

Department of Computer Science and Engineering

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

Lab 05

Flow control instructions and Looping structures

Discussion:

Just like for and while loops in high level programming languages, loops can also be implemented in assembly. There are 2 ways of implementing a loop: Explicit and Implicit.

Explicit: By using compare and jump instructions to decide whether to enter the loop or not and by using the inc/add/sub instruction for increments.

Java	Assembly
<pre>int x = 0; while (x < 5){ System.out.println(x); x++; }</pre>	<pre>mov ah,2 mov dl,30h ;30h is 0 start: cmp dl,35h ; jge end int 21h inc dl jmp start end:</pre>

Implicit: In this case we do not have to check whether the counter has reached the limit or not. This will be done automatically. The instructions will be `loop`. [*loop destination_line*](#) is the syntax. CX will be used as the counter always in order for the loop instruction to execute.

Java	Assembly
<pre>int x = 0; while (x < 5){ System.out.println(x); x++; }</pre>	<pre>mov cx,5 ;the bound will be in cx.(the number of times the loop will run) mov dl,30h mov ah,2 start: int 21h inc dl loop start NB. CX will always start from the specified count and will always decrement by 1.</pre>

Repeat Until loop: Repeat the loop until a condition is satisfied. For example you have been asked to take in characters from the user and print them until a space is pressed.

```
repeat:
mov ah,1
int 21h
mov ah,2
mov dl,al
int 21h
cmp al,' '
jne repeat
```

Problem Solving

Task 01

Write a count-controlled loop to display a row of 80 stars.

Task 02

Write a sequence of instructions to do each of the following:

- a. Put the sum of $1 + 4 + 7 + \dots + 148$ in AX.
- b. Put the sum $100 + 95 + 90 + \dots + 5$ in AX.

Task 03

Read a five character password and overprint it by executing a carriage return and displaying five X's. You need not store the input characters anywhere.

Task 04

The following algorithm may be used to carry out multiplication of two positive numbers M and N by repeated addition.

Initialize product to 0

REPEAT

add M to product

decrement N

UNTIL N equals 0

Write a sequence of instructions to multiply AX by BX, and put the product in CX. You may ignore the possibility of overflow.

Task 05

Write a program to display the extended ASCII characters (ASCII codes 80h to FFh). Display 10 characters per line, separated by blanks. Stop after the extended characters have been displayed once.

Task 06

Write a program that will prompt the user to enter a hex digit character ("0" ... "9" or "A" ... "F"), display it on the next line in decimal, and ask the user if he or she wants to do it again. If the user types "y" or "Y", the program repeats; If the user types anything else, the program terminates. If the user enters an illegal character, prompt the user to try again.

Sample execution:

```
ENTER A HEX DIGIT: 9
IN DECIMAL IT IS 9
DO YOU WANT TO DO IT AGAIN? y
ENTER A HEX DIGIT: c
ILLEGAL CHARACTER - ENTER 0...9 OR A...F: C
IN DECIMAL IT IS 12
DO YOU WANT TO DO IT AGAIN? N
```

Task 07

[Hard] Write a program that reads a string of capital letters, ending with a carriage return, and displays the longest sequence of consecutive alphabetically increasing capital letters read.

Sample execution:

```
ENTER A STRING OF CAPITAL LETTERS:
FGHADEFGHC
THE LONGEST CONSECUTIVELY INCREASING STRING IS:
DEFGH
```

Code Template

```
.MODEL SMALL

.STACK 100H

.DATA

.CODE
MAIN PROC

;iniitalize DS

MOV AX,@DATA
MOV DS,AX

;enter your code here


;exit to DOS

MOV AX,4C00H
INT 21H

MAIN ENDP
END MAIN
```