

Fall Semester 2021-2022
Microprocessor and Interfacing
Lab Report

Digital Assignment-5

Experiment No: 6

Task No: 5

Course Code: CSE2006

Slot: L7+L8



Submitted By: Alokam Nikhitha

Reg. Numb: 19BCE2555

Submitted To: Dr. Abdul Majed KK

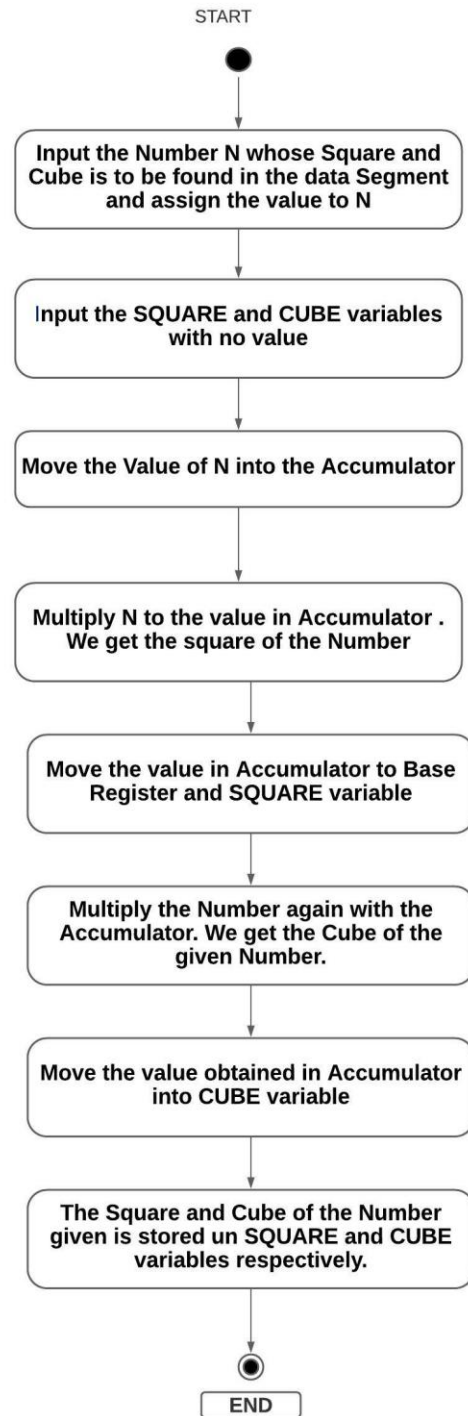
Experiment 6:

Question 1) Write an ALP program to find square and cube of a given number

ALGORITHM

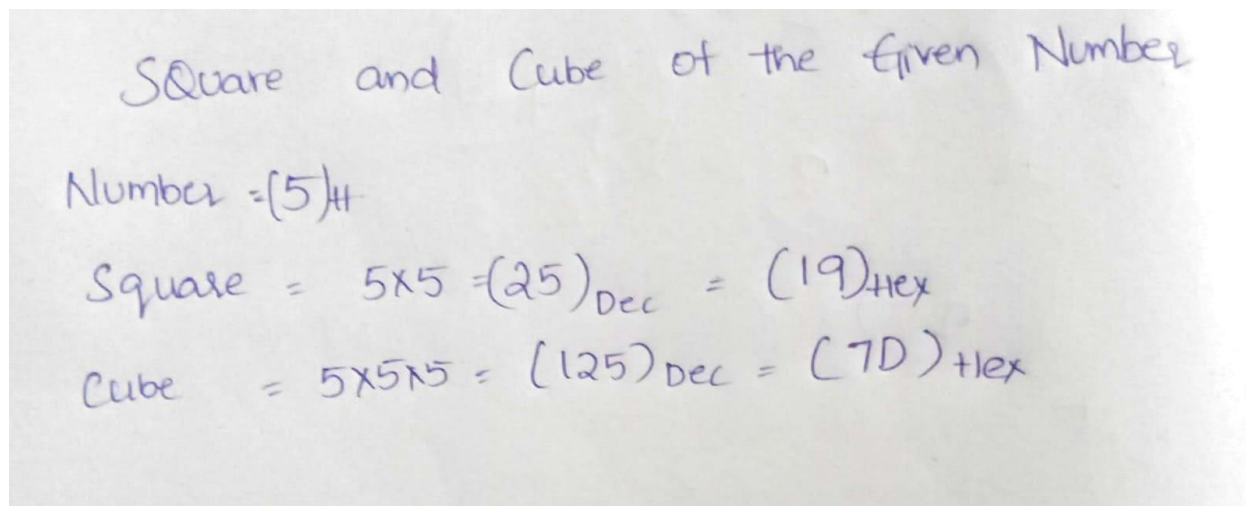
- **Input the Number N whose Square and Cube is to be found in the data Segment and assign the value to N**
- **Input the SQUARE and CUBE variables with no value**
- **Move the Value of N into the Accumulator**
- **Multiply N to the value in Accumulator . We get the square of the Number**
- **Move the value in Accumulator to Base Register and SQUARE variable**
- **Multiply the Number again with the Accumulator. We get the Cube of the given Number.**
- **Move the value obtained in Accumulator into CUBE variable**
- **The Square and Cube of the Number given is stored in SQUARE and CUBE variables respectively.**

