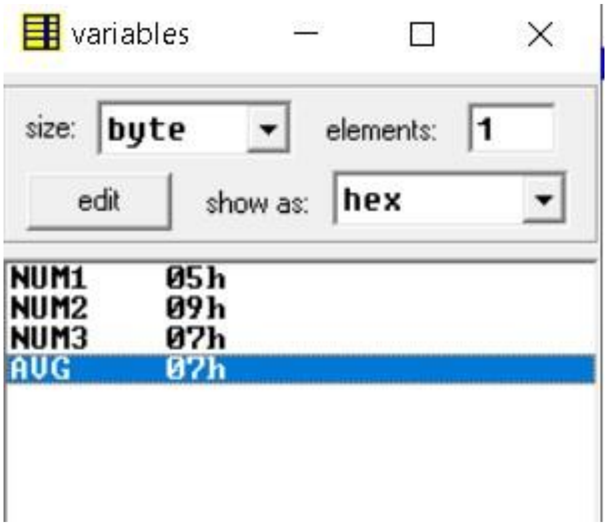
7.perform div operation.

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:



size: byte elements: 1
show as: hex

```
NUM1     05h
NUM2     09h
NUM3     07h
AVG      07h
```

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.

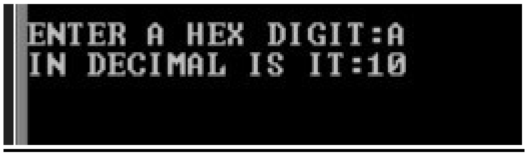
6.Stop the program.

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:



```
ENTER A HEX DIGIT:A
IN DECIMAL IS IT:10
```

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:

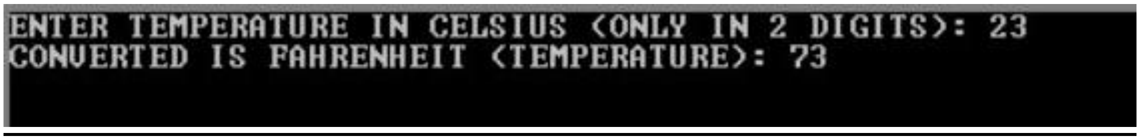
```
DATA SEGMENT

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
```