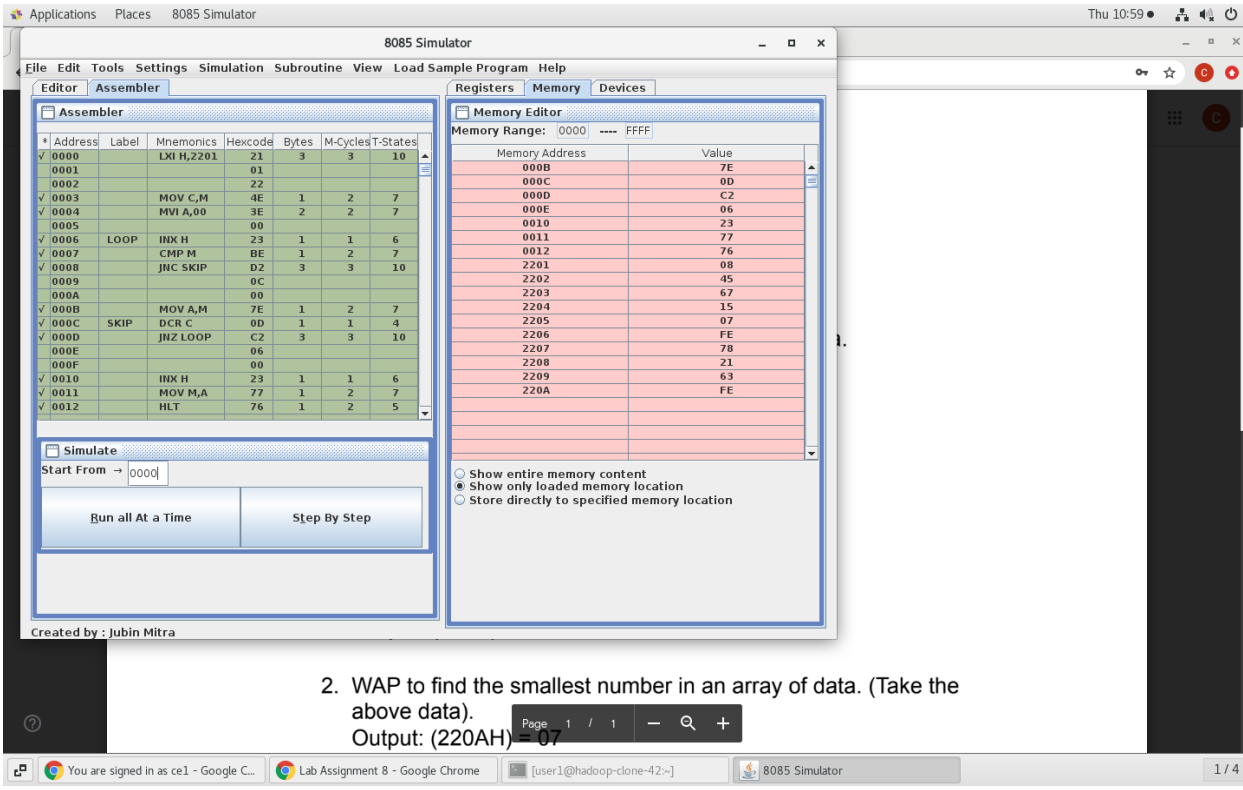
CSA

1. WAP to find the largest number in an array of data.

Input : (2201H) = 08 (Array Size)
(2202H) = 45
(2203H) = 67
(2204H) = 15
(2205H) = 07
(2206H) = FE
(2207H) = 78
(2208H) = 21
(2209H) = 63