FLOWCHART



Design and Calculations:

Input the Number N whose Square and Cube is to be found in the data Segment and assign the value to N. Input the SQUARE and CUBE variables with no value. Move the Value of N into the Accumulator. Multiply N to the value in Accumulator . We get the square of the Number. Move the value in Accumulator to Base Register and SQUARE variable. Multiply the Number again with the Accumulator. We get the Cube of the given Number. Move the value obtained in Accumulator into CUBE variable. The Square and Cube of the Number given is stored in SQUARE and CUBE variables respectively.



Handwritten calculations showing the square and cube of the number 5:

Square and Cube of the given Number

Number = (5)₁₀

Square = $5 \times 5 = (25)_{\text{Dec}} = (19)_{\text{Hex}}$

Cube = $5 \times 5 \times 5 = (125)_{\text{Dec}} = (7D)_{\text{Hex}}$

Program Code:

DATA SEGMENT

A DW 5H

SQUARE DW ?

CUBE DW ?

DATA ENDS

CODE SEGMENT

ASSUME DS:DATA,CS:CODE

START:

MOV AX,DATA

MOV DS,AX

MOV AX,A

MUL A

MOV SQUARE, AX

MOV BX, AX

MUL A

MOV CUBE, AX

INT 21H

CODE ENDS

END START

```

01 DATA SEGMENT
02 A DW 5H
03 SQUARE DW ?
04 CUBE DW ?
05 DATA ENDS
06 CODE SEGMENT
07     ASSUME DS:DATA,CS:CODE
08 START:
09     MOV AX,DATA
10     MOV DS,AX
11     MOV AX,A
12     MUL A
13     MOV SQUARE, AX
14     MOV BX, AX
15     MUL A
16     MOV CUBE, AX
17     INT 21H
18 CODE ENDS
19 END START

```

Output:

The screenshot displays three windows from an x86 emulator:

- variables**: Shows the memory addresses and values for variables. SQUARE is at 0005h with value 0019h. CUBE is at 0019h with value 007Dh.
- original source code**: Shows the assembly code. The instruction `INT 21H` at line 17 is highlighted in yellow.
- emulator: Exp6 Q1.exe**: Shows the CPU registers and memory dump. The registers window shows:

Register	H	L
AX	00	7D
BX	00	19
CX	00	2A
DX	00	00
CS	0711	
IP	002E	
SS	0710	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0710	
ES	0700	

 The memory dump shows the instruction at address 0711:002E: `HLT`.

Result and Inference:

The value in the Accumulator is the Cube of the number given and Square of the number is stored in Counter Register. We can see that SQUARE(0019H) and CUBE (007DH) variables are filled with values of square and cube of the number(0005H) given

Question 2)

If Reg. No is 20BCE0043 then address location of Data is 2043 and Data is 43(decimal).

2043	0	0	1	0	1	0	1	1
2044	0	0	0	0	0	0	1	1
2045	0	0	0	0	0	1	0	1

❖ Use your Reg. No for Datas.

