NAME : **Sujan Biswas**

BATCH: BSCE $2^{ND}$ YEAR , (LATERAL)

ROLL NO : 302010501003

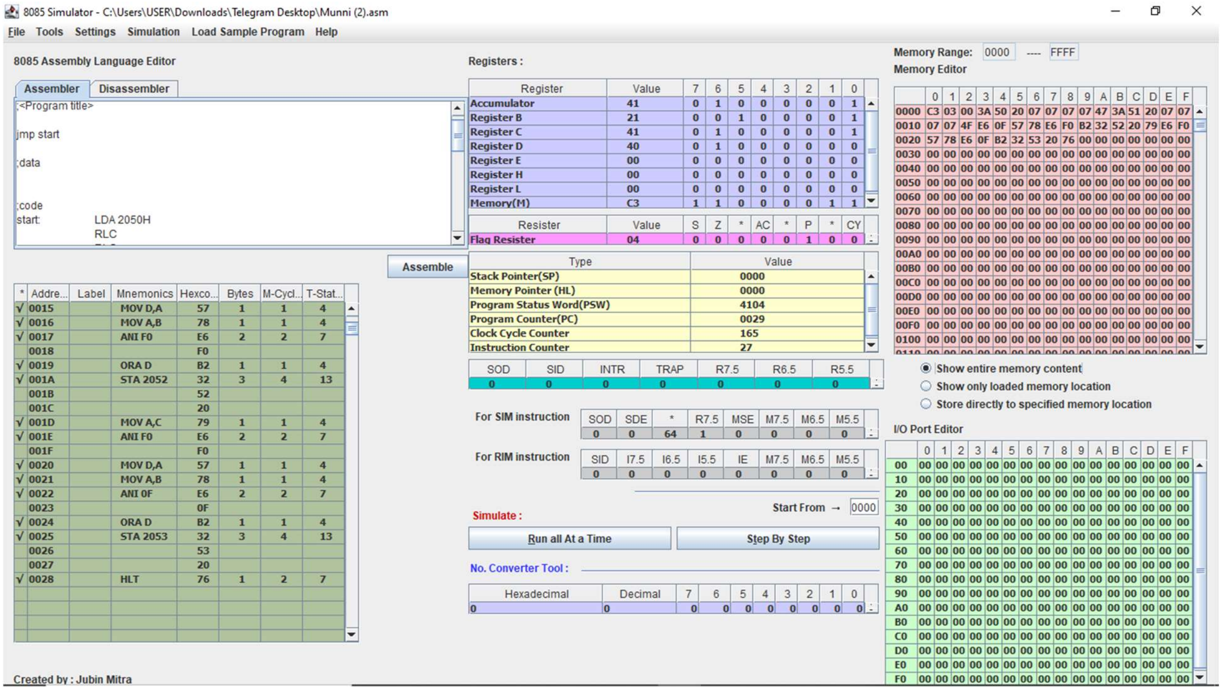
Microprocessor lab Assignment problem sheet #2

1. Two numbers $MN_H$ and $KL_H$ are stored in 2050 H and 2051 H , respectively . Write a program to assemble them a $NK_H$ and $LM_H$ store them in 2052 H and 2053 H.

| SL NO. | ADDRESS | OPCODE IN HEX CODE | LABEL | INSTRUCTIONS | COMMENTS |
|---|---|---|---|---|---|
| 1 | 0000 | C3 | | JMP START | Start the program |
| | 0001 | 03 | | | |
| | 0002 | 00 | | | |
| 2 | 0003 | 3A | START | LDA2050 H | Load accumulator direct memory location 2050 H [A=MN H] |
| | 0004 | 50 | | | |
| | 0005 | 20 | | | |
| 3 | 0006 | 07 | | RLC | Rotate accumulator left without carry |
| 4 | 0007 | 07 | | RLC | Rotate accumulator left without carry |
| 5 | 0008 | 07 | | RLC | Rotate accumulator left without carry |
| 6 | 0009 | 07 | | RLC | Rotate accumulator left without carry [After 4 rotation ,A=NM H] |
| 7 | 000A | 47 | | MOV B,A | Move accumulator to B register [B=A=NM H] |
| 8 | 000B | 3A | | LDA 2051 H | Load accumulator direct memory location 2051 H [A=KL H] |
| | 000C | 51 | | | |
| | 000D | 20 | | | |
| 9 | 000E | 07 | | RLC | Rotate accumulator left without carry |
| 10 | 000F | 07 | | RLC | Rotate accumulator left without carry |
| 11 | 0010 | 07 | | RLC | Rotate accumulator left without carry |
| 12 | 0011 | 07 | | RLC | Rotate accumulator left without carry [After 4 rotation, A= LK H] |
| 13 | 0012 | 4F | | MOV C,A | Move accumulator to C register [C=A= LK H] |
| 14 | 0013 | E6 | | ANI 0F | A=0K H |
| | 0014 | 0F | | | |
| 15 | 0015 | 57 | | MOV D,A | Move accumulator to D register [D=A=0K H] |
| 16 | 0016 | 78 | | MOV A,B | Move B register to accumulator [A=B=NM H] |
| 17 | 0017 | E6 | | ANI F0 | A=N0 H |

| | | | | | |
|---|---|---|---|---|---|
| | 0018 | F0 | | | |
| 18 | 0019 | B2 | | ORA D | A=N0 H\|0K H=NK H |
| 19 | 001A | 32 | | STA 2052 H | Load the contents of the accumulator in the address location 2052 H, M[2052= NK H] |
| | 001B | 52 | | | |
| | 001C | 20 | | | |
| 20 | 001D | 79 | | MOV A,C | Move C register to accumulator [A=C=LK H] |
| 21 | 001E | E6 | | ANI F0 | A=L0 H |
| | 001F | F0 | | | |
| 22 | 0020 | 57 | | MOV D,A | Move accumulator to D register[A=D=L0 H] |
| 23 | 0021 | 78 | | MOV A,B | Move B register accumulator to[A=B=NM H] |
| 24 | 0022 | E6 | | ANI 0F | A=0M H |
| | 0023 | 0F | | | |
| 25 | 0024 | B2 | | ORA D | A=0M H\|L0 H=LM H |
| 26 | 0025 | 32 | | STA 2053 H | Load the contents of the accumulator in the address location 2053 H, M[2053=LM H] |
| | 0026 | 53 | | | |
| | 0027 | 20 | | | |
| 27 | 0028 | 76 | | HLT | Terminate the program. |

Simulator with loaded machine code:



8085 Simulator - C:\Users\USER\Downloads\Telegram Desktop\Munni (2).asm
File Tools Settings Simulation Load Sample Program Help
8085 Assembly Language Editor
Assembler
Disassembler
Assemble
Registers :
Memory Range: 0000 ---- FFFF
Memory Editor
I/O Port Editor
Simulate :
Run all At a Time
Step By Step
No. Converter Tool :
Created by : Jubin Mitra

```
;<Program title>

jmp start

;data


;code
start:          LDA 2050H
                RLC
```

| * | Addre... | Label | Mnemonics | Hexco... | Bytes | M-Cycl... | T-Stat... |
|---|---|---|---|---|---|---|---|
| √ | 0015 | | MOV D,A | 57 | 1 | 1 | 4 |
| √ | 0016 | | MOV A,B | 78 | 1 | 1 | 4 |
| √ | 0017 | | ANI F0 | E6 | 2 | 2 | 7 |
| | 0018 | | | F0 | | | |
| √ | 0019 | | ORA D | B2 | 1 | 1 | 4 |
| √ | 001A | | STA 2052 | 32 | 3 | 4 | 13 |
| | 001B | | | 52 | | | |
| | 001C | | | 20 | | | |
| √ | 001D | | MOV A,C | 79 | 1 | 1 | 4 |
| √ | 001E | | ANI F0 | E6 | 2 | 2 | 7 |
| | 001F | | | F0 | | | |
| √ | 0020 | | MOV D,A | 57 | 1 | 1 | 4 |
| √ | 0021 | | MOV A,B | 78 | 1 | 1 | 4 |
| √ | 0022 | | ANI 0F | E6 | 2 | 2 | 7 |
| | 0023 | | | 0F | | | |
| √ | 0024 | | ORA D | B2 | 1 | 1 | 4 |
| √ | 0025 | | STA 2053 | 32 | 3 | 4 | 13 |
| | 0026 | | | 53 | | | |
| | 0027 | | | 20 | | | |
| √ | 0028 | | HLT | 76 | 1 | 2 | 7 |

| Register | Value | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Accumulator | 41 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| Register B | 21 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| Register C | 41 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| Register D | 40 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register E | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register H | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register L | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Memory(M) | C3 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |

| Resister | Value | S | Z | * | AC | * | P | * | CY |
|---|---|---|---|---|---|---|---|---|---|
| Flag Resister | 04 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

| Type | Value |
|---|---|
| Stack Pointer(SP) | 0000 |
| Memory Pointer (HL) | 0000 |
| Program Status Word(PSW) | 4104 |
| Program Counter(PC) | 0029 |
| Clock Cycle Counter | 165 |
| Instruction Counter | 27 |

| 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 | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 64 | 1 | 0 | 0 | 0 | 0 |

For RIM instruction

| SID | I7.5 | I6.5 | I5.5 | IE | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Start From → 0000

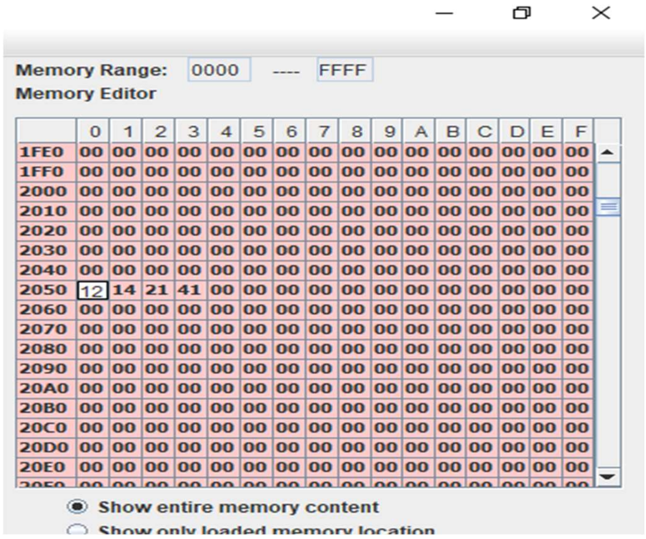
| Hexadecimal | Decimal | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000 | C3 | 03 | 00 | 3A | 50 | 20 | 07 | 07 | 07 | 07 | 47 | 3A | 51 | 20 | 07 | 07 |
| 0010 | 07 | 07 | 4F | E6 | 0F | 57 | 78 | E6 | F0 | B2 | 32 | 52 | 20 | 79 | E6 | F0 |
| 0020 | 57 | 78 | E6 | 0F | B2 | 32 | 53 | 20 | 76 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0030 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0040 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0090 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00C0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00E0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

● Show entire memory content
○ Show only loaded memory location
○ Store directly to specified memory location

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 30 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 40 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 50 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 60 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 70 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 80 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 90 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| C0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| E0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

Sample input and output (M[2050 H]=12 H, M[2051 H]=14 H)



Memory Range: 0000 ---- FFFF

Memory Editor

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1FE0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1FF0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2000 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2010 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2020 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2030 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2040 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2050 | 12 | 14 | 21 | 41 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2090 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20C0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20E0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

