

Name:	Pankhania Aanandi R.
Roll No:	IT081
Batch:	I1

Experiment - 7

AIM: Study of DOS and BIOS function calls

Using the following DOS function call to write programs.

1. AH = 01h / INT 21h - read character from standard input, with echo.
Return: AL = character read.
2. AH = 02h / INT 21h - write character to standard output.
Input: DL = character to write
Return: AL = last character output
3. AH = 09h/INT 21h -write string to standard output
Input: DS: DX -> offset address of the string and the string is terminated with '\$'.
Return: AL = 24h
4. AH = 0Ah /INT 21h -buffered input
Entry: DS:DX -> buffer (reads from standard input)
Return: buffer filled with user input.

1. Write a program to take one character from the keyboard and echo on-screen.

Write your code here:

```
DATA SEGMENT  
MESSAGE DB "ENTER CHARACTER : $"  
MESSAGE1 DB "ENTERED CHARACTER: $"  
X DB ?  
DATA ENDS
```

```
CODE SEGMENT  
ASSUME DS: DATA, CS: CODE  
START:  
    MOV AX,DATA  
    MOV DS,AX  
    LEA DX,MESSAGE  
    MOV AH, 9      ; Print message
```

```
    INT 21H
    MOV AH, 1          ; read a character
    INT 21H

    MOV X, AL          ; save input character into X

    MOV AH, 2          ; carriage return
    MOV DL, 0DH
    INT 21H

    MOV DL, 0AH        ; line feed
    INT 21H

    LEA DX,MESSAGE1
    MOV AH, 9          ; Print message1
    INT 21H

    MOV AH, 2          ; display the character stored in X
    MOV DL, X
    INT 21H

    MOV AH, 4CH        ; return control to DOS
    INT 21H
CODE ENDS
END START
```

Compilation /Running and Debugging steps:

(As given in the lab manual as an example of multiplication program on page no:5 of lab manual)

```

DOS
BOX
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...
A:\>tasm charecho.asm
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file: charecho.asm
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 476k

A:\>tlink charecho.obj
Turbo Link Version 3.0 Copyright (c) 1987, 1990 Borland International
Warning: No stack

A:\>tasm charecho
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file: charecho.ASM
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 476k

A:\>

```

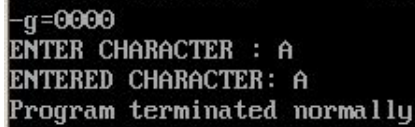
```

DOS
BOX
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...
DS=075A ES=075A SS=0769 CS=076D IP=0003 NV UP EI PL NZ NA PO NC
076D:0003 BDB8 MOV DS,AX
-q
A:\>debug CHARECHO.EXE
-l-u
076D:0000 B86A07 MOV AX,076A
076D:0003 BDB8 MOV DS,AX
076D:0005 BA0000 MOV DX,0000
076D:0008 B409 MOV AH,09
076D:000A CD21 INT 21
076D:000C B401 MOV AH,01
076D:000E CD21 INT 21
076D:0010 A22700 MOV [0027],AL
076D:0013 B402 MOV AH,02
076D:0015 B20D MOV DL,0D
076D:0017 CD21 INT 21
076D:0019 B20A MOV DL,0A
076D:001B CD21 INT 21
076D:001D BA1300 MOV DX,0013
-g=0000
ENTER CHARACTER : A
ENTERED CHARACTER: A
Program terminated normally
-

```

Output:

Screenshots of the output.



```
-g=0000
ENTER CHARACTER : A
ENTERED CHARACTER: A
Program terminated normally
```

2. Write a program to take one character from key board and convert into lowercase.

Write your code here:

DATA SEGMENT

MESSAGE DB "ENTER CHARACTER IN UPPERCASE : \$"

MESSAGE1 DB "CONVERTED CHARACTER INTO LOWERCASE : \$"

X DB ?

