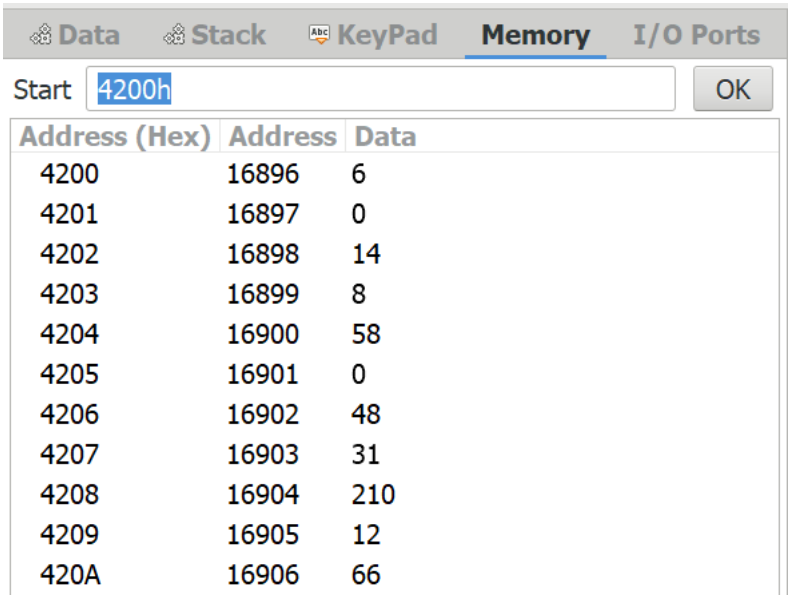
19CSE303: Embedded Systems

LABSHEET 3: ASSEMBLY LANGUAGE PROGRAMMING OF 8085 MICROPROCESSORS

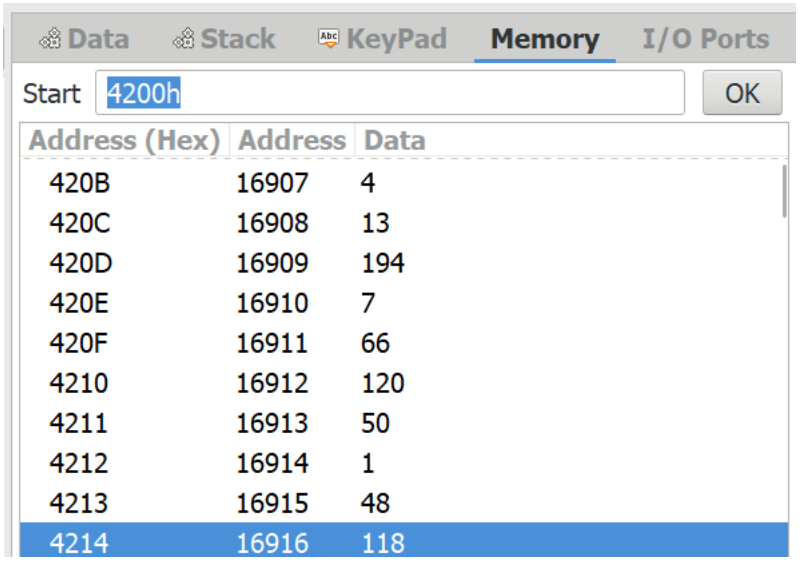
Name: Vinayak V Thayil **Roll Number:AM.EN.U4CSE21161**

---

1. Write the assembly program for the given object code where the starting address of the program is 4200H and the functionality of the ALP program.



