System Programming
Report

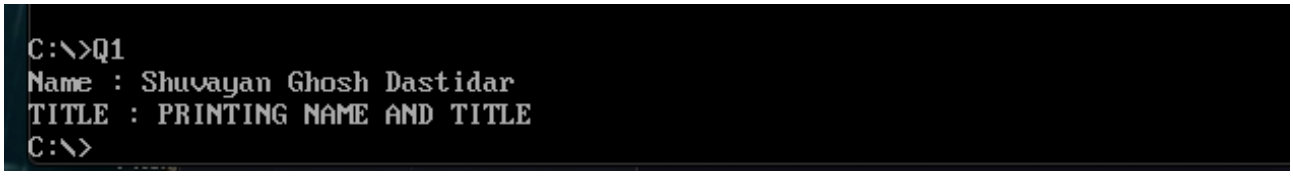
Name: Shuvayan Ghosh Dastidar
Roll : 001810501044
JU-BCSE-3

Assignment 1

Q1.Write and test a MASM program to Display your name and program title on the output screen.

```
.model SMALL
.stack 100H

.data

myname DB "Name : Shuvayan Ghosh Dastidar",13,10,"$"
mytitle DB "TITLE : PRINTING NAME AND TITLE$"


.code
        MOV AX, @DATA
        MOV DS, AX

        LEA DX, myname
        MOV AH, 09H
        INT 21H

        LEA DX, mytitle
        INT 21H

        MOV AH, 4CH
        INT 21H

end
```

OUTPUT



Q2.Write and test a MASM program to convert a letter from uppercase to lowercase.

```
.MODEL SMALL
.STACK 100H

.DATA

MSG1 DB "ENTER A UPPER CASE LETTER :$"
MSG2 DB 0DH,0AH, "IN LOWER CASE : "
CHAR DB ?, "$"

.CODE


MAIN PROC
        MOV AX,@DATA
        MOV DS,AX
        LEA DX,MSG1
        MOV AH,9
        INT 21H

        ; INPUT A CHARACTER

        MOV AH,1
        INT 21H
        ADD AL, 20H ; CONVERT TO UPPER CASE
        MOV CHAR, AL ;STORE THE CHARACTER

        LEA DX, MSG2
        MOV AH,9
        INT 21H

        ;EXIT
        MOV AH,4CH
        INT 21H


MAIN ENDP
END MAIN
```

```
Run File [Q2.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

C:\>q2
ENTER A UPPER CASE LETTER :U
IN LOWER CASE : u
C:\>
```

Output

Q3.Write and test a MASM program to add two Hexadecimal Numbers.

```
.MODEL SMALL
.STACK 100H

.DATA

MSG DB "ADDING NUMBERS IN VAL1 AND VAL2..",13,10,"$"
VAL1 DB 10H
VAL2 DB 31H
VAL3 DB ? ; EXPECTED 65 OR 41H

.CODE


MAIN PROC
        MOV AX,@DATA
        MOV DS,AX
        LEA DX,MSG
        MOV AH,9
        INT 21H
        MOV AL, VAL1
        ADD AL, VAL2 ; ADDING

        MOV VAL3, AL ; STORING IN VAL3
        MOV DL, VAL3
        MOV AH, 02H
        INT 21H ; PRINTING VAL3


        ;EXIT
        MOV AH,4CH
        INT 21H


MAIN ENDP
END MAIN


OUTPUT : EXPECTED 41H OR 65 OR ‘A’
```

```
Libraries [.LIB]:

C:\>Q3
ADDING NUMBERS IN VAL1 AND VAL2..
A
C:\>
```