● Show entire memory content

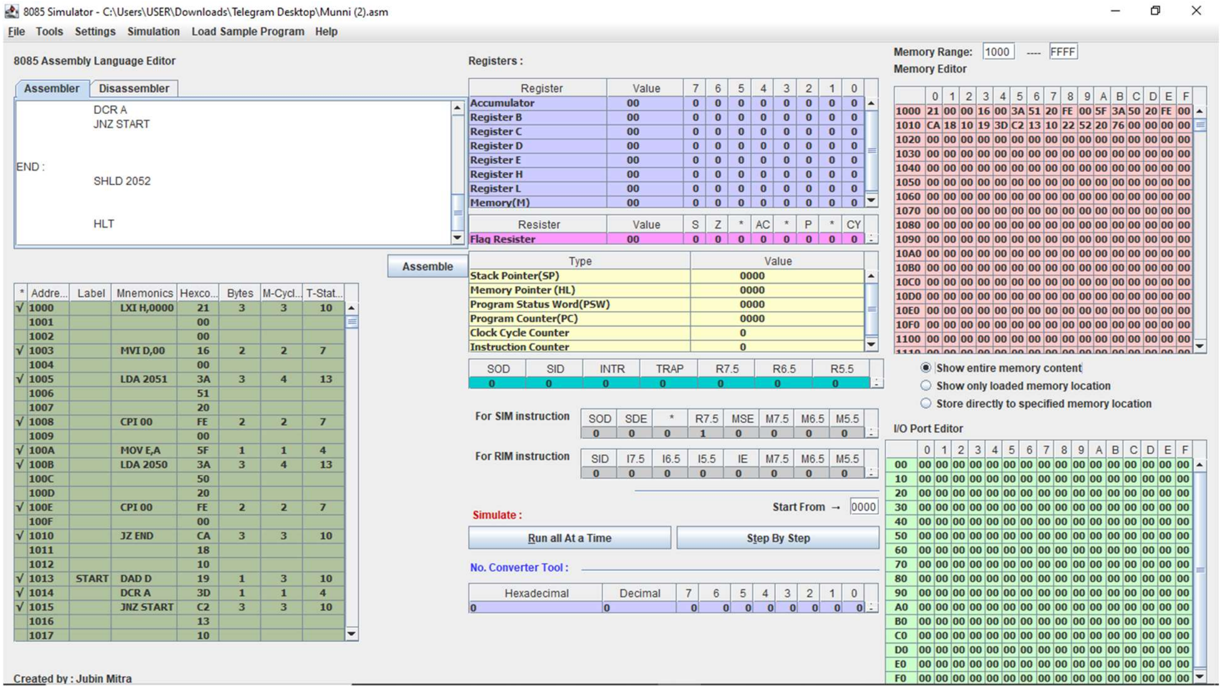
○ Show only loaded memory location

2. Two numbers A & B are stored in 2050 H and 2051 H , respectively . Write a program to perform A*B and store the results in 2052 H and 2053 H.

| SL NO. | ADDRESS | OPCODE IN HEX CODE | LABEL | INSTRUCTIONS | COMMENTS |
|---|---|---|---|---|---|
| 1 | 1000 | 21 | | LXI H,0000 | Contents of memory location 0000 H into HL register pair |
| | 1001 | 00 | | | |
| | 1002 | 00 | | | |
| 2 | 1003 | 16 | | MVI D,00 | Move immediate to D register |
| | 1004 | 00 | | | |
| 3 | 1005 | 3A | | LDA 2051 H | Load accumulator direct memory location 2051 H<br>Get B |
| | 1006 | 51 | | | |
| | 1007 | 20 | | | |
| 4 | 1008 | FE | | CPI 00 H | If B=0? |
| | 1009 | 00 | | | |
| 5 | 100A | 5F | | MOV E,A | Move accumulator to E register<br>E=A |
| 6 | 100B | 3A | | LDA 2050 H | Load accumulator direct memory location 2050H<br>Get A |
| | 100C | 50 | | | |
| | 100D | 20 | | | |
| 7 | 100E | FE | | CPI 00 H | Is A=0? |
| | 100F | 00 | | | |
| 8 | 1010 | CA | | JZ END | If A=0,nothing to do |
| | 1011 | 18 | | | |
| | 1012 | 10 | | | |
| 9 | 1013 | 19 | START | DAD D | HL + DE |
| 10 | 1014 | 3D | | DCR A | Decrement the A |
| 11 | 1015 | C2 | | JNZ START | If DE has not been added A times ,add again |
| | 1016 | 13 | | | |
| | 1017 | 10 | | | |
| 12 | 1018 | 22 | END | SHLD 2052 H | |

| | | | | | |
|---|---|---|---|---|---|
| | 1019 | 52 | | | Store results as specified |
| | 101A | 20 | | | |
| 13 | 101B | 76 | | HLT | Stop the program |

Simulator with loaded machine code:



8085 Simulator - C:\Users\USER\Downloads\Telegram Desktop\Munni (2).asm

8085 Assembly Language Editor

```
            DCR A
            JNZ START

END :
            SHLD 2052


            HLT
```

| * | Addre... | Label | Mnemonics | Hexco... | Bytes | M-Cycl... | T-Stat... |
|---|---|---|---|---|---|---|---|
| √ | 1000 | | LXI H,0000 | 21 | 3 | 3 | 10 |
| | 1001 | | | 00 | | | |
| | 1002 | | | 00 | | | |
| √ | 1003 | | MVI D,00 | 16 | 2 | 2 | 7 |
| | 1004 | | | 00 | | | |
| √ | 1005 | | LDA 2051 | 3A | 3 | 4 | 13 |
| | 1006 | | | 51 | | | |
| | 1007 | | | 20 | | | |
| √ | 1008 | | CPI 00 | FE | 2 | 2 | 7 |
| | 1009 | | | 00 | | | |
| √ | 100A | | MOV E,A | 5F | 1 | 1 | 4 |
| √ | 100B | | LDA 2050 | 3A | 3 | 4 | 13 |
| | 100C | | | 50 | | | |
| | 100D | | | 20 | | | |
| √ | 100E | | CPI 00 | FE | 2 | 2 | 7 |
| | 100F | | | 00 | | | |
| √ | 1010 | | JZ END | CA | 3 | 3 | 10 |
| | 1011 | | | 18 | | | |
| | 1012 | | | 10 | | | |
| √ | 1013 | START | DAD D | 19 | 1 | 3 | 10 |
| √ | 1014 | | DCR A | 3D | 1 | 1 | 4 |
| √ | 1015 | | JNZ START | C2 | 3 | 3 | 10 |
| | 1016 | | | 13 | | | |
| | 1017 | | | 10 | | | |

Created by : Jubin Mitra

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 | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register H | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register L | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Memory(M) | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Resister | Value | S | Z | * | AC | * | P | * | CY |
|---|---|---|---|---|---|---|---|---|---|
| Flag Resister | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Type | Value |
|---|---|
| Stack Pointer(SP) | 0000 |
| Memory Pointer (HL) | 0000 |
| Program Status Word(PSW) | 0000 |
| Program Counter(PC) | 0000 |
| Clock Cycle Counter | 0 |
| Instruction Counter | 0 |

| 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 | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

For RIM instruction

| SID | I7.5 | I6.5 | I5.5 | IE | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Start From → 0000

Simulate :

No. Converter Tool :

| Hexadecimal | Decimal | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Memory Range: 1000 ---- FFFF

Memory Editor

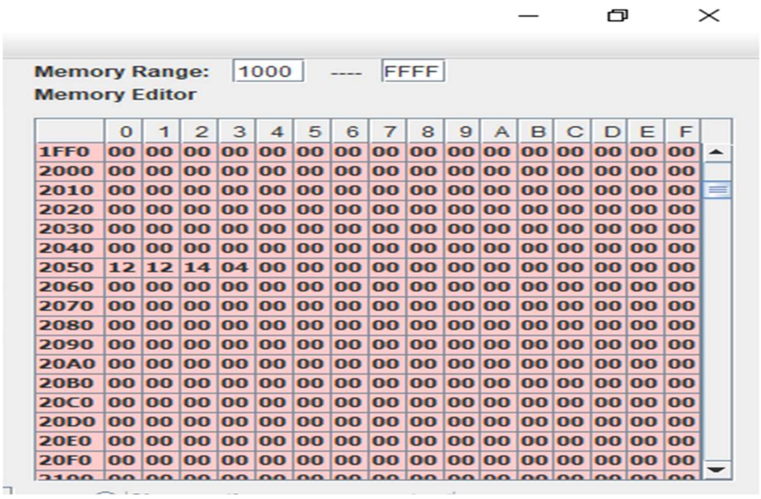
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1000 | 21 | 00 | 00 | 16 | 00 | 3A | 51 | 20 | FE | 00 | 5F | 3A | 50 | 20 | FE | 00 |
| 1010 | CA | 18 | 10 | 19 | 3D | C2 | 13 | 10 | 22 | 52 | 20 | 76 | 00 | 00 | 00 | 00 |
| 1020 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1030 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1040 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1090 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10C0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10E0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