DATA ENDS

CODE SEGMENT

ASSUME DS: DATA, CS: CODE

START:

MOV AX,DATA

MOV DS,AX

LEA DX,MESSAGE

MOV AH, 9 ; Print message

INT 21H

MOV AH, 1 ; read a character

INT 21H

MOV X, AL ; save input character into X

MOV DL, 0AH ; line feed

INT 21H

LEA DX,MESSAGE1

```
        MOV AH, 9             ; Print message1
        INT 21H

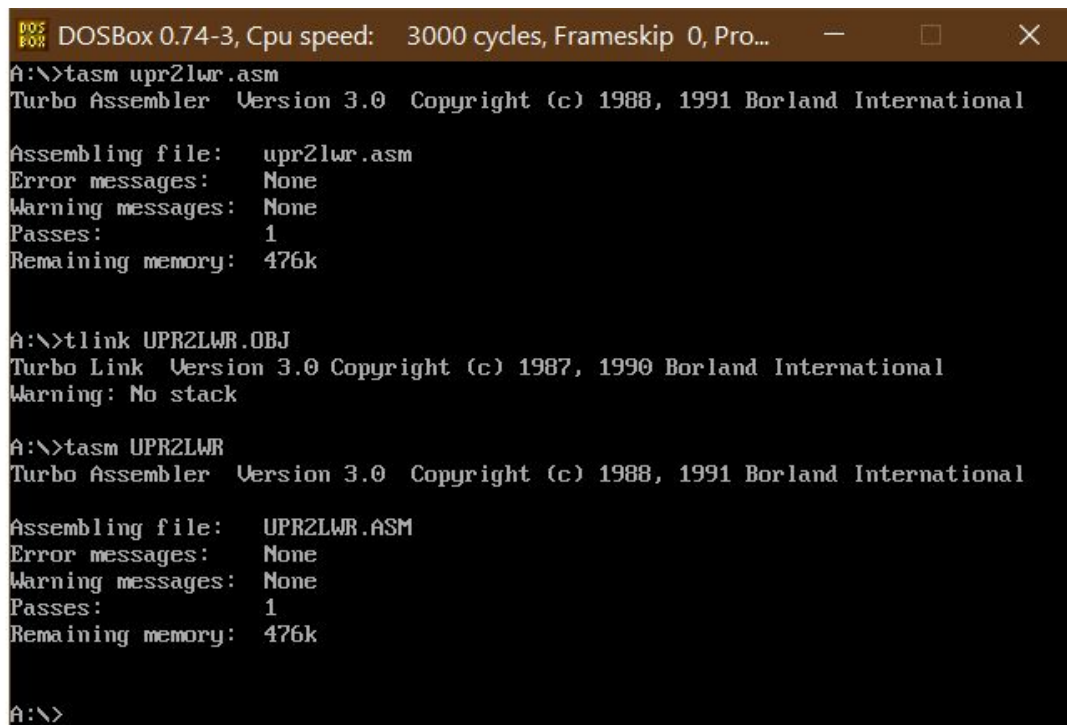
        OR X, 20H

        MOV AH, 2             ; display the character stored in X
        MOV DL, X
        INT 21H

        MOV AH, 4CH           ; return control to DOS
        INT 21H
CODE ENDS
END START
```

Compilation /Running and Debugging steps:

(As given in lab manual as an example of multiplication program on page no:5 of lab manual)

A screenshot of a DOSBox window titled "DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...". The terminal shows the following commands and output:

```
A:\>tasm upr2lwr.asm
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:   upr2lwr.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  476k

A:\>tlink UPR2LWR.OBJ
Turbo Link Version 3.0 Copyright (c) 1987, 1990 Borland International
Warning: No stack

A:\>tasm UPR2LWR
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file:   UPR2LWR.ASM
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  476k

A:\>
```

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...
-q
A:\>debug UPR2LWR.EXE
-u
076F:0000 B86A07      MOV     AX,076A
076F:0003 8ED8          MOV     DS,AX
076F:0005 BA0000      MOV     DX,0000
076F:0008 B409          MOV     AH,09
076F:000A CD21          INT     21
076F:000C B401          MOV     AH,01
076F:000E CD21          INT     21
076F:0010 A24600      MOV     [0046],AL
076F:0013 B20A          MOV     DL,0A
076F:0015 CD21          INT     21
076F:0017 BA2000      MOV     DX,0020
076F:001A B409          MOV     AH,09
076F:001C CD21          INT     21
076F:001E 800E460020    OR      BYTE PTR [0046],20
-g=0000
ENTER CHARACTER IN UPPERCASE : A
CONVERTED CHARACTER INTO LOWERCASE : a

```

Output:

Screenshots of the output.