Output: (220A) = FE

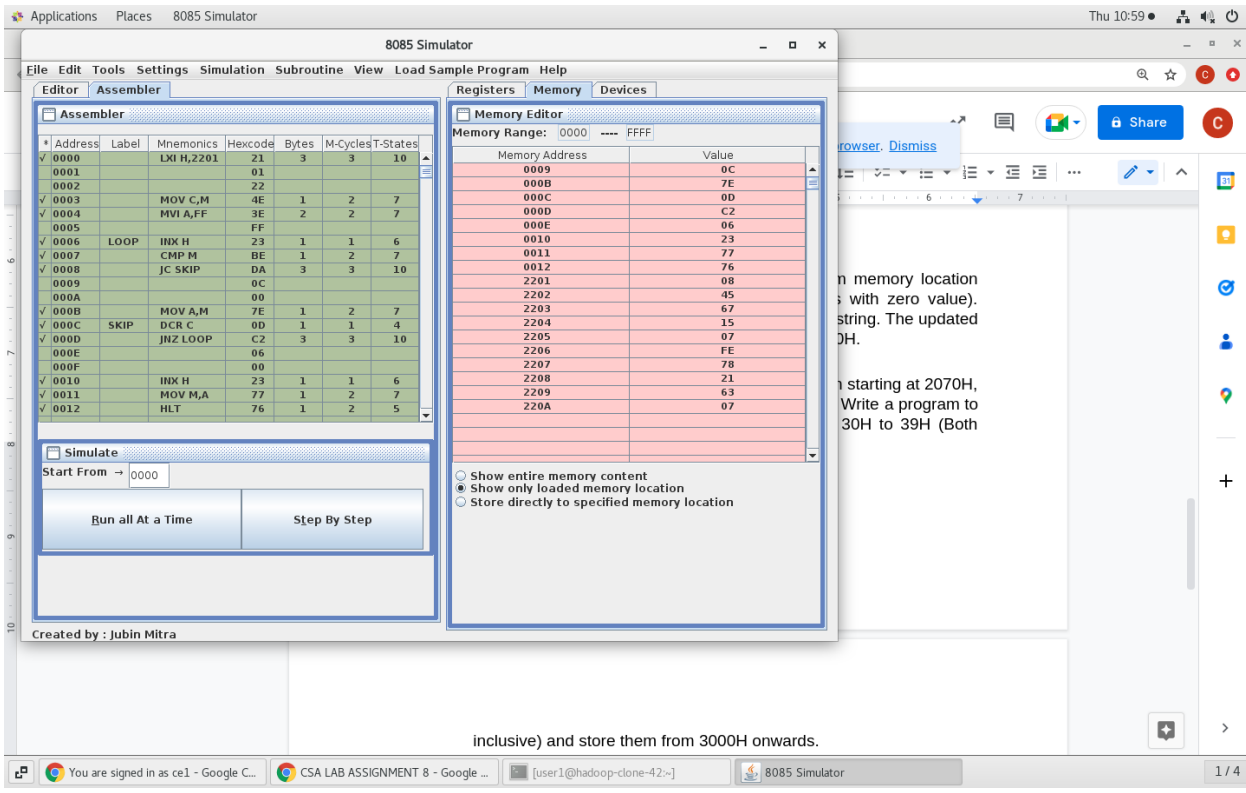
```
LXI H, 2201H
MOV C,M
MVI A,00H

loop:
INX H
CMP M
JNC skip
MOV A,M
skip:DCR C
JNZ loop

INX H
MOV M,A

HLT

#ORG 2201H
#DB 08,45,67,15,07,FE,78,21,63
```



2. WAP to find the smallest number in an array of data. (Take the above data).
Output: (220AH) = 07

```
LXI H, 2201H
MOV C,M
MVI A,FFH

loop:
INX H
CMP M
JC skip
MOV A,M
skip:DCR C
JNZ loop

INX H
MOV M,A

HLT
```

#ORG 2201H
#DB 08,45,67,15,07,FE,78,21,63



3. A string of six data bytes is stored starting from memory location 2050H. The string includes some blanks (Bytes with zero value). Write a program to eliminate the blanks from the string. The updated string should be stored from memory location 2070H.

