মেট্রোপলিটন ইউনিভার্সিটি
Metropolitan University

Department of Computer Science and Engineering

Course Code: CSE238	Credits: 1.5
Course Name: Microprocessor & Interfacing Lab	Faculty: FRS

Lab 03

Basic I/O, Advanced Arithmetic Operations

Activity Detail:

Discussion: In order to carry out basic I/O operation, function specific values need to be kept in AH register before calling INT 21h which is used for a system call. The function dispatcher maps the value to carry out specific function.

AH = 1; single character input
AH = 2; single character output
AH = 9; string output

i. *Single Key Input:*

```
MOV AH, 1
INT 21h
```

The program will wait for the user to give an input. It will be saved in AL.

ii. *Single Character Output:*

The item to be printed must be in DL

```
MOV DL, 5
MOV AH, 2
INT 21h
```

The hexadecimal values of new line are 0Ah and line feed is 0Dh.

iii. *Displaying a String:*

A string is an array of characters. Previously we have defined variables and arrays.

```
M1 db "a$"
M2 db "hello$"
```

The values of variables and arrays are stored in the data segment of the memory. In order to fetch them we need the offset address (offset + segment). To print a string, we need to store 9 in AH. While 9 is in AH, int 21h expects DX to hold the offset address of the array/variable.

LEA is the name of the instruction that provides the offset address. LEA stands for Load Effective Address. The syntax for using LEA is:

```
LEA destination, source
```

Source is the memory address (the array or the variable) whose offset address will be saved in the destination. The destination is a general register. e.g. for printing the string "hello":

```
M db "hello$"
LEA DX, M
MOV AH, 9
INT 21h
```

Our first program will read a character from the keyboard and display it at the beginning of the next line. We start by displaying a question mark:

```
MOV AH, 2 ; display character function
MOV DL, '?' ; character is '?'
INT 21h ; display character
```

The second instruction moves 3Fh, the ASCII code for "?", into DL.

Next, we read a character:

```
MOV AH, 1 ; read character function
INT 21h ; store character in AL
```

Now we would like to display the character on the next line. Before doing so, the character must be saved in another register. We'll see why in a moment.

MOV BL, AL ; *save it in BL*

To move the cursor to the beginning of the next line, we must execute a carriage return and line feed. We can perform these functions by putting the ASCII codes for them in DL and executing INT 21h.

MOV AH, 2 ; *display character function*

MOV DL, 0DH ; *carriage return*

INT 21h ; *execute carriage return*

MOV DL, 0AH ; *line feed*

INT 21h ; *execute line feed*

The reason why we had to move the input character from AL to BL is that the INT 21h function, changes AL.

Finally, we are ready to display the character:

MOV DL, BL ; *get character*

INT 21h ; *and display it*

Problem Solving

Task 01

Take a character input and display it. Display the message "Please insert a character: " when taking an input.

Task 02

Perform addition/subtraction/division/multiplication by taking inputs from the user. *Note: Display appropriate messages when taking input and showing the output.*

Task 03

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 lower case.

c. Read an uppercase letter, and display it at the next position on the next line in lower case.

Task 04

Write a program to:

a. Read two decimal digits whose sum is less than 10,

b. Display them and their sum on the next line, with an appropriate message.

Sample execution:
27
The Sum of 2 and 7 is 9

Task 05

Write a program to:

a. Prompt the user

b. Read first, middle, and last initials of a person's name, and then

c. Display them down the left margin.

Sample execution:
Enter Three Initials: FRS
F
R
S

Task 06

Write a program to read one of the hex digits A-F, and display it on the next line in decimal.

Sample execution:
Enter a Hex Digit: C
In Decimal it is: 12

Code Template

```
.MODEL SMALL

.STACK 100H

.DATA


.CODE
MAIN PROC

;iniitialize DS

MOV AX,@DATA
MOV DS,AX

;enter your code here









;exit to DOS

MOV AX,4C00H
INT 21H

MAIN ENDP
END MAIN
```