

Microprocessor and Interfacing CSE2006

Lab Assignment 2

Slot: L3+L4

Name: Kulvir Singh

Register Number: 19BCE2074

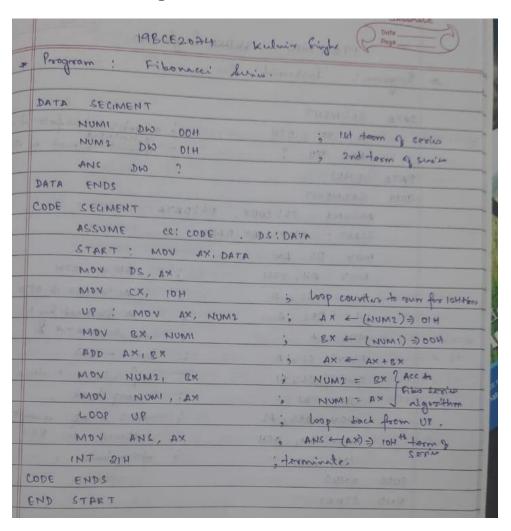
1)Fibonacci Series

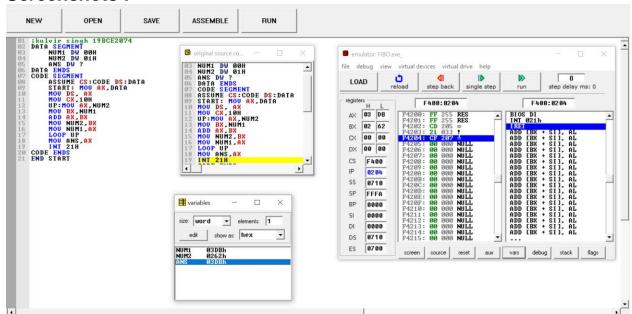
Aim:

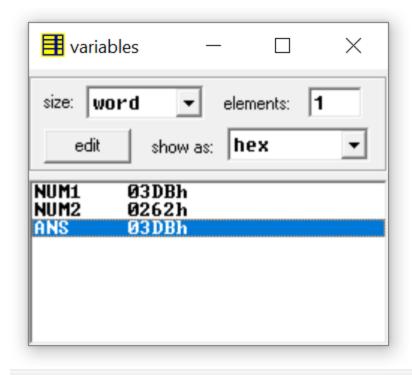
Write a program in 8086 Assembly Language to get the Fibonacci Series term

Requirements:

8086 EMU - An emulator to run the 8086 Assembly Language Code Operating System - Any valid operating system that can execute the emulator







Inference:

The program can successfully get the Fibonacci series terms as seen in the output screenshots

2)Factorial

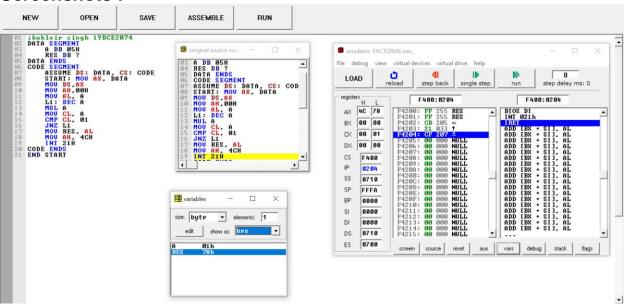
Aim:

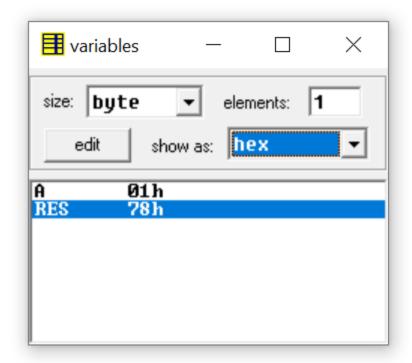
Write a program in 8086 Assembly Language to get the Factorial of a number

Requirements:

8086 EMU - An emulator to run the 8086 Assembly Language Code Operating System - Any valid operating system that can execute the emulator

	198652044	Kelvik Light Page
>		
	DATA SEGMENT	The state of the s
- Charles	A DE 05H	is mumber whose factorial or
	RES DB ?	calculated in this
	DATA ENDS	100 Hotel
	CODE SECMENT	SCHOOL AVAIL
	ASSUME CS! CODE	DS!DATA
	START: MON AX,	DATA
		AT MA MORE I WHAT
	MOV AH, OTH	; AH - 07H
2000	MOV AL, A) AL -(A) > 05H
Hite	LI: DEC A	is A reduced by 1
1000	MULA	AX + BAX A
	MOV CL, A	; CL (A)
- 1	CMP CL, 01	
2000	JNZ LI	is Loop back,
4 10	MOV RES, AL	
7-114	NOV AH, ACH	; RES & AL moult ; stored in RES
	INT 21H	; terminte.
1	CODE ENDS	The state of the s
	END START	20hr 1 20hr 3





Inference:

The program stores 120 as the factorial of 5 in res variable and hence is accurate.