Q4.Write and test a MASM program to find the second max and second min from an array.

```
.model SMALL
.stack 100H

;-----------------
;prints second maximum
;---------------

.data

array db 19, 50, 30, 24, 23, 16, 17, 98, 86

len dw $-array
max db 0
smax db 0
min db 100
smin db 100
linefeed db 13, 10, "$"

.code

MAIN PROC
        MOV AX, @DATA
        MOV DS, AX
        LEA SI, array
        MOV AX,0
        MOV AL, array[SI]
        MOV max,AL
        MOV smax, AL
        MOV CX, len

REPEAT:
        MOV AX,0
        MOV AL, array[SI]
        CMP max, AL
        JG CHECKSMAX
        MOV BL, max
        MOV smax, BL
        MOV max, AL ; UPDATE MAX IF MAX IS SMALL
        JMP CONT1
```

```
CHECKSMAX:

        CMP smax, AL
        JG CONT1
        MOV smax, AL
CONT1:
        MOV AX,0
        MOV AL, array[SI]
        CMP min, AL
        JL CHECKSMIN
        MOV BL, min
        MOV smin, BL
        MOV min, AL ; UPDATE MAX IF MAX IS SMALL
        JMP CONT2

CHECKSMIN:
        CMP smin, AL
        JL CONT2
        MOV smin, AL
CONT2:


        INC SI
        LOOP REPEAT

        ; PRINT MAX AND SECOND MAX


        MOV AX, 0
        MOV AL , smax
        CALL PRINT

        MOV AX, 0
        MOV AL , smin
        CALL PRINT
        ; CALL NEWLINE

        MOV AH, 4CH
        INT 21H
MAIN ENDP

NEWLINE PROC
        MOV DX,0
        MOV AH,0
        MOV AH,9
        LEA DX, linefeed
        INT 21H
NEWLINE ENDP

PRINT PROC
```

```
        ;initilize count
        MOV CX,0
        MOV DX,0
label1:
        ; if ax is zero
        cmp AX,0
        je print1

        ;initilize bx to 10
        mov BX,10

        ; extract the last digit
        div BX

        ;push it in the stack
        push DX

        ;increment the count
        inc CX

        ;set dx to 0
        xor DX, DX
        jmp label1
        print1:
        ;check if count
        ;is greater than zero
        cmp CX,0
        je exit

        ;pop the top of stack
        pop DX

        ;add 48 so that it
        ;represents the ASCII
        ;value of digits
        add DX,48

        ;interuppt to print a
        ;character
        mov AH,02h
        int 21h

        ;decrease the count
        dec CX
        jmp print1
exit:
        RET
PRINT ENDP
END MAIN
```

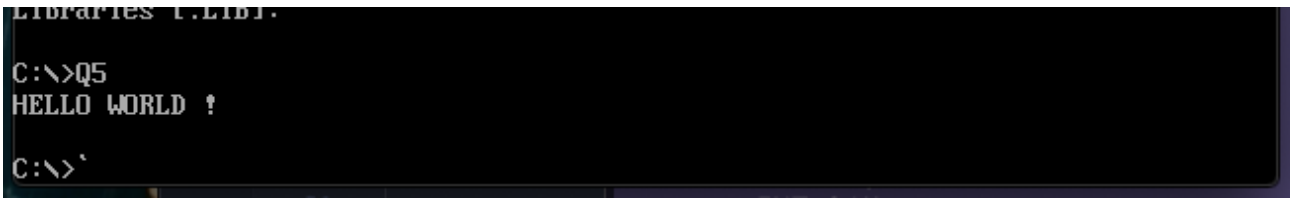
OUTPUT

```
OF ARRAY: array db 19, 50, 30, 24, 23, 16, 17, 98, 86
EXPECTED : 86;17
```

Q5.Write and test a MASM program to display a terminating message.

```
.model SMALL
.stack 100H

.data

MESSAGE DB "HELLO WORLD !",13,10,"$"



.code
        MOV AX, @DATA
        MOV DS, AX

        LEA DX, MESSAGE
        MOV AH, 09H
        INT 21H

        MOV AH, 4CH
        INT 21H

end
```

OUTPUT



Q6.Write and test a MASM program to Take a character from the keyboard and print it.

```
.MODEL SMALL
.STACK 100H

.DATA

MSG1 DB "INPUT A CHARACTER :","$"

.CODE


MAIN PROC
        MOV AX,@DATA
        MOV DS,AX
        LEA DX,MSG1
        MOV AH,9
        INT 21H


        ; INPUT A CHARACTER
        MOV AH,1
        INT 21H

        MOV DL, AL ; STORE IN DL FOR PRINTING

        MOV AH, 02H
        INT 21H


        ;EXIT
        MOV AH,4CH
        INT 21H


MAIN ENDP
END MAIN
```

