**PMAS Arid Agriculture University Rawalpindi**

**University Institute of Information Technology**

**LAB MANUAL - IX**

**Class/Program: BS (CS)** **Course: COAL (CS530)**



**Objectives:**

1. **More examples using LOOP statement**
2. **How to print a table of a user entered Number**
3. **How to run a loop in forward order using LOOP statement**
4. **How to solve different expressions e.g. n! = n\*(n-1)!**
5. **How to make programs optimized**
6. **Fibonacci Series Algorithm**

**More examples using LOOP Command:**

**How to run a LOOP in Forward order:**

**Previously we have seen that LOOP iterations depends on the number stored in the ecx register. While it runs in the reverse direction and subtract after each successful iteration, it is not possible to run the forward direction as no built-in strategy is available. But we can create a LOOP that runs in the forward direction using another register. Actually the LOOP is still running in the backward direction but the extra register helps to use the values from 0 or 1. The following program will help to understand it in better way.**

**Write an assembly program that calculates the factorial of a user entered number using LOOP command.**

**org 100h**

**.DATA**

**ANS DB ?**

**.CODE**

**MAIN PROC**

**MOV AX,@DATA**

**MOV DS,AX**

**MOV AL,5**

**MOV CL,4**

**MOV BL,AL**

**SUB BL,1**

**L:**

**MUL BL**

**SUB BL,1**

**LOOP L**

**MOV ANS,AL**

**END MAIN**

ret

**Here declaring ebx with one and using it for multiplication instead of ecx register help to start loop variable in the forward direction.**

**Write an assembly program that prints the table of a user entered number.**

**.model small**

**.stack 100h**

**.data**

**num db ?**

**msg db 'Enter a digit:$'**

**string db 5 dup(?)**

**.code**

**main proc**

**mov ax, @data**

**mov ds, ax**

**lea dx, msg ;; display enter a digit:**

**mov ah, 9**

**int 21h**

**mov ah, 1 ;; gets a digit as character**

**int 21h**

**sub al, 48 ;; convert character into number**

**mov num, al**

**xor bx, bx**

**mov bl, 1**

**mov cx, 10**

**top:mov al, num ;; multiplication step**

**mul bl**

**push bx ;; reserving bx and cx for later use by pushin stack**

**push cx**

**lea si, string**

**mov bx, 10 ;; storing 2 digits number as character in string**

**xor cx, cx**

**division: xor dx,dx**

**div bx**

**push dx**

**inc cx**

**cmp ax, 0**

**jne division**

**store: pop dx**

**add dl, 48**

**mov [si], dl**

**inc si**

**loop store**

**mov [si], '$ '**

**pop cx**

**pop bx**

**mov dl, 13 ;;; new line**

**mov ah, 2**

**int 21h**

**mov dl, 10**

**mov ah, 2**

**int 21h**

**lea dx, string ;;get that 2 digit number which is character from string**

**mov ah, 9**

**int 21h**

**inc bl**

**loop top ;; multiplication table stops**

**mov ah, 4ch ;; end of program**

**int 21h**

**main endp**

**end**

**How to solve equations in Assembly Language:**

**We can also solve different equation in assembly language like finding area, perimeter, circumference, different conversions and equations etc.**

**Q.Write an assembly program that proves that n! = n\*(n-1)!.**

**Fibonacci Series:**

**Q.Write an assembly program that prints the Fibonacci series elements.**