3)Square and Cube

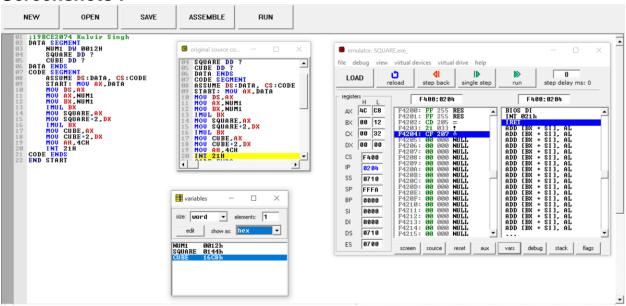
Aim:

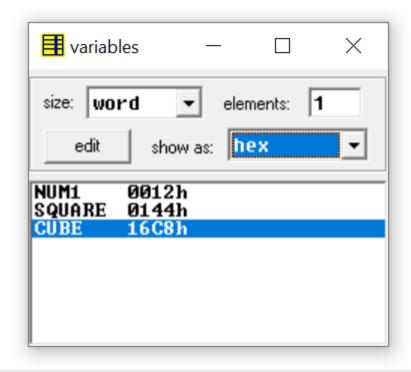
Write a program in 8086 Assembly Language to get the Square and Cube of a number

Requirements:

8086 EMU - An emulator to run the 8086 Assembly Language Code Operating System - Any valid operating system that can execute the emulator

A SECIME NUM! SRUAFE CUBE ENDS SECIME ASSUME START MOV MOV	DW OO ? OO ? THE ! SE ! DAT ! SE ! DAT ! DAT ! DAT ! DAT ! DAT ! DE ! D	12 H A 60 A× 1	CS! CA	Stone	cube of	13000
A SECIME NUM! SRUAFE CUBE ENDS SECIME ASSUME START MOV MOV	DD ? DD ? NT DD ? NT S DS! DAT : MOV DS, RO	12 H A 60 A× 1	CS! CA	Stone	cube of	Misery Misery
NUMI SRUARE CUBE ENDS SECIME ASSUME START MOV MOV	DW OO ? OO ? THE ! SE ! DAT ! SE ! DAT ! DAT ! DAT ! DAT ! DAT ! DE ! D	A CO	CS. CATA	Stone	cube of	Misery Misery
SEVARE CUBE ENDS SECIME ASSUME START MOV MOV MOV	TO T	A CO	CS. CATA	Stone	cube of	Misery Misery
SEVARE CUBE ENDS SECIME ASSUME START MOV MOV MOV	TO T	A CO	CS. CATA	Stone	cube of	Misery Misery
CUBE ENDS SECIMENT ASSUME START MOV MOV MOV	TO 7 THE SE STATE TO VOM TO SE	00 A) 00 , 17 A	cs: co	Stone	cube of	Misery Misery
SECIMENT SECUMENTS TART MOV MOV MOV	NT DS! DAT : MOV : MOV DS, RM	A×	CS! CA	ope	ente of	13000
SECIME ASSUME START MOV MOV	DS! DA! DA! MOV : MOV . DS . DED	A×	CS! CA	ope 1	12 101 12 101 13 101 13 101 13 101	
START NOV MOV	DS! DA! DA! MOV : MOV . DS . DED	A×	CS! CA	ope 1	Saction .	
MOV	DS, Rea	A×	TA PO	200	Saction .	
MOV	DS, Rea	A×	TA PO	200	Saction .	
MOV	AX, NUM	1	1 Alain		·	
10100	EX, NUM	1	1 della	.0.5	the former	la me
10100	IST, NUA			71.0	- Grown	13) 124
IMOL		W.I	E I I	1 25	t 4 hours	
	LX		10000	DXAX	4- 12-14 10	N - 1011 4 10
1010	- MAKE	, AX	194	Lower	Rule of San	unce by are
MOV	SOU ARE +	2 , DX	,	Higher	Byte of sa	mark to DX.
IMUL	LX	Maga	3	DX AX	← CARD and	A) = Cole
MON	CURE, AX	1	,	Lower	Byke A	cube
MON	CUBE+2	, 0%	1015	high	- byt of	ende
MOV	AH , 4CH		162.10	torry	mination.	
					0	
100	XI H	U.S.	4-74 3	24537	22.2	
1						1 3 1 7
	MON	MOV CUBE, AX MOV CUBE+2 MOV AH, 4CH NOVEN INT RIH ENDS	MON CURE, AX MON CURE+2, DX MON AH, 4CH NOTEN INT RIH ENDS	MOV CURE, AX ; MOV CURE+2, DX ; MOV AH, 4CH , NOTEN INT RIH ENDS	MON CUBE, AX , Lowers MON CUBE+2, DX ; higher MON AH, 4CH , Jenn INT 21H ENDS	ENDS





Inference:

The program stores the accurate square and cube values of 0012H in the respective variables