● Show entire memory content
○ Show only loaded memory location
○ Store directly to specified memory location

I/O Port Editor

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 30 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 40 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 50 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 60 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 70 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 80 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 90 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| C0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| E0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

Sample input and output (A= M[2050 H]=12 H,B=M[2051 H]=12 H, A*B=144 H, M[2052 H]= 12 H, M[2053 H]=04 H).



Memory Range: 1000 ---- FFFF

Memory Editor

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1FF0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2000 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2010 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2020 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2030 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2040 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2050 | 12 | 12 | 14 | 04 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2090 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20C0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20E0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

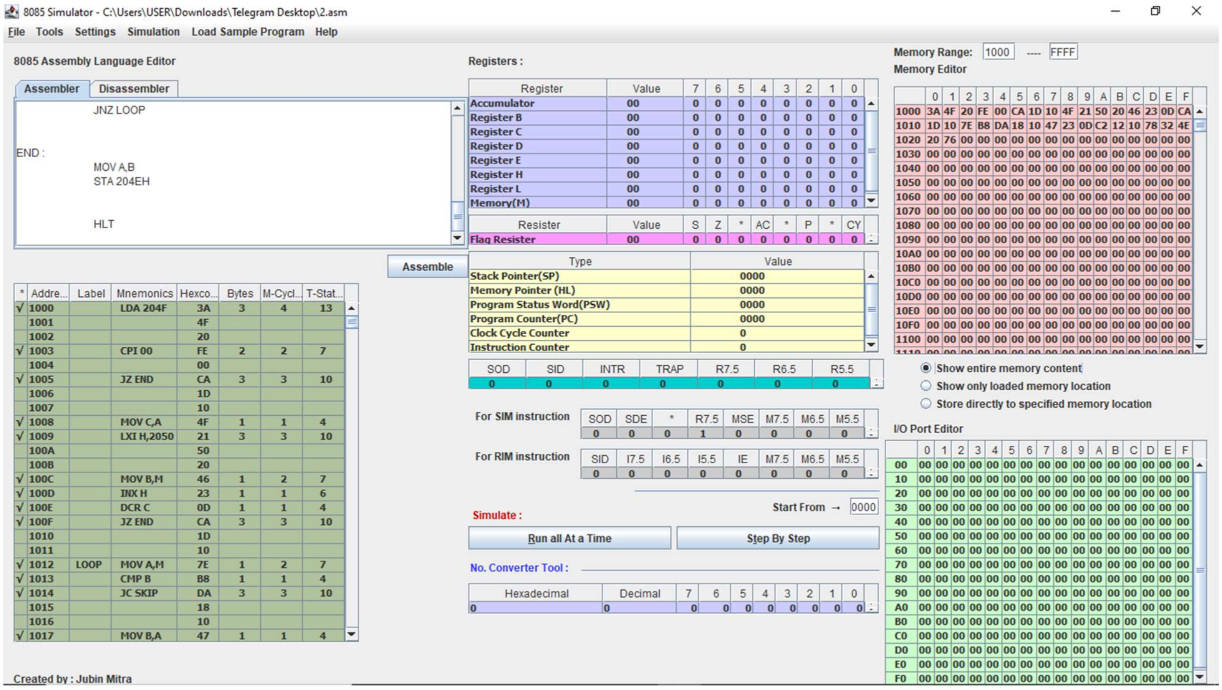
3.N numbers are stored in consecutive m/m location starting the from 2050 H. The value N is stored in 204F H.

I)Find maximum among the N numbers.

| SL NO. | ADDRESS | OPCODE IN HEX CODE | LABEL | INSTRUCTIONS | COMMENTS |
|---|---|---|---|---|---|
| 1 | 1000 | 3A |  | LDA 204F H | Load accumulator direct memory location 204F H [A=N H] |
|  | 1001 | 4F |  |  |  |
|  | 1002 | 20 |  |  |  |
| 2 | 1003 | FE |  | CPI  00 H | Is N =0? |
|  | 1004 | 00 |  |  |  |
| 3 | 1005 | CA |  | JZ END | If N =0,nothing to do |
|  | 1006 | 1D |  |  |  |
|  | 1007 | 10 |  |  |  |
| 4 | 1008 | 4F |  | MOV C,A | Move accumulator to  C register |
| 5 | 1009 | 21 |  | LXI 2050 H | Contents of memory location 2050 H into HL register pair |
|  | 100A | 50 |  |  |  |
|  | 100B | 20 |  |  |  |
| 6 | 100C | 46 |  | MOV B,M | Move memory address to B register B is current maximum number |
| 7 | 100D | 23 |  | INX H | Increase the HL register ,then go the next number |
| 8 | 100E | 0D |  | DCR C | Decrement the C register , then  check the number |
| 9 | 100F | CA |  | JZ END | If end , nothing to do ,store this |
|  | 1010 | 1D |  |  |  |
|  | 1011 | 7E |  |  |  |
| 10 | 1012 | 7E | LOOP | MOV A,M | Move the memory address to accumulator |
| 11 | 1013 | B8 |  | CMP B | Compare against current maximum |
| 12 | 1014 | DA |  | JC SKIP | If B>A, do nothing |
|  | 1015 | 18 |  |  |  |
|  | 1016 | 10 |  |  |  |
| 13 | 1017 | 47 |  | MOV B,A | Move accumulator to B register load a new maximum |
| 14 | 1018 | 23 | SKIP | INX H | Increase the HL ,then go the next number |
| 15 | 1019 | 0D |  | DCR C | Decrement the C then check the number |
| 16 | 101A | C2 |  | JNZ LOOP | If numbers left, continue checking |
|  | 101B | 12 |  |  |  |
|  | 101C | 10 |  |  |  |
| 17 | 101D | 78 | END | MOV A,B | Move B register to accumulator [A=Maximum] |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 18 | 101E | 32 |  | STA 204E H | Store the maximum |
|  | 101F | 4E |  |  |  |
|  | 1020 | 20 |  |  |  |
| 19 | 1021 | 76 |  | HLT | Stop the program |