Q7. Write and test a MASM program to validate second numbers is less than the first.

```
.MODEL SMALL
.STACK 100H

.DATA

MSG1 DB "ENTER TWO NUMBERS",13,10,"$"
NUM1 DB ?
NUM2 DB ?
```

```
MSG2 DB 13,10,"SECOND NUMBER IS SMALLER OK..$"
MSG3 DB 13,10,"SECOND NUMBER IS GREATER FAILED VALIDATION ..$"
IN1 DB 13,10,"ENTER FIRST NUMBER: $"
IN2 DB 13,10,"ENTER SECOND NUMBER: $"

.CODE

MAIN PROC
        MOV AX,@DATA
        MOV DS,AX
        LEA DX,MSG1
        MOV AH,9
        INT 21H

        ; FIRST NUMBER
        MOV AH,9
        LEA DX, IN1
        INT 21H ;PROMPT
        MOV AH,1
        INT 21H  ;INPUT
        MOV NUM1,AL

        ; SECOND NUMBER
        MOV AH,9
        LEA DX, IN2
        INT 21H ;PROMPT
        MOV AH,1
        INT 21H ;INPUT
        MOV NUM2, AL


        MOV AL, NUM1
        CMP AL, NUM2
        JGE OK ; VALIDATION OK
        MOV AH,9
        LEA DX,MSG3
        INT 21H
        JMP EXIT


OK:
        MOV AH,9
        LEA DX, MSG2
        INT 21H
        ;EXIT
EXIT:
        MOV AH,4CH
        INT 21H


MAIN ENDP
END MAIN
```

```
C:\>Q7
ENTER TWO NUMBERS

ENTER FIRST NUMBER: 3
ENTER SECOND NUMBER: 4
SECOND NUMBER IS GREATER FAILED VALIDATION ..
C:\>Q7
ENTER TWO NUMBERS

ENTER FIRST NUMBER: 7
ENTER SECOND NUMBER: 4
SECOND NUMBER IS SMALLER OK..
C:\>
```

OUTPUT

Q8. Write and test a MASM program to find maximum and minimum from an array.

```
.MODEL SMALL
.STACK 100H

.DATA

array db 23, 19, 50, 30, 24, 23, 16, 17, 98, 86

len dw $-array
max db ?
min db ?
prompt1 db "The maximum  number : $"
prompt2 db "The minimum number : $"


linefeed db 13, 10, "$"

.CODE

MAIN PROC
        MOV AX, @DATA
        MOV DS, AX
        LEA SI, array
        MOV BL, array[SI]
        MOV max, BL
        MOV min, BL
        MOV CX, len
REPEAT:
        MOV AL, array[SI]
        CMP max, AL
        JG CONT1
        MOV max, AL ; UPDATE MAX IF MAX IS SMALL
CONT1:
        MOV AL, array[SI]
        CMP min, AL
        JL CONT2
```

```
        MOV min, AL ; UPDATE MAX IF MAX IS SMALL
CONT2:
        INC SI
        LOOP REPEAT



        MOV AL , max
        CBW            ; converts from byte to word
        CALL PRINT



        MOV AL, min
        CBW             ; converts from byte to word
        CALL PRINT



        MOV AH, 4CH
        INT 21H
MAIN ENDP

PRINT PROC

        ;initilize count
        MOV CX,0
        MOV DX,0
label1:
        ; if ax is zero
        cmp AX,0
        je print1

        ;initilize bx to 10
        mov BX,10

        ; extract the last digit
        div BX

        ;push it in the stack
        push DX

        ;increment the count
        inc CX

        ;set dx to 0
        xor DX, DX
        jmp label1
        print1:
        ;check if count
        ;is greater than zero
        cmp CX,0
        je exit

        ;pop the top of stack
        pop DX

        ;add 48 so that it
        ;represents the ASCII
        ;value of digits
```

```
        add DX,48

        ;interuppt to print a
        ;character
        mov AH,02h
        int 21h

        ;decrease the count
        dec CX
        jmp print1
exit:
        RET
PRINT ENDP
END MAIN
```