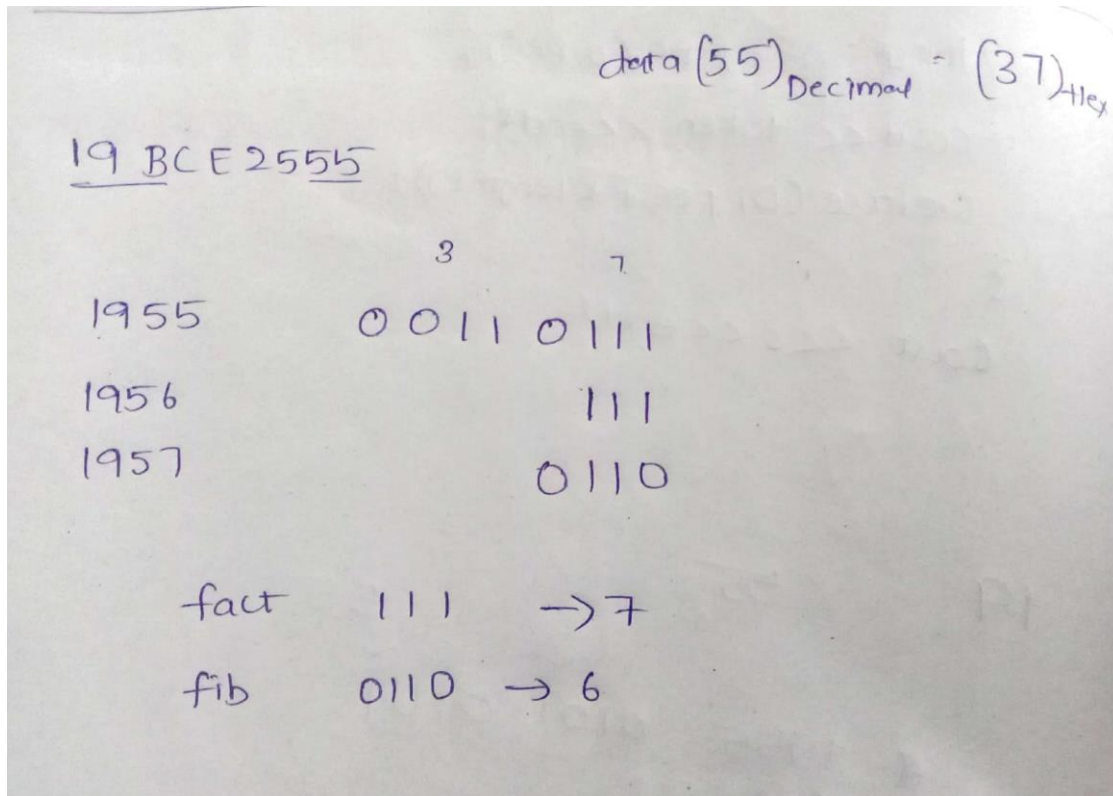
2. A Find the factorial (last three bits 011 = 3) $\rightarrow 1 \times 2 \times 3 = 6$

2. B Find the fibonacci series (0101 = 5) $\rightarrow 0, 1, 1, 2, 3, 5, 8,$

Documents should be in the file

1. Flow chart of 2.A and 2.B
2. Handwritten Assembly Language program (ALP) of 2.A to 2.B
3. Snapshot of Typed ALP of 2.A to 2.B
4. Snapshot of Output and status of Flag register.

INITIAL CALCULATIONS WITH REGISTRATION NUMBER:



Factorial Input: 7

Fibonacci Input: 6

ALGORITHM

Factorial:

- **Move the value of whose factorial to be found in to the Accumulator from the given Location.**
- **Move the value in Accumulator to both Base Register and Counter Register.**
- **Decrement both Base Register and Counter Register**
- **Multiply the Base Register to Accumulator and decrement the Base Register inside a loop**
- **Loop is repeated and Base register is multiplied to Accumulator till the counter becomes 0.**
- **The Factorial of the Given value is stored in Accumulator.**

