**Practical - 25**

**Aim:** Write an assembly language program to fill the memory locations starting from 3000h, with n Fibonacci numbers.

**Description of instructions used:**

**DB** **(DEFINE BYTE):** The **DB** directive is used to declare a byte type variable, or a set aside one or more storage locations of type byte in memory.

**LEA (Load Effective Address):** LEA and MOV both are same but in that there are quite difference between both of them.

* LEA means Load Effective Address
* MOV means Load Value

In short, LEA loads a pointer to the item you're addressing whereas MOV loads the actual value at that address. The purpose of LEA is to allow one to perform a non-trivial address calculation and store the result.

**LEA AX, [BP+SI+5]; Compute address of value**

**MOV AX, [BP+SI+5]; Load value at that address**

**LOOP:** Used to loop a group of instructions until the condition satisfies, i.e., CX = 0

**INT:** Used to interrupt the program during execution and calling service specified.

**INC:** Used to increment the provided byte/word by 1.

**ADD:** Used to add the provided byte to byte/word to word.

**Code:**

.MODEL SMALL

.DATA

RES DB ?

CNT DB 0AH ; Initialize the counter for the no of Fibonacci No needed

.CODE

START:

MOV AX,3000H

MOV DS,AX

LEA SI,RES

MOV CL,CNT ; Load the count value for CL for looping

MOV AX,00H ; Default No

MOV BX,01H ; Default No

;Fibonacci Part

**L1:**

**ADD AX,BX**

**MOV [SI],AX**

**MOV AX,BX**

**MOV BX,[SI]**

**INC SI**

**LOOP L1**

**INT 3H** ; Terminate the Program

**END START**

**Output:**