```

-g=0000
ENTER CHARACTER IN UPPERCASE : A
CONVERTED CHARACTER INTO LOWERCASE : a

```

3. Write a program to get a string and convert this string from uppercase to lowercase.

Rules for Operands: Take your name as an input string and convert it.

Write your code here:

DATA SEGMENT

MESSAGE DB "ENTER STRING : \$"

STR1 DB 255 DUP(?)

MESSAGE1 DB "STRING AFTER CONVERSION : \$"

DATA ENDS

CODE SEGMENT

ASSUME DS: DATA, CS: CODE, ES: DATA

START:

MOV AX,DATA

MOV DS,AX

```
MOV ES,AX
```

```
MOV AH,09H
```

```
LEA DX,MESSAGE ; Print message
```

```
INT 21H
```

```
LEA SI,STR1
```

```
MOV AH,01H
```

```
;;Logic for conversion
```

```
READ:
```

```
INT 21H
```

```
MOV BL,AL
```

```
CMP AL,0DH
```

```
JE PRNT
```

```
XOR AL,20H
```

```
MOV BYTE PTR [SI],AL
```

```
INC SI
```

```
JMP READ
```

```
PRNT:
```

```
MOV AL,'$'
```

```
MOV BYTE PTR [SI],AL
```

```
;;end logic : string is converted
```

```
MOV AH,09H
```

```
LEA DX,MESSAGE1 ; Print message1
```

```
INT 21H
```

```
LEA DX,STR1 ; Print converted string
```

```
MOV AH,09H
```

INT 21H

MOV AH, 4CH ; return control to DOS

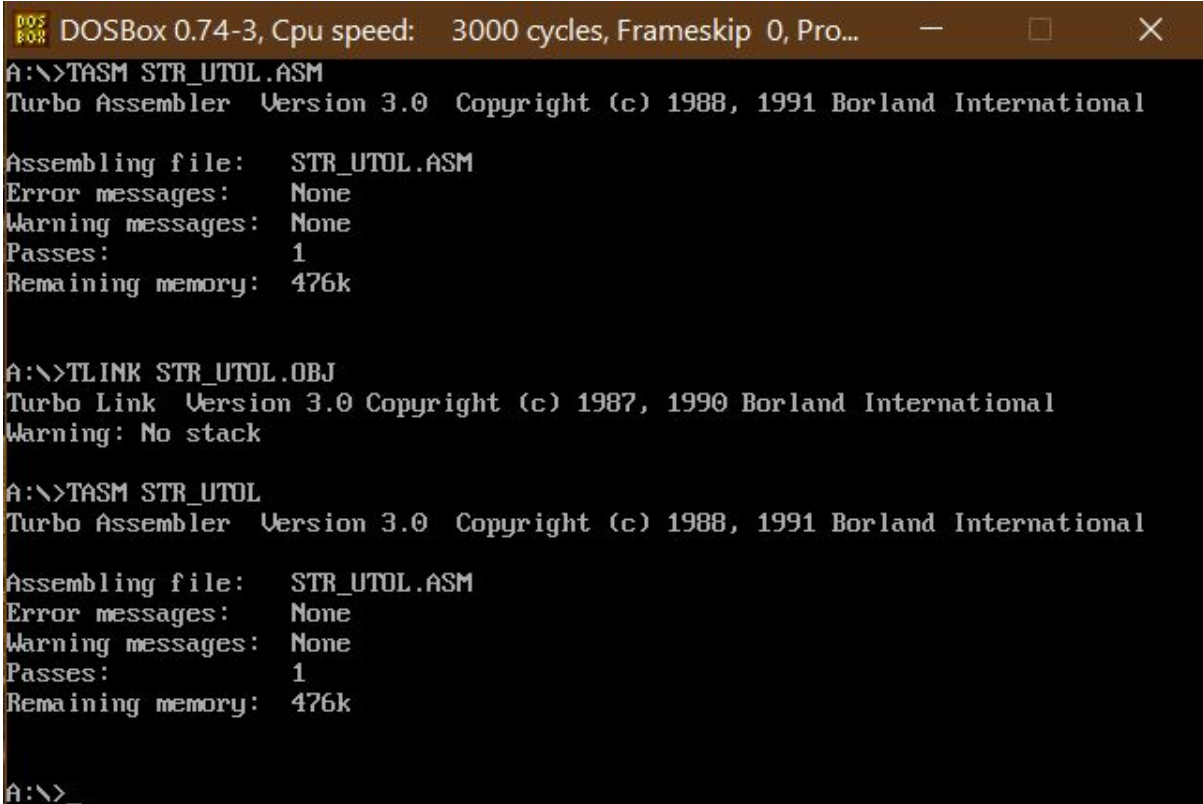
INT 21H

CODE ENDS

END START

Compilation /Running and Debugging steps:

(As given in lab manual as an example of multiplication program on page no:5 of lab manual)



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...
A:\>TASM STR_UTOL.ASM
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file: STR_UTOL.ASM
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 476k

A:\>TLINK STR_UTOL.OBJ
Turbo Link Version 3.0 Copyright (c) 1987, 1990 Borland International
Warning: No stack

A:\>TASM STR_UTOL
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file: STR_UTOL.ASM
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 476k

A:\>_
```



```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...
A:\>DEBUG STR_UTOL.EXE
-U
077D:0000 B86A07      MOV     AX,076A
077D:0003 8ED8        MOV     DS,AX
077D:0005 8EC0        MOV     ES,AX
077D:0007 B409        MOV     AH,09
077D:0009 BA0000     MOV     DX,0000
077D:000C CD21        INT     21
077D:000E BE1000     MOV     SI,0010
077D:0011 B401        MOV     AH,01
077D:0013 CD21        INT     21
077D:0015 8AD8        MOV     BL,AL
077D:0017 3C0D        CMP     AL,0D
077D:0019 7407        JZ      0022
077D:001B 3420        XOR     AL,20
077D:001D 8804        MOV     [SI],AL
077D:001F 46          INC     SI
-G=0000
ENTER STRING : AANANDI
STRING AFTER CONVERSION : aanandi
Program terminated normally
-G=0000
ENTER STRING : aanandi
STRING AFTER CONVERSION : AANANDI
A:\>_