Simulator with loaded machine code:


8085 Simulator - C:\Users\USER\Downloads\Telegram Desktop\2.asm
File Tools Settings Simulation Load Sample Program Help
8085 Assembly Language Editor
Assembler Disassembler
Assemble
Registers :
Simulate :
Run all At a Time
Step By Step
No. Converter Tool :
Created by : Jubin Mitra

```
                JNZ LOOP

END :
                MOV A,B
                STA 204EH


                HLT
```

| * | Addre... | Label | Mnemonics | Hexco... | Bytes | M-Cycl... | T-Stat... |
|---|---|---|---|---|---|---|---|
| √ | 1000 |  | LDA 204F | 3A | 3 | 4 | 13 |
|  | 1001 |  |  | 4F |  |  |  |
|  | 1002 |  |  | 20 |  |  |  |
| √ | 1003 |  | CPI 00 | FE | 2 | 2 | 7 |
|  | 1004 |  |  | 00 |  |  |  |
| √ | 1005 |  | JZ END | CA | 3 | 3 | 10 |
|  | 1006 |  |  | 1D |  |  |  |
|  | 1007 |  |  | 10 |  |  |  |
| √ | 1008 |  | MOV C,A | 4F | 1 | 1 | 4 |
| √ | 1009 |  | LXI H,2050 | 21 | 3 | 3 | 10 |
|  | 100A |  |  | 50 |  |  |  |
|  | 100B |  |  | 20 |  |  |  |
| √ | 100C |  | MOV B,M | 46 | 1 | 2 | 7 |
| √ | 100D |  | INX H | 23 | 1 | 1 | 6 |
| √ | 100E |  | DCR C | 0D | 1 | 1 | 4 |
| √ | 100F |  | JZ END | CA | 3 | 3 | 10 |
|  | 1010 |  |  | 1D |  |  |  |
|  | 1011 |  |  | 10 |  |  |  |
| √ | 1012 | LOOP | MOV A,M | 7E | 1 | 2 | 7 |
| √ | 1013 |  | CMP B | B8 | 1 | 1 | 4 |
| √ | 1014 |  | JC SKIP | DA | 3 | 3 | 10 |
|  | 1015 |  |  | 18 |  |  |  |
|  | 1016 |  |  | 10 |  |  |  |
| √ | 1017 |  | MOV B,A | 47 | 1 | 1 | 4 |

| 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 | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register H | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register L | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Memory(M) | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Resister | Value | S | Z | * | AC | * | P | * | CY |
|---|---|---|---|---|---|---|---|---|---|
| Flag Resister | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Type | Value |
|---|---|
| Stack Pointer(SP) | 0000 |
| Memory Pointer (HL) | 0000 |
| Program Status Word(PSW) | 0000 |
| Program Counter(PC) | 0000 |
| Clock Cycle Counter | 0 |
| Instruction Counter | 0 |

| 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 | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

For RIM instruction

| SID | I7.5 | I6.5 | I5.5 | IE | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Start From → 0000

| Hexadecimal | Decimal | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Memory Range: 1000 ---- FFFF

Memory Editor

```
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
1000 3A 4F 20 FE 00 CA 1D 10 4F 21 50 20 46 23 0D CA
1010 1D 10 7E B8 DA 18 10 47 23 0D C2 12 10 78 32 4E
1020 20 76 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

● Show entire memory content
○ Show only loaded memory location
○ Store directly to specified memory location

I/O Port Editor

```
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

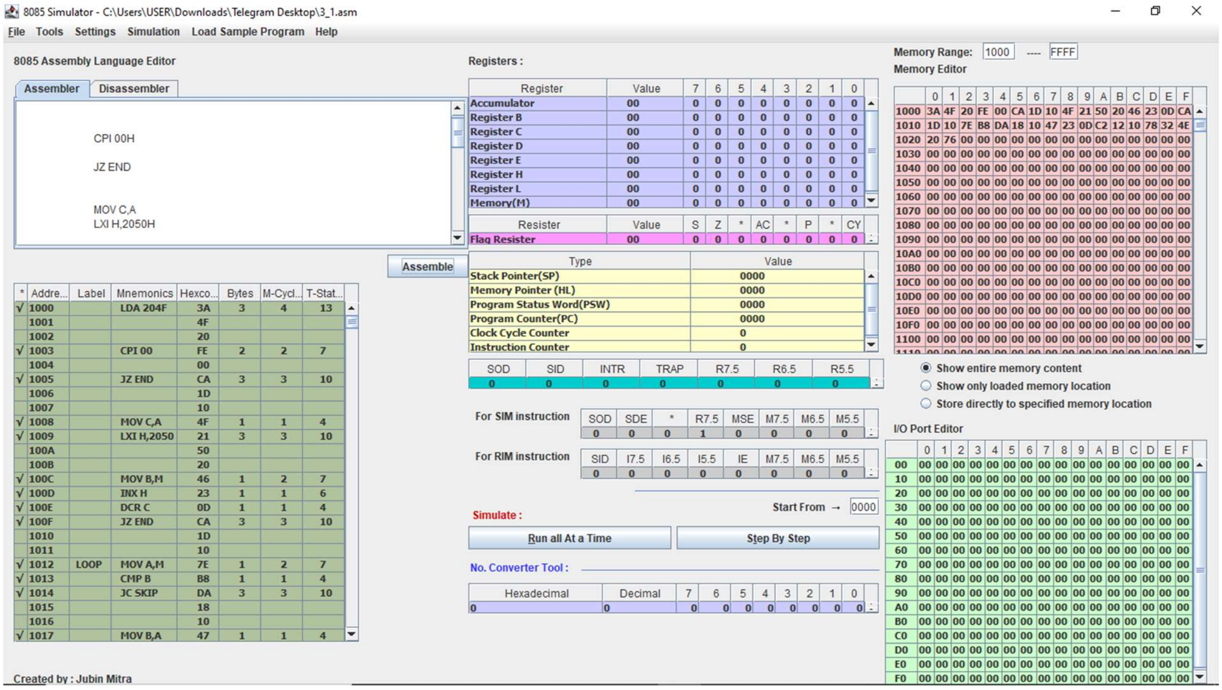
ii) Find the minimum among the N numbers.

