COAL LAB MANUAL

**Submitted to:**

Dr. Tauqir

Teacher assistant : Sir Shahzad

**Submitted By:**

Hamza Farooq

Registeration no : 2016-CS-122

**Section:**

Section B

**Department:**

Department of Computer Science

**Institute:**

University of Engineering and Technology Lahore (Main Campus)



**Date of Submission:** March 29, 2018

**COAL Lab 5**

**Sorting three numbers**

**Description**

In this lab we will learn about

* Defining uninitialized variables
* Sorting numbers by comparing

**Syntax**

**Var1 db ?:** Syntax for defining an uninitialized variable of 1 byte

**Suitable code block labels:** OPENING, BEFORE\_SORTING, START,

REPLACEN1N2, REPLACEN2N3, REPLACEN1N3, OUTPUT,

EXIT.

**Lab Work**  
Write an assembly language program that takes three numbers as input. Sort them by four stages of comparison. Display the numbers in ascending order.

**Program**

.MODEL SMALL

.data

STR1 DB 10, 13, 'Enter 1st Number$'

STR2 DB 10, 13, 'Enter 2nd Number$'

STR3 DB 10, 13, 'Enter 3rd Number$'

STR4 DB 10, 13, 'Do you want to run again?(type y)$'

n1 db ?

n2 db ?

n3 db ?

.code

OPENING:

mov AX, @data

mov DS, AX

jmp BEFORE\_SORTING

BEFORE\_SORTING:

LEA DX, STR1

mov ah, 9

int 21h

mov ah, 1

int 21h

mov n1, al

LEA DX, STR2

mov ah, 9

int 21h

mov ah, 1

int 21h

mov n2, al

LEA DX, STR3

mov ah, 9

int 21h

mov ah, 1

int 21h

mov n3, al

START:

mov bl, n1

CMP bl, n2

JL REPLACEN1N2

mov bl, n2

CMP bl, n3

JL REPLACEN2N3

mov bl, n1

CMP bl, n3

JL REPLACEN1N3

JMP OUTPUT

REPLACEN1N2:

mov bl, n1

mov bh, n2

mov dh, bl

mov bl, bh

mov bh, dh

mov n1, bl

mov n2, bh

JMP START

REPLACEN2N3:

mov bl, n2

mov bh, n3

mov dh, bl

mov bl, bh

mov bh, dh

mov n2, bl

mov n3, bh

JMP START

REPLACEN1N3:

mov bl, n1

mov bh, n3

mov dh, bl

mov bl, bh

mov bh, dh

mov n1, bl

mov n3, bh

JMP START

OUTPUT:

mov ah, 2

mov dl, 0Ah

int 21h

mov dl, 0Dh

int 21h

mov dl, n3

int 21h

mov dl, n2

int 21h

mov dl, n1

int 21h

LEA DX, STR4

mov ah, 9

int 21h

mov ah, 1

int 21h

CMP al, 079h

JNE exit

JMP OPENING

exit:

.exit

**Program 2**

.MODEL SMALL

.data

.code

mov AX, @data

mov DS, AX

mov CX, 9

mov ah, 2

mov bl, 0

START:

add bl, 1

mov dl, bl

add dl, 30h

int 21h

LOOP START

ret