LXI H, 2250H

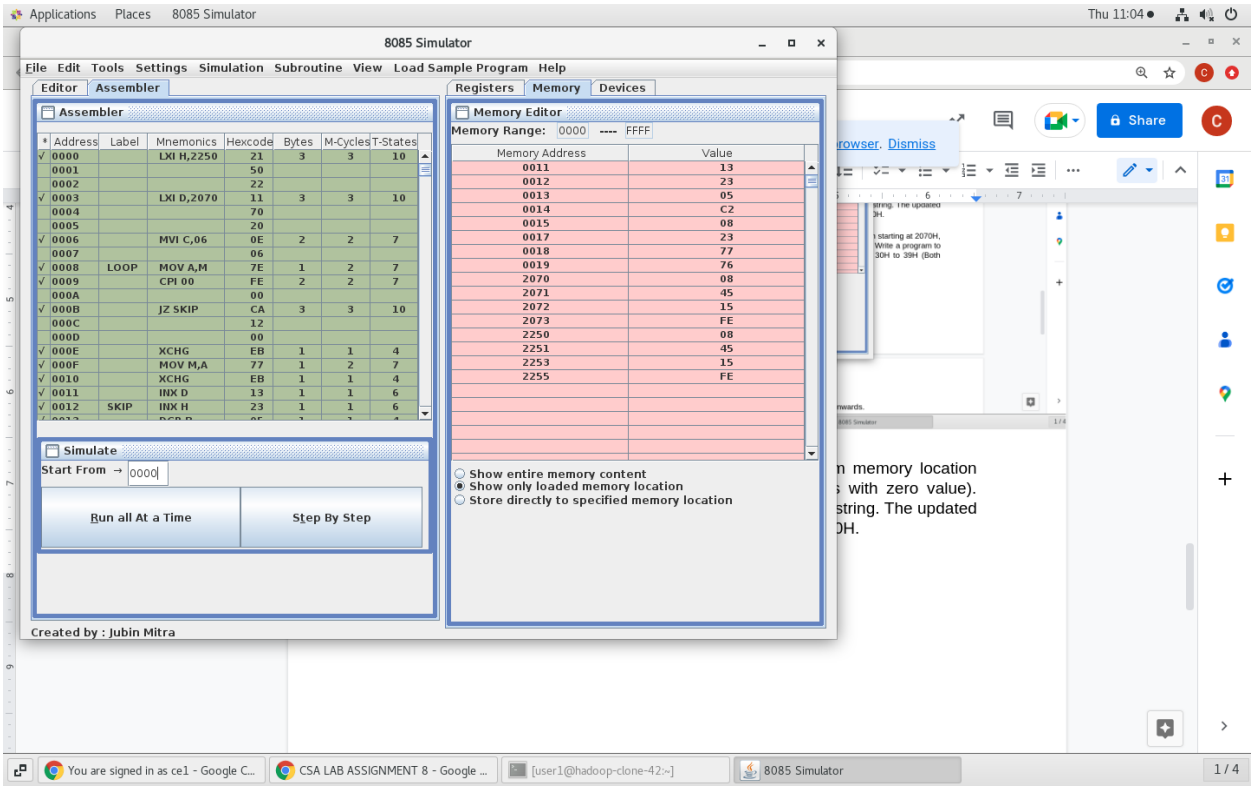
LXI D,2070H

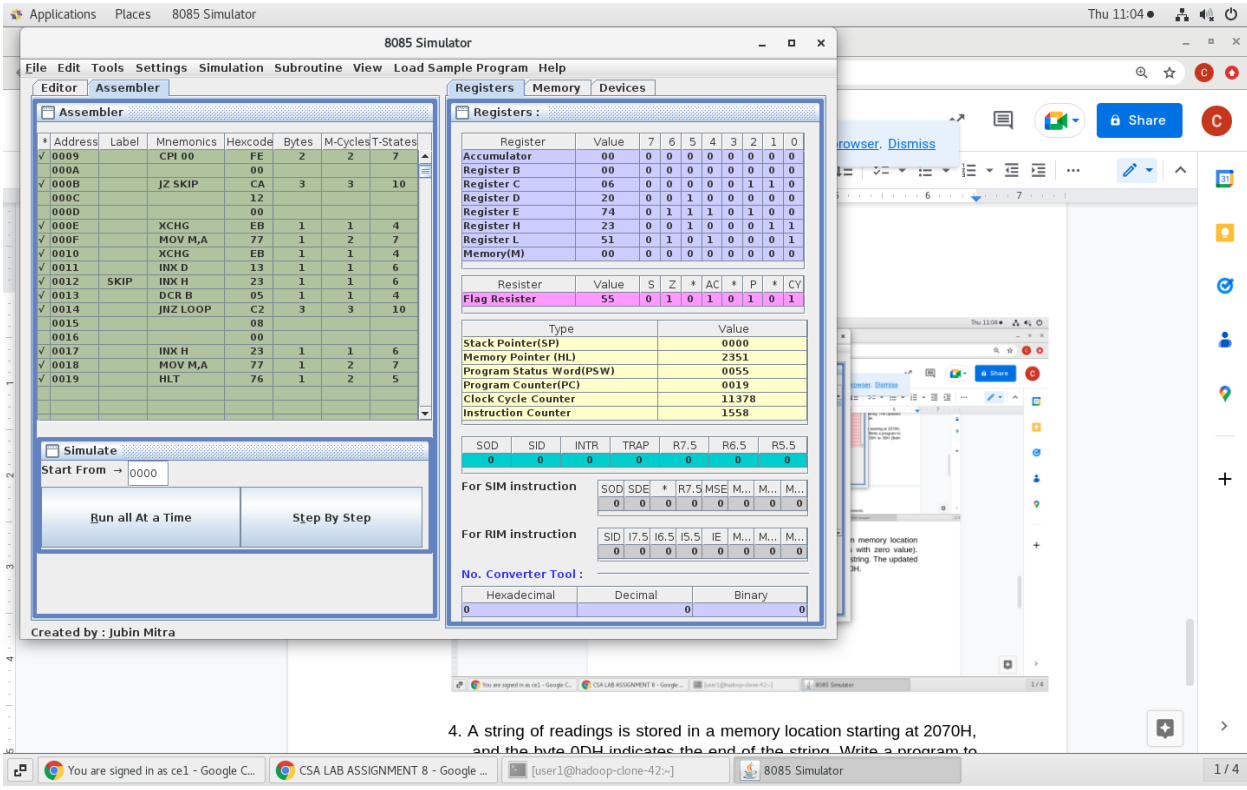
MVI C,06H

loop:

MOV A,M

```
CPI 00H

JZ skip

XCHG

MOV M,A

XCHG

INX D

skip:

INX H

DCR B

JNZ loop


INX H

MOV M,A


HLT


#ORG 2250H

#DB 08,45,00,15,00,FE
```





4. A string of readings is stored in a memory location starting at 2070H, and the byte 0DH indicates the end of the string. Write a program to

count the number of bytes that range between 30H to 39H (Both inclusive) and store them from 3000H onwards.

```
LXI D,3000H

LXI H,2070H

MVI C,0DH

MVI B,00H


loop:

MOV A,M

CPI 0DH

JZ exit

CPI 30H

JC skip

CPI 40H

JNC skip

INR B

skip:

INX H

JMP loop
```

exit:

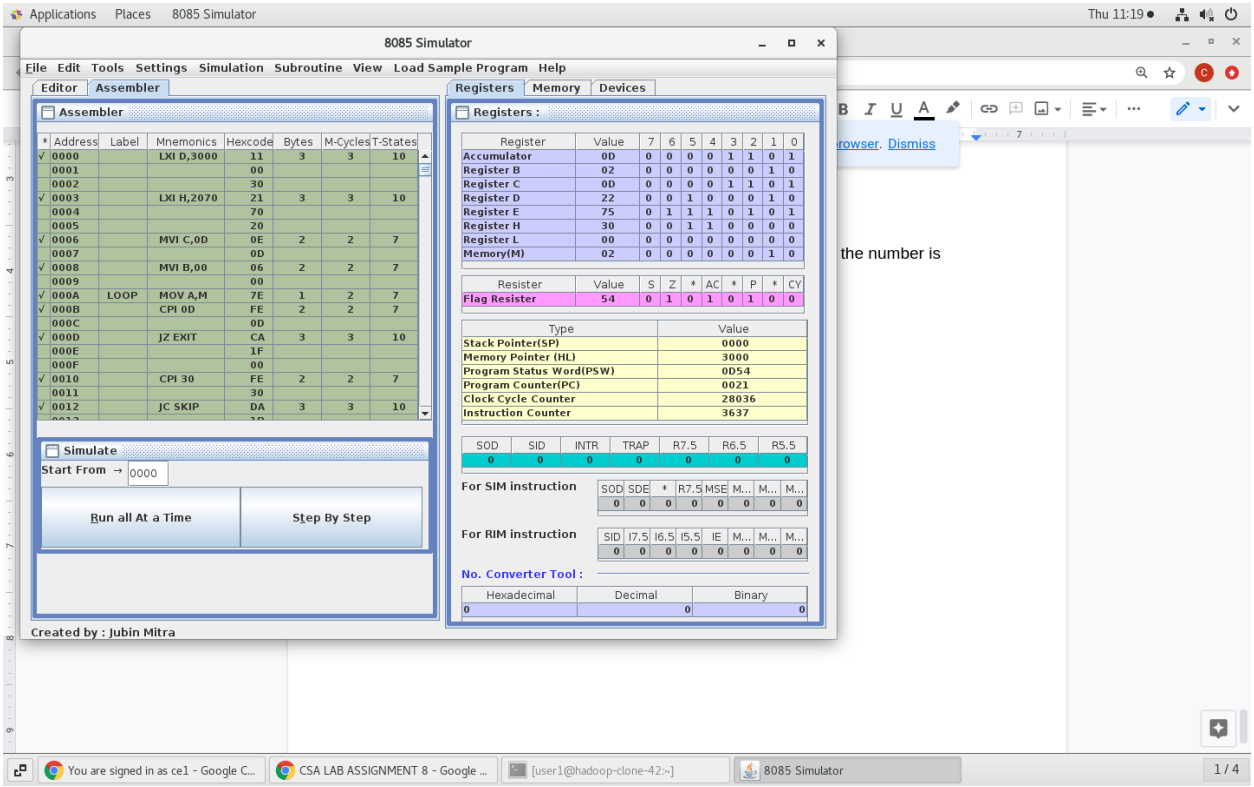
XCHG

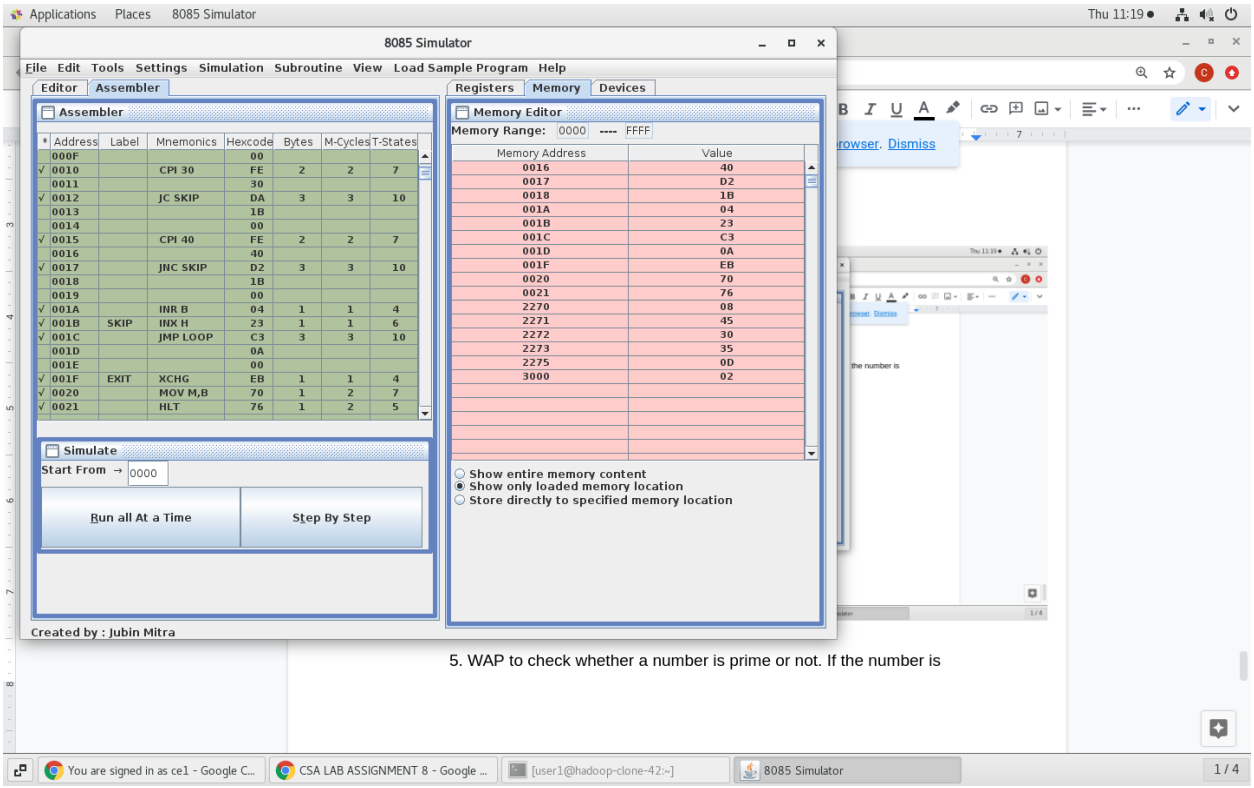
MOV M,B

HLT

#ORG 2270H

#DB 08,45,30,35,00,0DH