We store the result (Minimum) in 204E H memory location.

| SL NO. | ADDRESS | OPCODE IN HEX CODE | LABEL | INSTRUCTIONS | COMMENTS |
|---|---|---|---|---|---|
| 1 | 1000 | 3A |  | LDA 204F H | Load accumulator direct memory location 204F H<br>[A=N H] |
|  | 1001 | 4F |  |  |  |
|  | 1002 | 20 |  |  |  |
| 2 | 1003 | FE |  | CPI 00 H | Is N=0? |
|  | 1004 | 00 |  |  |  |
| 3 | 1005 | CA |  | JZ END | If N =0,nothing to do |
|  | 1006 | 1D |  |  |  |
|  | 1007 | 10 |  |  |  |

| | | | | | |
|---|---|---|---|---|---|
| 4 | 1008 | 4F | | MOV C,A | Move accumulator to C register |
| 5 | 1009 | 21 | | LXI H ,2050 | Contents of memory location 2050 H into HL register pair |
| | 100A | 50 | | | |
| | 100B | 20 | | | |
| 6 | 100C | 46 | | MOV B,M | Move memory address to B register , B is current minimum number |
| 7 | 100D | 23 | | INX H | Increase the HL register ,then go the next number |
| 8 | 100E | 0D | | DCR C | Decrement the C register ,then check the number |
| 9 | 100F | CA | | JZ END | If end ,nothing to do ,store this |
| | 1010 | 1D | | | |
| | 1011 | 10 | | | |
| 10 | 1012 | 7E | LOOP | MOV A,M | Move the memory address to accumulator |
| 11 | 1013 | B8 | | CMP B | Compare against current minimum |
| 12 | 1014 | DA | | JC SKIP | If B>A, do nothing |
| | 1015 | 18 | | | |
| | 1016 | 10 | | | |
| 13 | 1017 | 47 | | MOV B,A | Move accumulator to B register, load a new minimum |
| 14 | 1018 | 23 | SKIP | INX H | Increase the HL ,then go to the next number |
| 15 | 1019 | 0D | | DCR C | Decrement the C register ,then check the number |
| 16 | 101A | C2 | | JNZ LOOP | If numbers left, continue checking |
| | 101B | 12 | | | |
| | 101C | 10 | | | |
| 17 | 101D | 78 | END | MOV A, B | Move B register to accumulator [A=Minimum] |
| 18 | 101E | 32 | | STA 204E H | Store the minimum |
| | 101F | 4E | | | |
| | 1020 | 20 | | | |
| 19 | 1021 | 76 | | HLT | Stop the program |

Simulator with loaded machine code:



8085 Simulator - C:\Users\USER\Downloads\Telegram Desktop\3_1.asm

```
CPI 00H

JZ END


MOV C,A
LXI H,2050H
```

| * | Addre... | Label | Mnemonics | Hexco... | Bytes | M-Cycl... | T-Stat... |
|---|---|---|---|---|---|---|---|
| √ | 1000 | | LDA 204F | 3A | 3 | 4 | 13 |
| | 1001 | | | 4F | | | |
| | 1002 | | | 20 | | | |
| √ | 1003 | | CPI 00 | FE | 2 | 2 | 7 |
| | 1004 | | | 00 | | | |
| √ | 1005 | | JZ END | CA | 3 | 3 | 10 |
| | 1006 | | | 1D | | | |
| | 1007 | | | 10 | | | |
| √ | 1008 | | MOV C,A | 4F | 1 | 1 | 4 |
| √ | 1009 | | LXI H,2050 | 21 | 3 | 3 | 10 |
| | 100A | | | 50 | | | |
| | 100B | | | 20 | | | |
| √ | 100C | | MOV B,M | 46 | 1 | 2 | 7 |
| √ | 100D | | INX H | 23 | 1 | 1 | 6 |
| √ | 100E | | DCR C | 0D | 1 | 1 | 4 |
| √ | 100F | | JZ END | CA | 3 | 3 | 10 |
| | 1010 | | | 1D | | | |
| | 1011 | | | 10 | | | |
| √ | 1012 | LOOP | MOV A,M | 7E | 1 | 2 | 7 |
| √ | 1013 | | CMP B | B8 | 1 | 1 | 4 |
| √ | 1014 | | JC SKIP | DA | 3 | 3 | 10 |
| | 1015 | | | 18 | | | |
| | 1016 | | | 10 | | | |
| √ | 1017 | | MOV B,A | 47 | 1 | 1 | 4 |

| 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 | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register H | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register L | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Memory(M) | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Resister | Value | S | Z | * | AC | * | P | * | CY |
|---|---|---|---|---|---|---|---|---|---|
| Flag Resister | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Type | Value |
|---|---|
| Stack Pointer(SP) | 0000 |
| Memory Pointer (HL) | 0000 |
| Program Status Word(PSW) | 0000 |
| Program Counter(PC) | 0000 |
| Clock Cycle Counter | 0 |
| Instruction Counter | 0 |

| 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 | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

For RIM instruction

| SID | I7.5 | I6.5 | I5.5 | IE | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Hexadecimal | Decimal | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Memory Range: 1000 ---- FFFF

Memory Editor

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1000 | 3A | 4F | 20 | FE | 00 | CA | 1D | 10 | 4F | 21 | 50 | 20 | 46 | 23 | 0D | CA |
| 1010 | 1D | 10 | 7E | B8 | DA | 18 | 10 | 47 | 23 | 0D | C2 | 12 | 10 | 78 | 32 | 4E |
| 1020 | 20 | 76 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1030 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1040 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1090 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10C0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10E0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

● Show entire memory content

○ Show only loaded memory location

○ Store directly to specified memory location

I/O Port Editor

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 30 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 40 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 50 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 60 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 70 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 80 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 90 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| C0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| E0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

Sample input  and output (M[204F H]=06 H, 6 numbers 6A,10,45,7E,B6, and DF,Starting from M[2050 H],