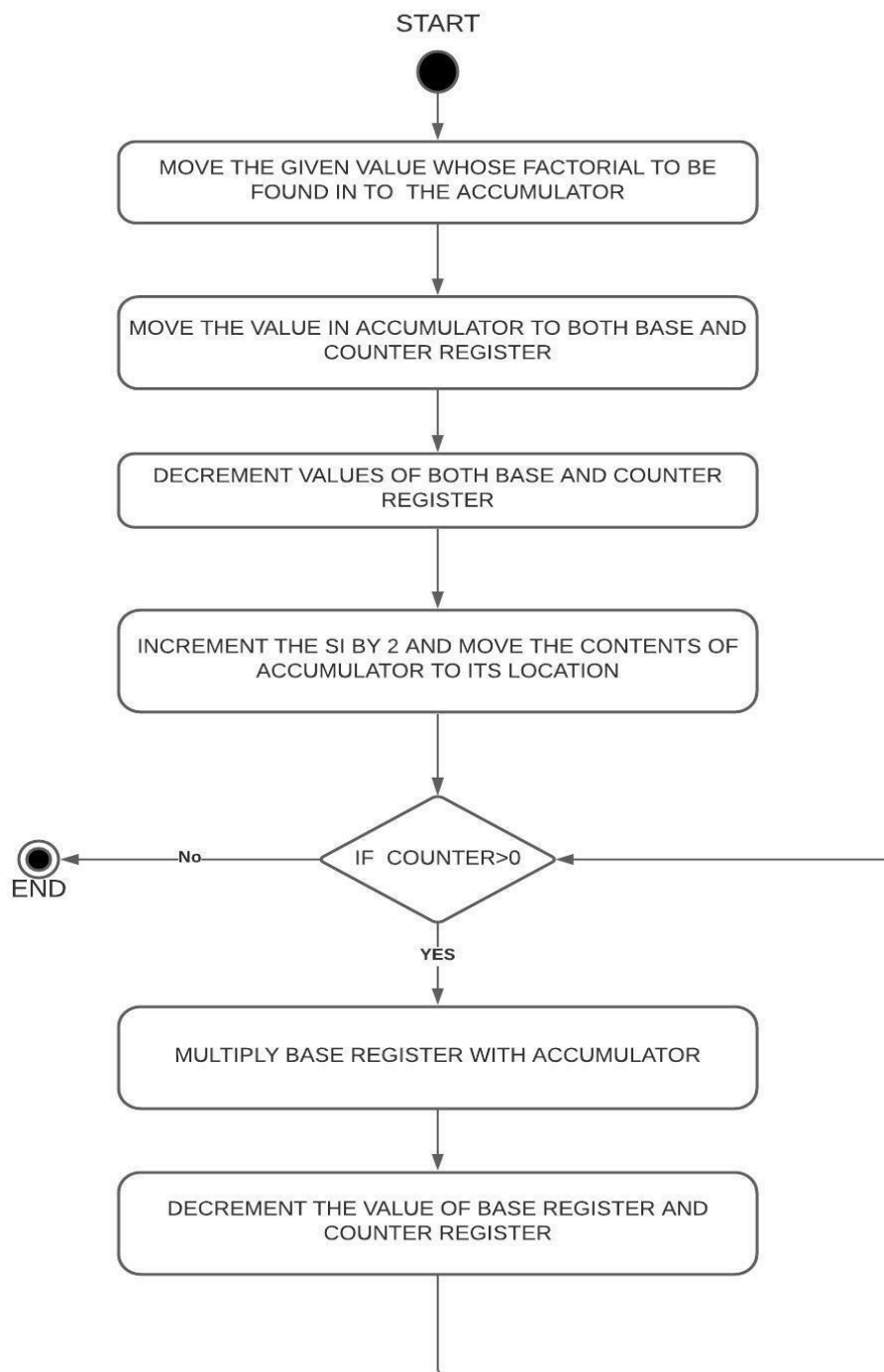
Fibonacci:

- **Input the value of the whose Fibonacci series to be found from the given Location.**
- **Move 500h to SI and input 00h to that location([SI])**
- **Increment the SI value and input 01h to it**
- **Move the N value into counter**
- **Run a loop if counter >0**
- **Move the value at SI-1 location to Accumulator**
- **Add the Value at Location SI to the Accumulator**
- **Increment the SI value**
- **Move the value in Accumulator to location of SI**
- **Decrement the counter**