```

Output:

Screenshots of the output. (Both strings should be present in your screenshot)

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...
A:\>_
-G=0000
ENTER STRING : AANANDI
STRING AFTER CONVERSION : aanandi
Program terminated normally
-G=0000
ENTER STRING : aanandi
STRING AFTER CONVERSION : AANANDI
A:\>_

```

- Here is a string "Hello Sunil Welcome to 8086 Microprocessor". Write an assembly Language program to convert the above string to "Hollo Sunil Wolcomo to 8086 Microprocossor" (Replace 'e' with 'o').

Rules for Operands: Take your name in place of "Sunil" and write the program.

Write your code here:

DATA SEGMENT

MESSAGE DB "HELLO AANANDI WELCOME TO 8086 MICROPROCESSOR \$"

```
    LEN EQU $-MESSAGE  
DATA ENDS
```

```
CODE SEGMENT
```

```
    ASSUME DS: DATA, CS: CODE
```

```
START:
```

```
    MOV AX,DATA
```

```
    MOV DS,AX
```

```
    MOV AH,09H
```

```
    MOV BX,SEG MESSAGE
```

```
    MOV DS,BX
```

```
    MOV DX,OFFSET MESSAGE
```

```
    INT 21H
```

```
    MOV AH,02H
```

```
    MOV DL,0AH
```

```
    INT 21H
```

```
    MOV AL,LEN
```

```
    LEA BX,MESSAGE
```

```
LOOP1:
```

```
    CMP MESSAGE[BX],65H
```

```
    JE RPLC
```

```
    CMP MESSAGE[BX],45H
```

```
    JE RPLC
```

```
    INC BX
```

```
    DEC AL
```

```
    CMP AL,00H
```

```
    JNE LOOP1
```

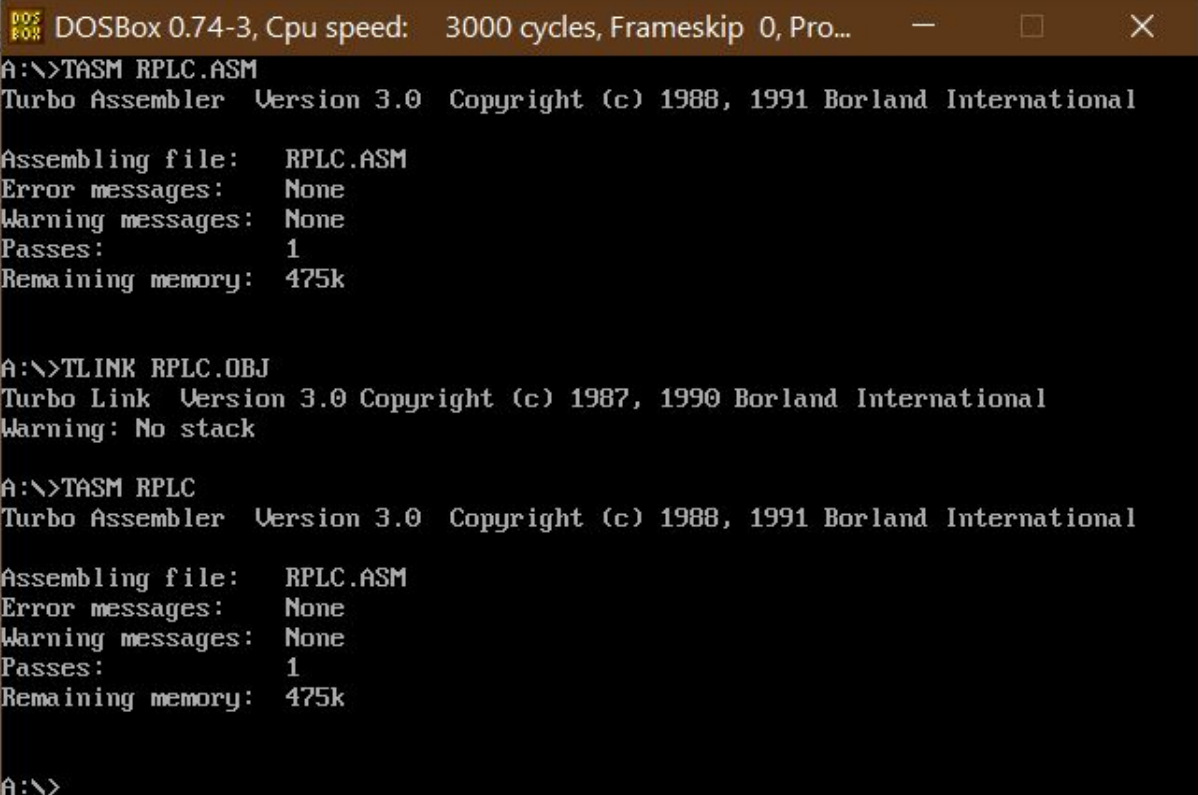
```
SHOW:
```

```
    MOV AH,09H
```

```
MOV BX,SEG MESSAGE
MOV DS,BX
MOV DX,OFFSET MESSAGE
INT 21H
EXIT:
INT 3H
RPLC:
ADD MESSAGE[BX],0AH
DEC AL
CMP AL,00H
JNE LOOP1
CODE ENDS
END START
```

Compilation /Running and Debugging steps:

(As given in the lab manual as an example of multiplication program on page no:5 of lab manual)