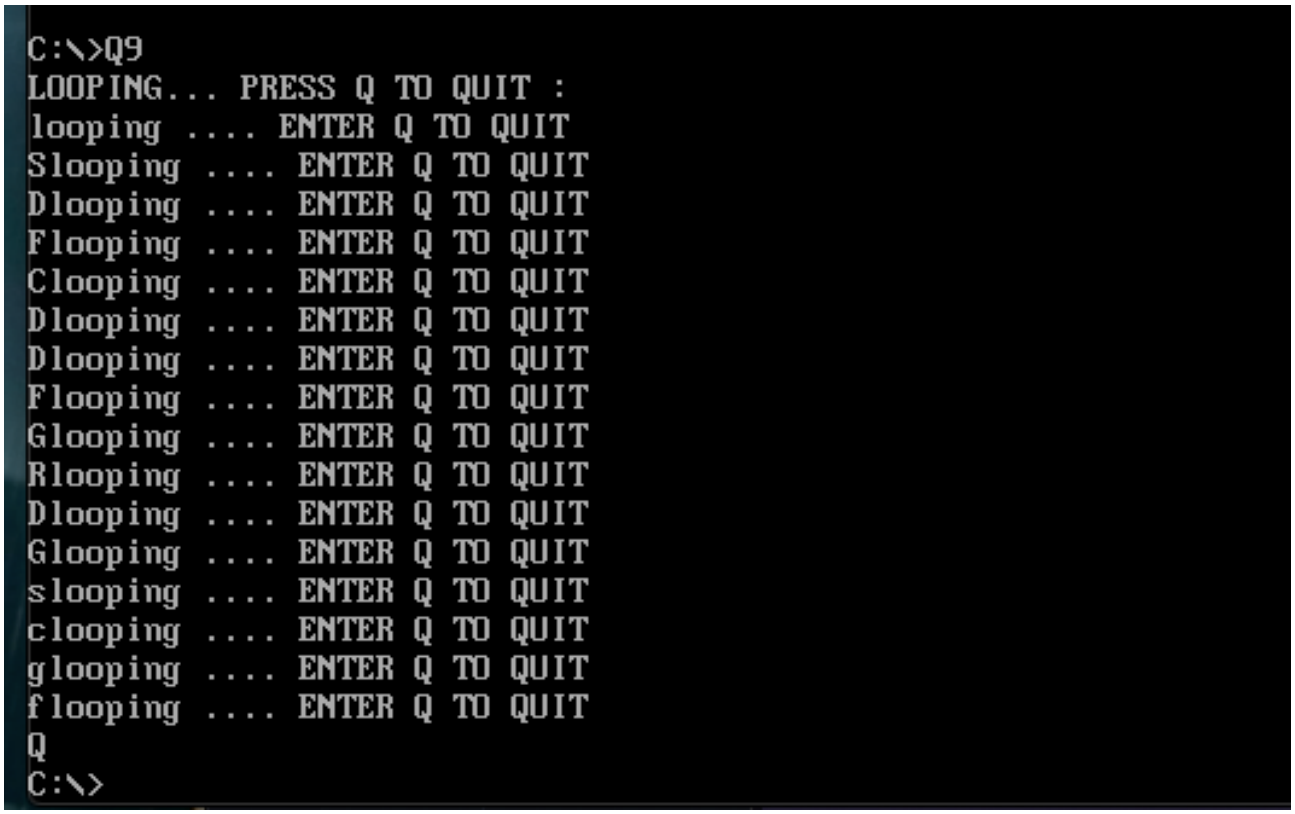
OUTPUT

Q9. Write and test a MASM program to loop until the user decides to quit.

```
.MODEL SMALL
.STACK 100H

.DATA

MSG1 DB "LOOPING... PRESS Q TO QUIT :",13,10,"$"
LMSG DB "looping .... ENTER Q TO QUIT",13,10,"$"
.CODE


MAIN PROC
        MOV AX,@DATA
        MOV DS,AX
        LEA DX,MSG1
        MOV AH,9
        INT 21H



LOOP:
        MOV AH,9
        LEA DX, LMSG
        INT 21H
        MOV AH,1
        INT 21H
        CMP AL, 'Q'
        JNZ LOOP

        ;EXIT
```

```
        MOV AH,4CH
        INT 21H


MAIN ENDP
END MAIN
```

OUTPUT



Q10. Write and test a MASM program to print all the characters from A-Z.

```
.MODEL SMALL
.STACK 100H

.DATA

MSG1 DB "PRINTING CHARACTERS FROM A TO Z",13,10,"$"

.CODE


MAIN PROC
        MOV AX,@DATA
        MOV DS,AX
        LEA DX,MSG1
        MOV AH,9
```

```
        INT 21H

        MOV DL,'A'
        MOV CX, 1AH
        MOV AH, 2
LOOP:

        INT 21H
        INC DL
        DEC CX
        JNZ LOOP

        ;ENDING PROG
        MOV AH, 4CH
        INT 21H


MAIN ENDP
END MAIN
```

OUTPUT