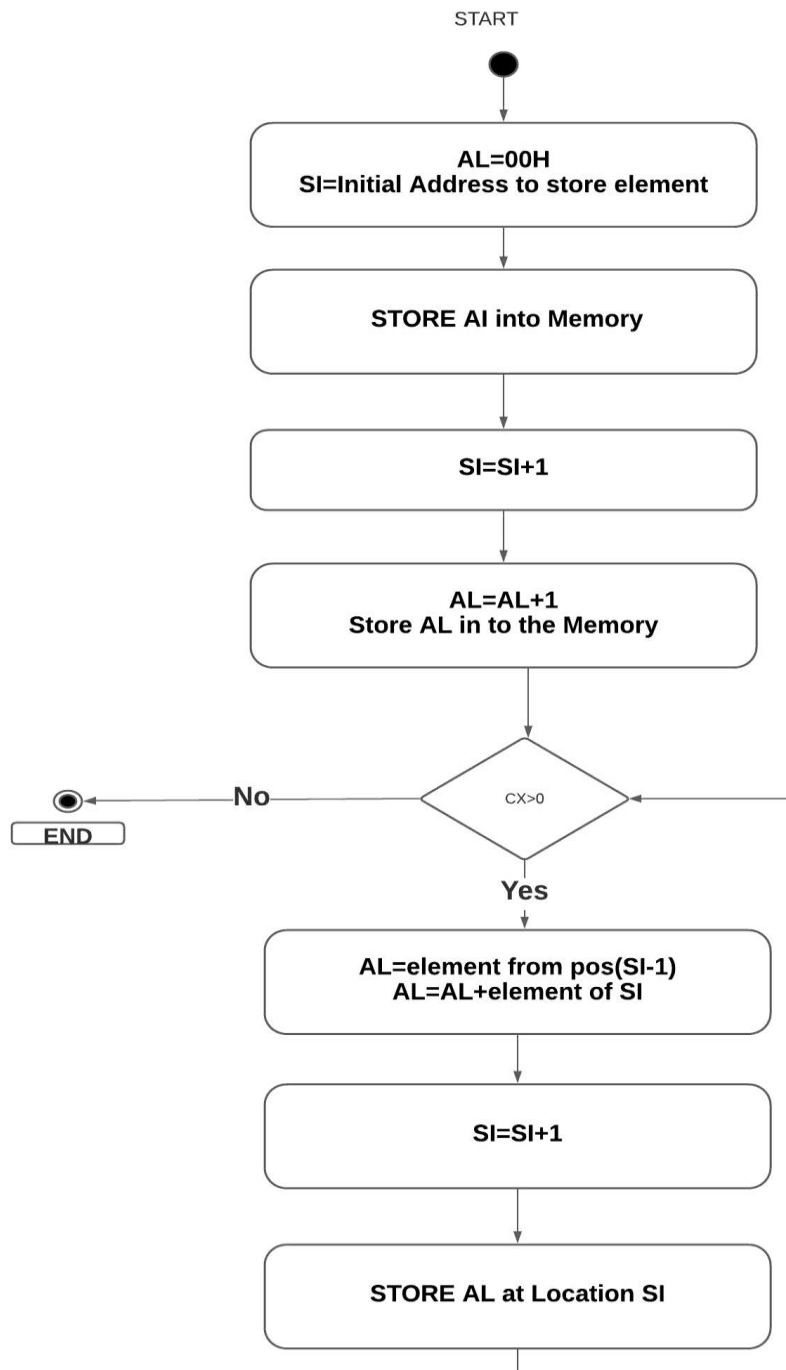
➤ Repeat Loop

FLOWCHART

Factorial:



Fibanocci:



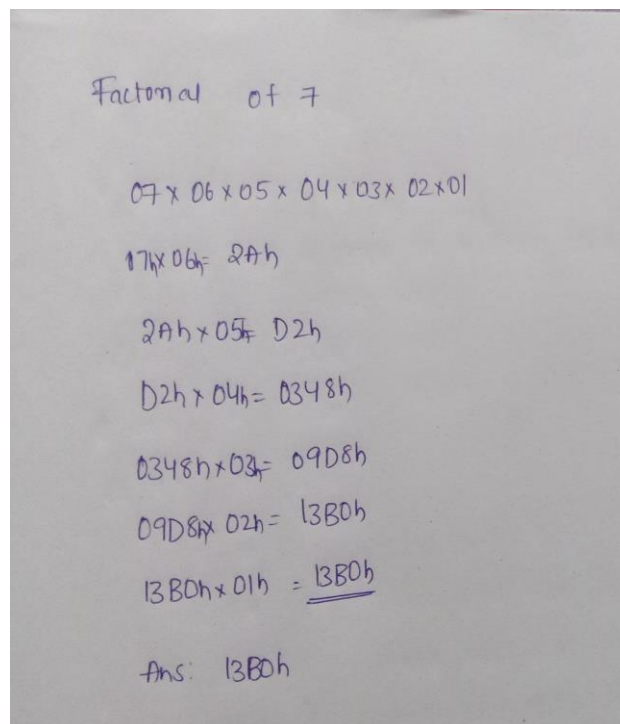
Factorial

Design and Calculations:

Here we input the value from location 1956h in to the accumulator and Move the value from Accumulator to Base Register and Counter Register and Decrement the value of both Base and Counter registers and run a loop and Multiply the Base register . And decrement the Base Register and loop is repeated until the counter becomes 0

Calculations:

$$7*6*5*4*3*2*1=13B0[\text{Hex}] (5040[\text{Dec}])$$



Handwritten calculation of 7 factorial in hexadecimal:

Factorial of 7

$$07 \times 06 \times 05 \times 04 \times 03 \times 02 \times 01$$
$$17h \times 06h = 2Ah$$
$$2Ah \times 05h = D2h$$
$$D2h \times 04h = 0348h$$
$$0348h \times 03h = 09D8h$$
$$09D8h \times 02h = 13B0h$$
$$13B0h \times 01h = \underline{13B0h}$$

Ans: 13B0h

Program Code:

Assume CS: Code DS: Data

DATA SEGMENT

N DW 1956h

Fact DW ?

DATA ENDS

CODE SEGMENT

START:

MOV AX, @DATA

MOV DS, AX

MOV SI, N

MOV AX, [SI]

MOV CX, AX

DEC CX

MOV BX, AX

DEC BX

L:

MUL BX

DEC BX

LOOP L

MOV Fact, AX

CODE ENDS

END START

```

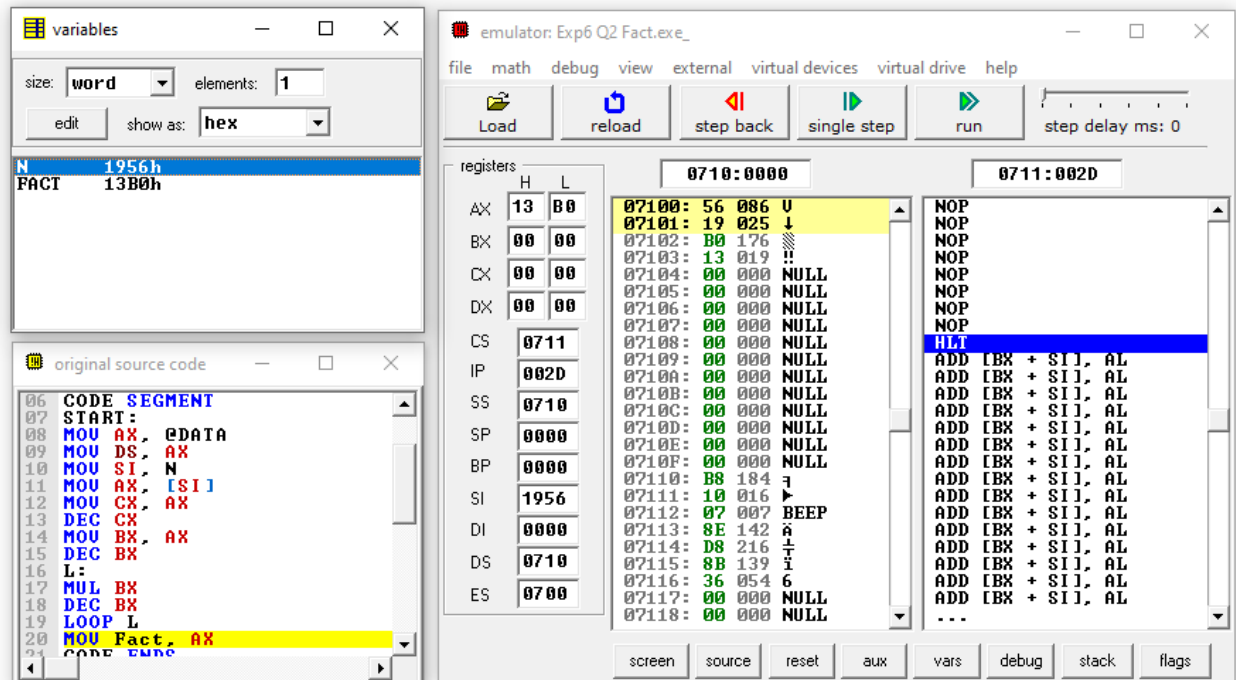
01 Assume CS: Code DS: Data
02 DATA SEGMENT
03     N DW 1956h
04     Fact DW ?
05 DATA ENDS
06 CODE SEGMENT
07     START:
08     MOV AX, @DATA
09     MOV DS, AX
10     MOV SI, N
11     MOV AX, [SI]
12     MOV CX, AX
13     DEC CX
14     MOV BX, AX
15     DEC BX
16     L:
17     MUL BX
18     DEC BX
19     LOOP L
20     MOV Fact, AX
21 CODE ENDS
22 END START
23