Address (Hex)	Address	Data
4200	16896	6
4201	16897	0
4202	16898	14
4203	16899	8
4204	16900	58
4205	16901	0
4206	16902	48
4207	16903	31
4208	16904	210
4209	16905	12
420A	16906	66



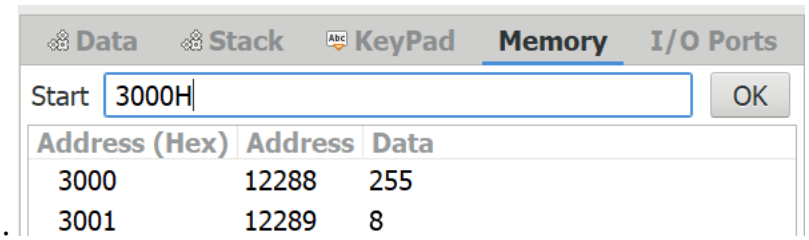
Address (Hex)	Address	Data
420B	16907	4
420C	16908	13
420D	16909	194
420E	16910	7
420F	16911	66
4210	16912	120
4211	16913	50
4212	16914	1
4213	16915	48
4214	16916	118

a)

Assembly	Instruction Size	Memory Address	Object Code in Hex	Flags or Register change details
MVI B,00	2bytes	4200	06 00	PC 4202

MVI C,08	2bytes	4202	0E	C 08,PC 4204
LDA 3000H	3bytes	4204	3A	A FF,PC 4204
RAR	1bytes	4207	1F	A 7F,C 1,PC 4208
JNC 420CH	3bytes	4208	D2	PC 420B
INR B	1bytes	420B	04	B 01,PC 420C
DCR C	1bytes	420C	0D	C 07,PC 420D
JNZ 4207H	3bytes	420D	C2	PC 4207
MOV A,B	1bytes	4210	78	Z1 P1,C 1,A 08 ,PC4211
STA 3001H	3bytes	4211	32	PC 4214
HLT	1bytes	4214	76	PC 4215

Note: Input data and the result of the program is in the following locations



Show the usages of address bus, data bus, control bus and also the status of PC and relevant registers in the execution of each of the instruction.

MVI B,00H:

Address Bus: 4200 4201
Data Bus: 06H 00
Control Bus: Opcode Fetch Memory Read MemR= 0 Me
PC : 4200+1 4201+1
Registers: B =00H, C = 08H, A= X (content in the mem location)

MVI C, 08H:

Address Bus : 4202 4203
Data Bus : 0E 08
Control Bus: Write to Register C
PC: 4202+1 4203+1
Register B = 00H, C= 08H, A= X(content in the mem location)

LDA 3000H:

Address Bus: 4204		4205	4206
Data Bus: 3A		00	30
Control Bus: Memory Read			
PC: 4204+1		4205+1	4206+1

Registers: B= X, C= 08H, A = X (content in the mem location)

RAR:

Address Bus: 4207
Data Bus: 1F
Control Bus : Rotate Accumulator Right
PC : 4207+1 = 4208
Registers : B = X, C= 08H, A =(CY)X (CY is the least significant bit of original A)

JNC 420CH:

Address Bus:	4208	4209	420A	
Data Bus :	D2	OC	42	(Not involved)
Control Bus:	Jump if no carry			
PC :	4208H	4209H+1	4210+1	

Register: B = X, C = 08H, A = (CY)X

INR B:

Address Bus:	420B	420B
Data Bus:	(no mess access)	04
Control Bus:	increment by 1	
PC:	420C	420B+1

Registers: B = X+1, C = 07H, A = (CY)X

JNZ 4207H:

Address Bus:	420D	420E	420F
Data Bus:	C2	07	42 (Not involved)
Control Bus:	4200H	420E+1	420F+1

PC: 4210 if jump not taken, updated to 4207h if jump taken
Registers: B = X+1, C = 07H, A = (CY)X

MOV A, B:

Adress Bus: 4210
Data Bus: 78
Control Bus: Move B to A
PC: 4211 4210+1
Registers: B = X+1, C= 07H, A = X+1

STA 3001H:

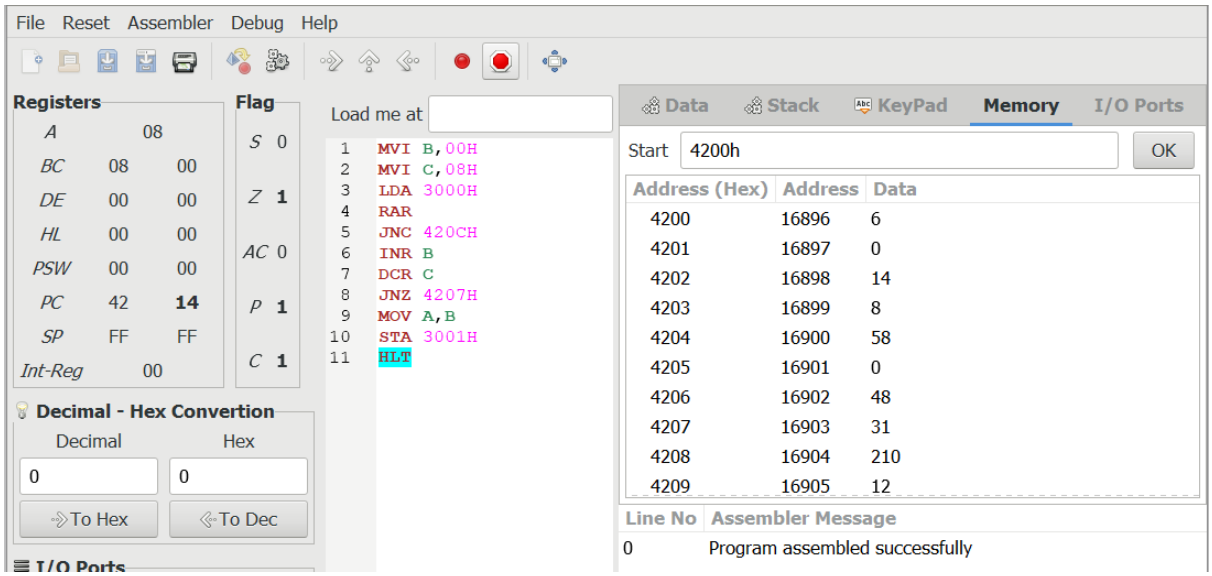
Address Bus: 3001H	4211	4212	4213
Data Bus: have value	32	01	30
Control Bus: Memory Write			
PC :	4211+1	4212+1	4213+1

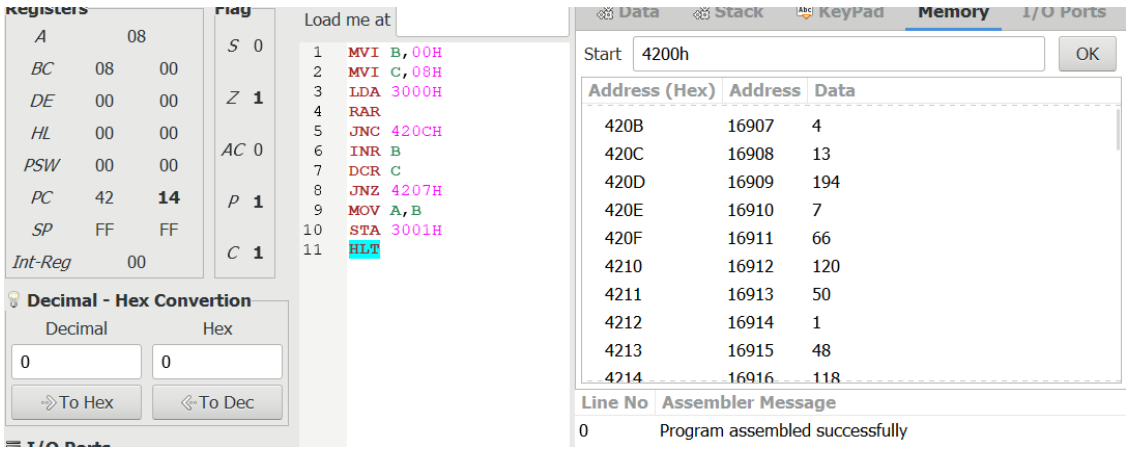
Registers: B = X+1, C= 07H, A = X+1

HLT:

Address Bus:	4214	
Data Bus:	76	(unchanged)
Control Bus:	stop	
PC:	4214	

Registers: B = X+1, C = 07H , A = X+1





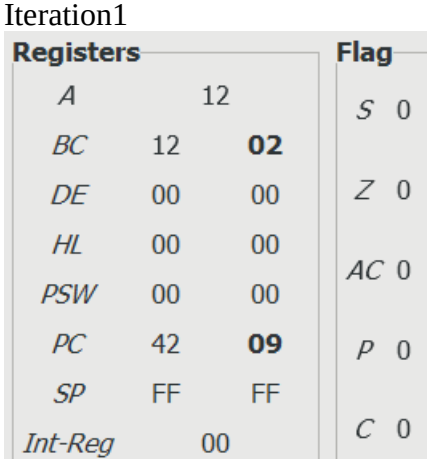
2. Fill the table for each of the assembly programs by selecting suitable instructions from 8085 Instruction Set. Show the output in 8085 simulators for the following programs and trace the program for 2 or 3 iterations. Explain each instruction and finally say what the program does.

Assembly	Instruction Size	Memory Address	Object Code in Hex	Flags or Register change details
MVI A,12H	2 bytes	4200	3E	A 12,PC 4202
MOV B,A	1 byte	4202	47	B 12,PC 4203
MVI C,03H	2 bytes	4203	0E	C 03,PC 4205
MVI A,00H	2 bytes	4205	3E	A 00,PC4207
Label1:ADD B	1 byte	4207	80	A 12-A 24-A 36 P 1,PC 4208
DCR C	1 byte	4208	0D	C 02-C 01 -C 00 PC 4209
JNZ Label1	3 bytes	4209	C2	PC 4207
STA 2001H	3 bytes	420C	32	PC 420F
HLT	1 byte	420F	76	PC 4210

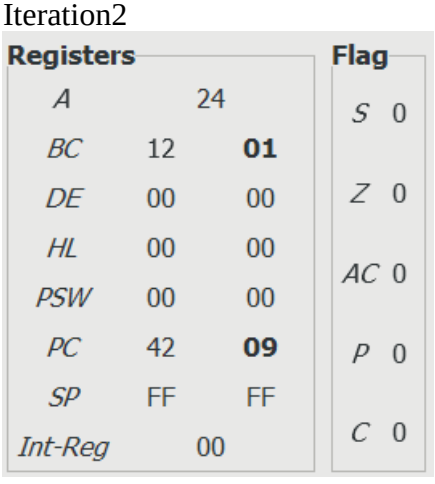
a) Mention the significance of Branch instruction JNZ. You can take the screenshot of register status in each iteration and copy it in the form of table.

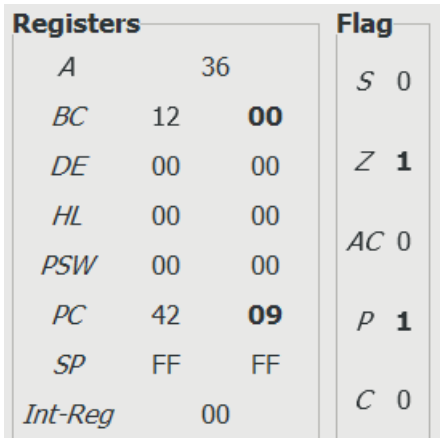
```
        MVI A, 12H
        MOV B, A
        MVI C, 03H
        MVI A, 00H
Label1: ADD B
        DCR C
        JNZ Label1
        STA 2001H
        HLT
```

Iteration1



Iteration2



Iteration3	Final Result
	

3. Write an assembly language program using loops to add the numbers starting from 1 to 50. B=1+2+3+……..+49+50