4)GCD – greatest common divisor

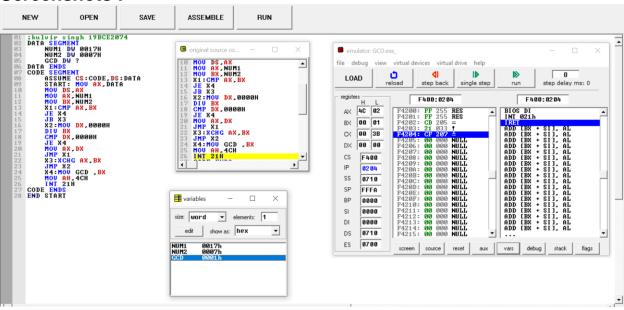
Aim:

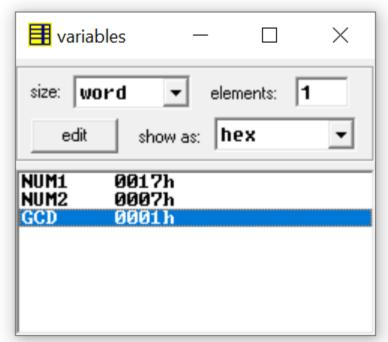
Write a program in 8086 Assembly Language to get the GCD of a number

Requirements:

8086 EMU - An emulator to run the 8086 Assembly Language Code Operating System - Any valid operating system that can execute the emulator

100	19BCE 20AU Kulut Righ						
	Program: GCD. & 2 norman						
	DATA SEGMENT						
Balls 199	NOW DW COIRM , CKO of STATION						
- Third	hts 600381 9						
- Course	CICD DW .						
	DATA ENDS						
	CODE SEGMENT						
	ASSUME & CS ! CODE , DS ! DATA						
	START : MON ANY DATA						
	MOV DE, AX						
	MOV AX, NUMI ; AX - (NUM) & OPICH						
	MOV BY, NUMP , BX + (NUME) 9 MOZH						
	- XI: EMP AXIBX ; compare Ax 2 Ex						
78 10 1	JE X4 12 of excel act X4						
20.00	JB X3 if 1900 then cale vs						
	X2: MON DX, DOODH; Clear DX						
26.0	DIN BX DXAX + DXAX - DX						
-	CMP DX, DODOH ; Compare remainder with						
	It enul ante yu						
	MOV AX, DX 1 AX C DX						
-	JW6 XI						
	CHU AX, BX						
	JME X2						
	CICD, RX . C.C.						
	INT DIU						
	CODE ENDS , terminate						





Inference:

The program stores the accurate GCD 23 and 7 in GCD variable.

.....

5)LCM - least common divisor

Aim:

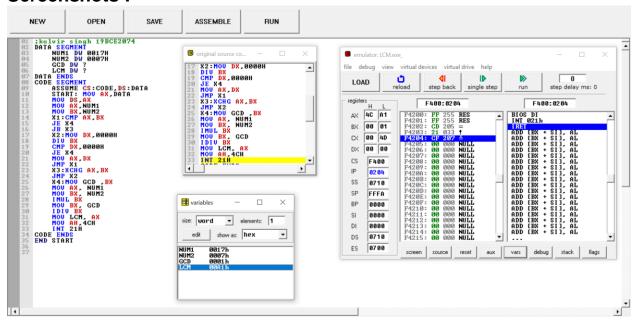
Write a program in 8086 Assembly Language to get the LCM of a number

Requirements:

8086 EMU - An emulator to run the 8086 Assembly Language Code Operating System - Any valid operating system that can execute the emulator

19BCE2074 KWW.	r dingh Piege
0	numbers LCM = NI * NZ
THE CONTRACT OF SALES	ALL CONTRACTOR OF THE PARTY OF
DATA SECIMENT	H.S. 10300
HEIDO WE IMUN	
NAMS DM 0005H	my write
GCD DID ?	5 Variable to store aco
LCW DW ?	is variable to store Lem
DATA ENDS	1600 1960
CODE SECIMENT	2000 V.100
ASSUME CS! CODE , DS	DATA LIATA COM
START ! MOV AX, DATA	
MOV DS,AX	
MON AX, NUM	HFIPE ((MOW) + XA ;
MOV EX, NUM2	8x 4 (NOM2) 5) 5087H
XI: CMP AX, BX	; compare AX 2 EX
26 ×4	; if equal gots ×4
₹ ×3	; if less then goto X3
X2: NOV DX, 0000H	is cluse tox
DIV BX	; DXAX COXAX : BX
CMP DX , 0000H	; compare remainder with 0
J€ X4	; if equal gots ×4
	; Ax < DX
MOV AX, DX	, goto XI
JMP KI,	; exchange AX , BX
X3: XCHQ AX 18X	, goto X2
JMP X2	; GCD - EX
X 4: MOV GCD, BX	

```
AX + (NUMD ) DOISE
                             HERON ( GMUKA) & X 8
          AX, NUM
     NIOV
          BX , NUM?
     MON
                            DY AN - AN AN EN
     IMUL BX
                               EX + 000
           BX, GLD
     MON
     IDIN
           RK
     MOV
           LCM, AX
                           terminates
     MOV
          AH, DYCH
     INT DIH
CODE
    ENDS
END START
```





Inference:

The program stores the accurate LCM 23 and 7 in GCD variable.