M[204E H]=10

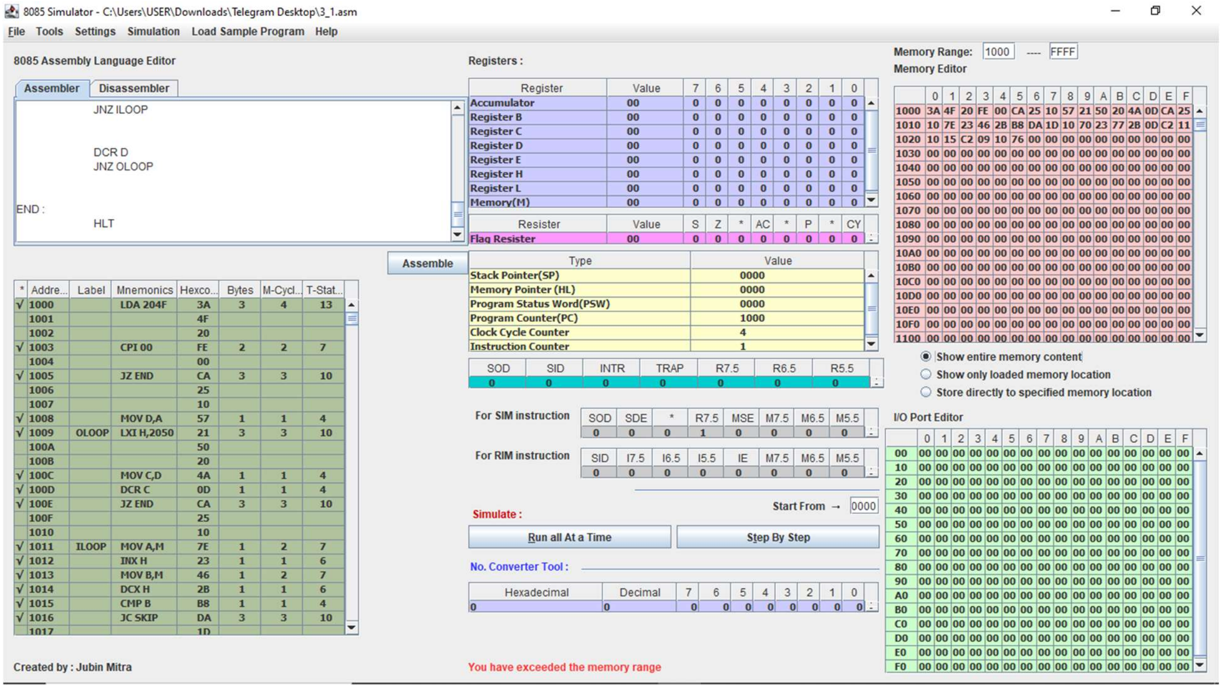
iii)Sort the N numbers in ascending order.

We will be using bubble sort algorithm.

| SL NO. | ADDRESS | OPCODE IN HEX CODE | LABEL | INTRUCTIONS | COMMENTS |
|---|---|---|---|---|---|
| 1 | 1000 | 3A | | LDA 204F H | Get N |
| | 1001 | 4F | | | |
| | 1002 | 20 | | | |
| 2 | 1003 | FE | | CPI 00 H | Is N =? |
| | 1004 | 00 | | | |
| 3 | 1005 | CA | | JZ END | Is N =0,nothing to do |
| | 1006 | 25 | | | |
| | 1007 | 10 | | | |
| 4 | 1008 | 57 | | MOV D,A | D=outer loop counter |
| 5 | 1009 | 21 | OLOOP | LXI H ,2050 | Contents of memory location 2050 H into HL register pair |
| | 100A | 50 | | | |
| | 100B | 20 | | | |
| 6 | 100C | 4A | | MOV C,D | C=inner loop counter |

| | | | | | |
|---|---|---|---|---|---|
| 7 | 100D | 0D | | DCR C | Numbers of comparisons is 1 less than the length |
| 8 | 100E | CA | | JZ END | If no comparisons are to be made ,do nothing |
| | 100F | 25 | | | |
| | 1010 | 10 | | | |
| 9 | 1011 | 7E | ILOOP | MOV A,M | Get first number |
| 10 | 1012 | 23 | | INX H | Go to next number |
| 11 | 1013 | 46 | | MOV B,M | Get second number |
| 12 | 1014 | 2B | | DCX H | Go back to current position |
| 13 | 1015 | B8 | | CMP B | Compare 2nd number against 1st number |
| 14 | 1016 | DA | | JC SKIP | If 2nd number>1st number, do nothing |
| | 1017 | 1D | | | |
| | 1018 | 10 | | | |
| 15 | 1019 | 70 | | MOV M,B | Put 2nd number first |
| 16 | 101A | 13 | | INX H | Go to next location |
| 17 | 101B | 77 | | MOV M,A | Put 1st number second |
| 18 | 101C | 2B | | DCX H | Go back to previous location |
| 19 | 101D | 0D | SKIP | DCR C | Comparison done |
| 20 | 101E | C2 | | JNZ ILOOP | Start from the next location |
| | 101F | 11 | | | |
| | 1020 | 10 | | | |
| 21 | 1021 | 15 | | DCR D | One pass finished |
| 22 | 1022 | C2 | | JNZ OLOOP | Go to the next pass |
| | 1023 | 09 | | | |
| | 1024 | 10 | | | |
| 23 | 1025 | 76 | | HLT | Stop the program |

Simulator with loaded machine code:


8085 Simulator - C:\Users\USER\Downloads\Telegram Desktop\3_1.asm
File Tools Settings Simulation Load Sample Program Help
Assembler
Disassembler
Assemble
Run all At a Time
Step By Step

8085 Assembly Language Editor

```
               JNZ ILOOP


               DCR D
               JNZ OLOOP


END :
               HLT
```

| * | Addre... | Label | Mnemonics | Hexco... | Bytes | M-Cycl... | T-Stat... |
|---|---|---|---|---|---|---|---|
| √ | 1000 | | LDA 204F | 3A | 3 | 4 | 13 |
| | 1001 | | | 4F | | | |
| | 1002 | | | 20 | | | |
| √ | 1003 | | CPI 00 | FE | 2 | 2 | 7 |
| | 1004 | | | 00 | | | |
| √ | 1005 | | JZ END | CA | 3 | 3 | 10 |
| | 1006 | | | 25 | | | |
| | 1007 | | | 10 | | | |
| √ | 1008 | | MOV D,A | 57 | 1 | 1 | 4 |
| √ | 1009 | OLOOP | LXI H,2050 | 21 | 3 | 3 | 10 |
| | 100A | | | 50 | | | |
| | 100B | | | 20 | | | |
| √ | 100C | | MOV C,D | 4A | 1 | 1 | 4 |
| √ | 100D | | DCR C | 0D | 1 | 1 | 4 |
| √ | 100E | | JZ END | CA | 3 | 3 | 10 |
| | 100F | | | 25 | | | |
| | 1010 | | | 10 | | | |
| √ | 1011 | ILOOP | MOV A,M | 7E | 1 | 2 | 7 |
| √ | 1012 | | INX H | 23 | 1 | 1 | 6 |
| √ | 1013 | | MOV B,M | 46 | 1 | 2 | 7 |
| √ | 1014 | | DCX H | 2B | 1 | 1 | 6 |
| √ | 1015 | | CMP B | B8 | 1 | 1 | 4 |
| √ | 1016 | | JC SKIP | DA | 3 | 3 | 10 |
| | 1017 | | | 1D | | | |

Created by : Jubin Mitra

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 | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register H | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register L | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Memory(M) | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Resister | Value | S | Z | * | AC | * | P | * | CY |
|---|---|---|---|---|---|---|---|---|---|
| Flag Resister | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Type | Value |
|---|---|
| Stack Pointer(SP) | 0000 |
| Memory Pointer (HL) | 0000 |
| Program Status Word(PSW) | 0000 |
| Program Counter(PC) | 1000 |
| Clock Cycle Counter | 4 |
| Instruction Counter | 1 |

| 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 | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