5. WAP to check whether a number is prime or not. If the number is prime it should have only two factors.

Input: 4401H = 07H

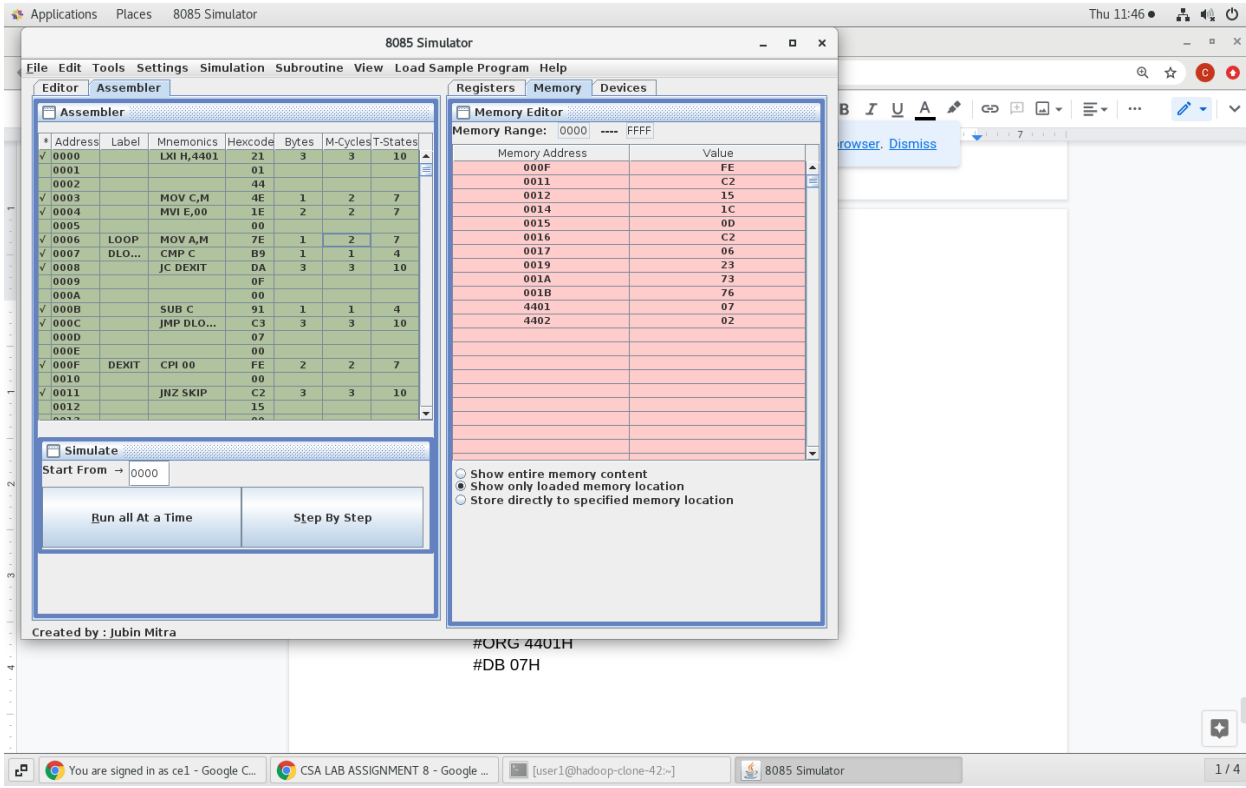
Output: 4402H = 02H