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...
A:\>TASM RPLC.ASM
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file: RPLC.ASM
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 475k

A:\>TLINK RPLC.OBJ
Turbo Link Version 3.0 Copyright (c) 1987, 1990 Borland International
Warning: No stack

A:\>TASM RPLC
Turbo Assembler Version 3.0 Copyright (c) 1988, 1991 Borland International

Assembling file: RPLC.ASM
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 475k

A:\>
```

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Pro...
A:\>DEBUG RPLC.EXE
-U
076D:0000 B86A07      MOV     AX,076A
076D:0003 8ED8        MOV     DS,AX
076D:0005 B409        MOV     AH,09
076D:0007 BB6A07      MOV     BX,076A
076D:000A 8EDB        MOV     DS,BX
076D:000C BA0000      MOV     DX,0000
076D:000F CD21        INT     21
076D:0011 B402        MOV     AH,02
076D:0013 B20A        MOV     DL,0A
076D:0015 CD21        INT     21
076D:0017 B02E        MOV     AL,2E
076D:0019 BB0000      MOV     BX,0000
076D:001C 80BF000065    CMP     BYTE PTR [BX+0000],65
-G=0000
HELLO AANANDI WELCOME TO 8086 MICROPROCESSOR
HOLLO AANANDI WOLCOMD TO 8086 MICROPROCOSSOR
AX=0900 BX=076A CX=0079 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076D IP=003D  NV UP EI PL ZR NA PE NC
076D:003D CC          INT     3
-

```

Output:

Screenshots of the output. (Both strings should be present in your screenshot)

```

-G=0000
HELLO AANANDI WELCOME TO 8086 MICROPROCESSOR
HOLLO AANANDI WOLCOMD TO 8086 MICROPROCOSSOR
AX=0900 BX=076A CX=0079 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076D IP=003D  NV UP EI PL ZR NA PE NC
076D:003D CC          INT     3
-

```