```

OUTPUT:

Memory Location:

[illegible]



Result and Inference:

- The Factorial of 7 (5040[dec]-> 13B0[hex])
13B0 is stored in Accumulator and in Fact Variable.
- Both Base and Counter Registers becomes 0.

Fibonacci:

Design and Calculations:

We can observe that DS=0710h from fig1. Then we collected data from Memory location 1957h. To generate Fibonacci sequence, we are putting the 00H and 01H into memory at first. Then we are taking the limit from

location offset 500. We input the N value in to the counter and. Now we are taking number from previous location, then add it with the value of current location, after that storing the result into next location, till the counter becomes 0

Fibonacci Series

Input : 6

Initiat. Values	0	1	2	3	5	8	13
		1	2	3	4	5	6

Program Code:

ASSUME DS:DATA,CS:CODE

DATA SEGMENT

N DW 1957h

FIB DB ?

DATA ENDS

CODE SEGMENT

START:

```
MOV AX,DATA
MOV DS,AX
MOV AX, N
MOV SI, AX
MOV CX, [SI]
MOV AL, 00H
MOV SI, 500H
MOV [SI], AL
INC SI
ADD AL, 01H
MOV [SI], AL
L1: MOV AL, [SI-1]
ADD AL, [SI]
INC SI
MOV [SI], AL
LOOP L1
MOV FIB, AL
INT 21H
CODE ENDS
```

```
END START
```



```

01 ASSUME DS:DATA,CS:CODE
02
03 DATA SEGMENT
04     N DW 1957h
05     FIB DB ?
06 DATA ENDS
07
08 CODE SEGMENT
09     START:
10         MOV AX,DATA
11         MOV DS,AX
12         MOV AX,N
13         MOV SI,AX
14         MOV CX,[SI]
15         MOV AL,00H
16         MOV SI,500H
17         MOV [SI],AL
18         INC SI
19         ADD AL,01H
20         MOV [SI],AL
21     L1: MOV AL,[SI-1]
22         ADD AL,[SI]
23         INC SI
24         MOV [SI],AL
25         LOOP L1
26         MOV FIB,AL
27         INT 21H
28     CODE ENDS
29
30 END START
31

```

Output:

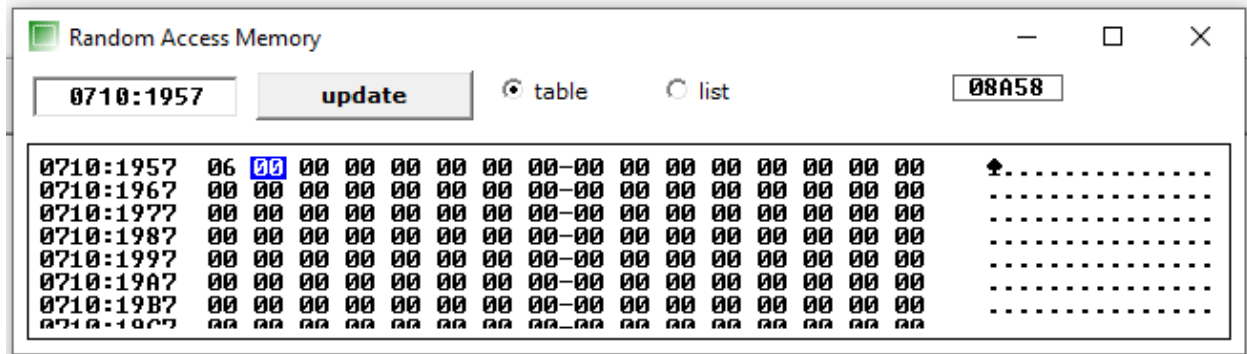
The screenshot displays a DOS emulator interface with three main windows:

- variables:** Shows the variable `N` with value `1957h` and `FIB` with value `00h`. The size is set to `byte` and elements to `1`.
- original source code:** Shows the assembly code with line 28 (`INT 21H`) highlighted in yellow.
- emulator: Exp6 Q2 Fib.exe_:** Shows the register values and memory dump. The registers window shows:

Register	H	L
AX	00	00
BX	00	00
CX	00	00
DX	00	00
CS	F400	
IP	0204	
SS	0710	
SP	FFFA	
BP	0000	
SI	0507	
DI	0000	
DS	0710	
ES	0700	

 The memory dump shows the BIOS interrupt vector table starting at `F400:0204`, with `INT 021h` at `F4204:CF 207` highlighted.

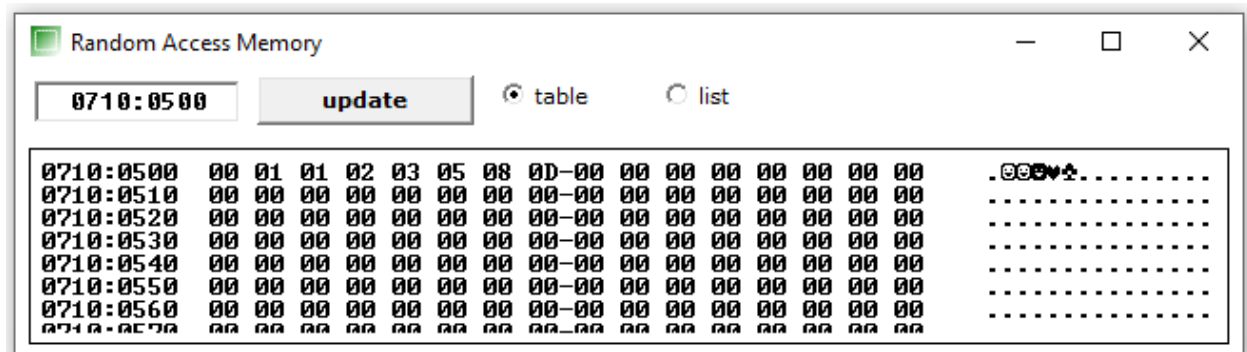
Memory Location: BEFORE (INPUT):



The screenshot shows a 'Random Access Memory' window. The address field contains '0710:1957', the 'update' button is highlighted, and the 'table' radio button is selected. The address '08A58' is shown in the top right. The memory table displays addresses from 0710:1957 to 0710:19B7, with all data fields set to '00'.

Address	06	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00
0710:1957	06	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00
0710:1967	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00
0710:1977	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00
0710:1987	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00
0710:1997	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00
0710:19A7	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00
0710:19B7	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00

AFTER:



The screenshot shows the 'Random Access Memory' window after input. The address field now contains '0710:0500', the 'update' button is highlighted, and the 'table' radio button is selected. The address '08A58' is still in the top right. The memory table displays addresses from 0710:0500 to 0710:0570. The data at 0710:0500 is '00 01 01 02 03 05 08 0D', while all other addresses contain '00'.

Address	00	01	01	02	03	05	08	0D-00	00	00	00	00	00	00	00	00
0710:0500	00	01	01	02	03	05	08	0D-00	00	00	00	00	00	00	00	00
0710:0510	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00
0710:0520	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00
0710:0530	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00
0710:0540	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00
0710:0550	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00
0710:0560	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00
0710:0570	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00

Result and Inference:

We can observe that DS=0710h.

Then we collected data from Memory location 1957h

We have the list of Fibonacci series for N=6 we have DS=0710 and they are located from 0710:0500 to 0710:0507(00,01,01,02,03,05,08,0D)