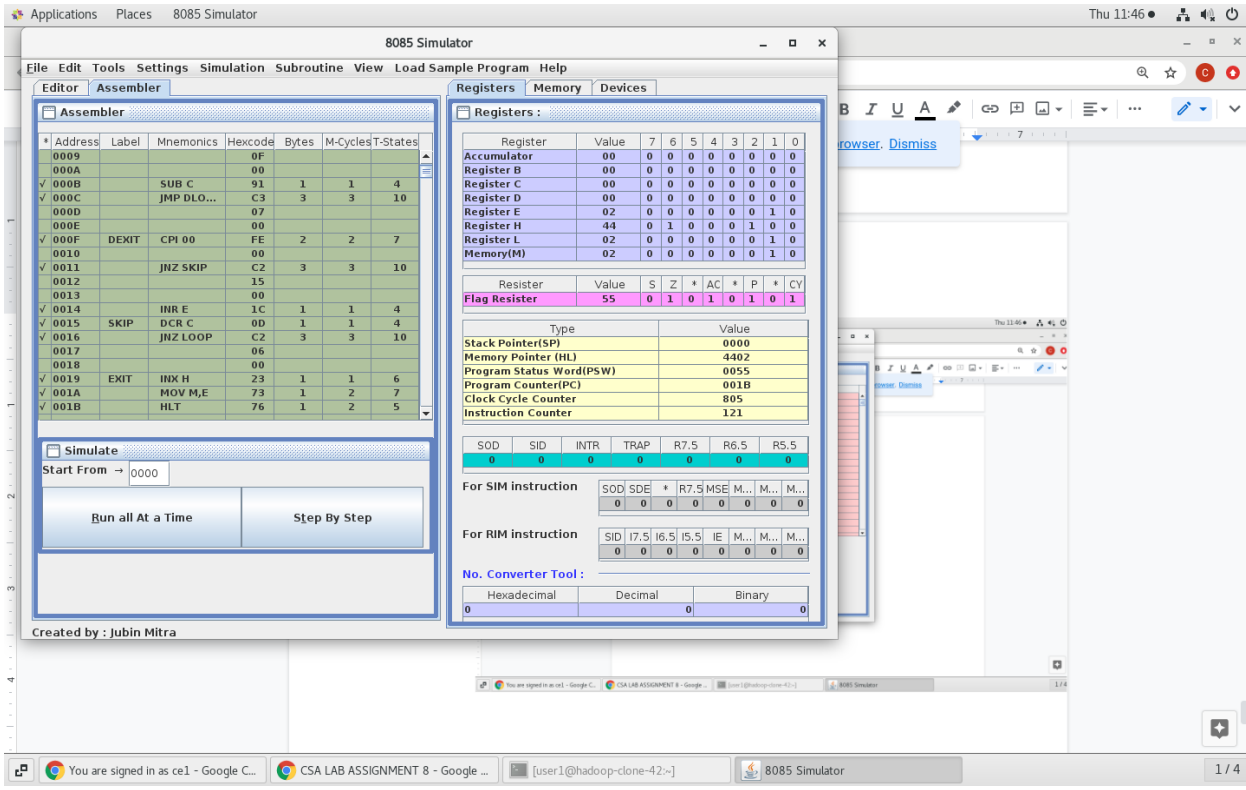
```
LXI H,4401H
MOV C,M
MVI E,00H

loop:
MOV A,M
dloop1:
CMP C
JC dexit
SUB C
JMP dloop1
dexit:
CPI 00H
JNZ skip
INR E
```

skip:
DCR C
JNZ loop

exit:
INX H
MOV M,E
HLT

#ORG 4401H
#DB 07H




Applications
Places
8085 Simulator
Thu 11:46
8085 Simulator
File Edit Tools Settings Simulation Subroutine View Load Sample Program Help
Editor
Assembler
Assembler
* Address Label Mnemonics Hexcode Bytes M-Cycles T-States
0009 0F
000A 00
√ 000B SUB C 91 1 1 4
√ 000C JMP DLO... C3 3 3 10
000D 07
000E 00
√ 000F DEXIT CPI 00 FE 2 2 7
0010 00
√ 0011 JNZ SKIP C2 3 3 10
0012 15
0013 00
√ 0014 INR E 1C 1 1 4
√ 0015 SKIP DCR C 0D 1 1 4
√ 0016 JNZ LOOP C2 3 3 10
0017 06
0018 00
√ 0019 EXIT INX H 23 1 1 6
√ 001A MOV M,E 73 1 2 7
√ 001B HLT 76 1 2 5
Simulate
Start From → 0000
Run all At a Time
Step By Step
Created by : Jubin Mitra
Registers
Memory
Devices
Registers :
Register Value 7 6 5 4 3 2 1 0
Accumulator 00 0 0 0 0 0 0 0 0
Register B 00 0 0 0 0 0 0 0 0
Register C 00 0 0 0 0 0 0 0 0
Register D 00 0 0 0 0 0 0 0 0
Register E 02 0 0 0 0 0 0 1 0
Register H 44 0 1 0 0 0 1 0 0
Register L 02 0 0 0 0 0 0 1 0
Memory(M) 02 0 0 0 0 0 0 1 0
Resister Value S Z * AC * P * CY
Flag Resister 55 0 1 0 1 0 1 0 1
Type Value
Stack Pointer(SP) 0000
Memory Pointer (HL) 4402
Program Status Word(PSW) 0055
Program Counter(PC) 001B
Clock Cycle Counter 805
Instruction Counter 121
SOD SID INTR TRAP R7.5 R6.5 R5.5
0 0 0 0 0 0 0
For SIM instruction
SOD SDE * R7.5 MSE M... M... M...
0 0 0 0 0 0 0 0
For RIM instruction
SID I7.5 I6.5 I5.5 IE M... M... M...
0 0 0 0 0 0 0 0
No. Converter Tool :
Hexadecimal Decimal Binary
0 0 0
You are signed in as ce1 - Google C...
CSA LAB ASSIGNMENT 8 - Google ...
[user1@hadoop-clone-42:~]
8085 Simulator
1 / 4