For RIM instruction

| SID | I7.5 | I6.5 | I5.5 | IE | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Start From → 0000

Simulate :

No. Converter Tool :

| Hexadecimal | Decimal | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

You have exceeded the memory range

Memory Range: 1000 ---- FFFF

Memory Editor

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1000 | 3A | 4F | 20 | FE | 00 | CA | 25 | 10 | 57 | 21 | 50 | 20 | 4A | 0D | CA | 25 |
| 1010 | 10 | 7E | 23 | 46 | 2B | B8 | DA | 1D | 10 | 70 | 23 | 77 | 2B | 0D | C2 | 11 |
| 1020 | 10 | 15 | C2 | 09 | 10 | 76 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1030 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1040 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1090 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10C0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10E0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

● Show entire memory content
○ Show only loaded memory location
○ Store directly to specified memory location

I/O Port Editor

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 30 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 40 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 50 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 60 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 70 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 80 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 90 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| C0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| E0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

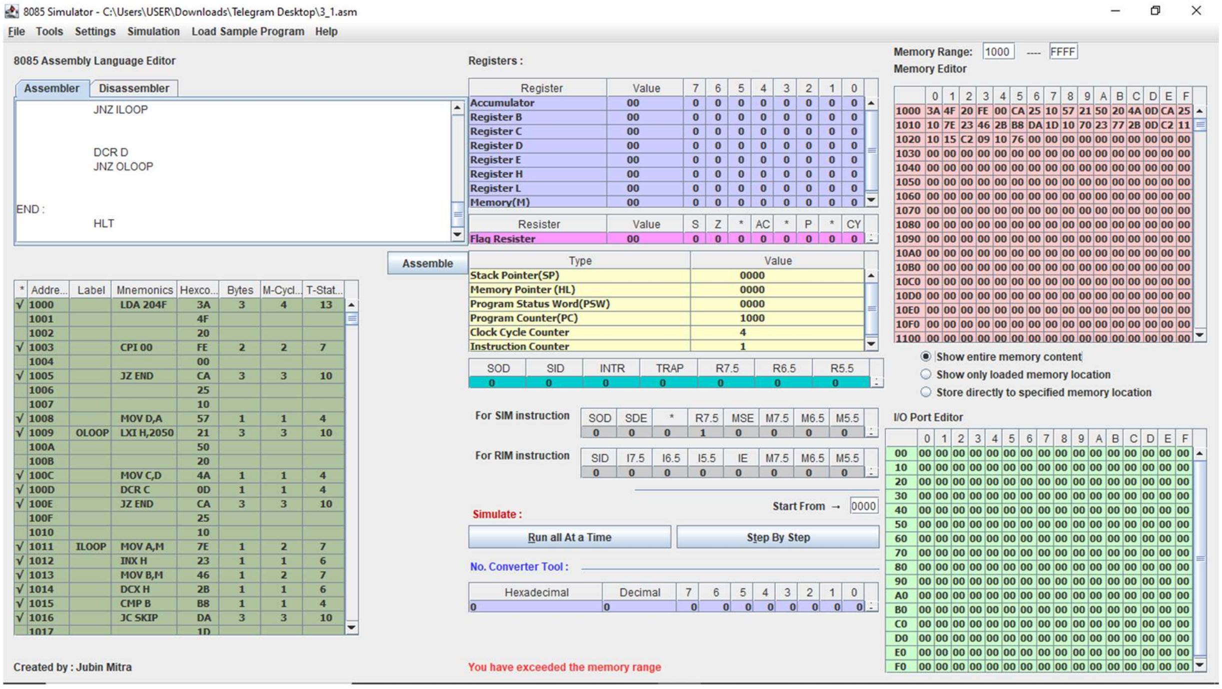
iv)Sort the N numbers in descending order.

We will be using bubble sort algorithm.

| SL NO. | ADDRESS | OPCODE IN HEX CODE | LABEL | INTRUCTIONS | COMMENTS |
|---|---|---|---|---|---|
| 1 | 1000 | 3A | | LDA 204F H | Get N |
| | 1001 | 4F | | | |
| | 1002 | 20 | | | |
| 2 | 1003 | FE | | CPI 00 H | Is N =? |
| | 1004 | 00 | | | |
| 3 | 1005 | CA | | JZ END | Is N =0,nothing to do |
| | 1006 | 25 | | | |
| | 1007 | 10 | | | |
| 4 | 1008 | 57 | | MOV D,A | D=outer loop counter |
| 5 | 1009 | 21 | OLOOP | LXI H ,2050 | Contents of memory location 2050 H into HL register pair |
| | 100A | 50 | | | |

| | | | | | |
|---|---|---|---|---|---|
| | 100B | 20 | | | |
| 6 | 100C | 4A | | MOV C,D | C=inner loop counter |
| 7 | 100D | 0D | | DCR C | Numbers of comparisons is 1 less than the length(D) |
| 8 | 100E | CA | | JZ END | If no comparisons are to be made ,do nothing |
| | 100F | 25 | | | |
| | 1010 | 10 | | | |
| 9 | 1011 | 7E | ILOOP | MOV A,M | Get first number |
| 10 | 1012 | 23 | | INX H | Go to next number |
| 11 | 1013 | 46 | | MOV B,M | Get second number |
| 12 | 1014 | 2B | | DCX H | Go back to current position |
| 13 | 1015 | B8 | | CMP B | Compare 2nd number against 1st number |
| 14 | 1016 | DA | | JC SKIP | If 2nd number<1st number, do nothing |
| | 1017 | 1D | | | |
| | 1018 | 10 | | | |
| 15 | 1019 | 70 | | MOV M,B | Put 2nd number first |
| 16 | 101A | 13 | | INX H | Go to next location |
| 17 | 101B | 77 | | MOV M,A | Put 1st number second |
| 18 | 101C | 2B | | DCX H | Go back to previous location |
| 19 | 101D | 0D | SKIP | DCR C | Comparison done |
| 20 | 101E | C2 | | JNZ ILOOP | Start from the next location |
| | 101F | 11 | | | |
| | 1020 | 10 | | | |
| 21 | 1021 | 15 | | DCR D | One pass finished |
| 22 | 1022 | C2 | | JNZ OLOOP | Go to the next pass |
| | 1023 | 09 | | | |
| | 1024 | 10 | | | |
| 23 | 1025 | 76 | | HLT | Stop the program |

Simulator with loaded machine code:



8085 Simulator - C:\Users\USER\Downloads\Telegram Desktop\3_1.asm
File Tools Settings Simulation Load Sample Program Help
8085 Assembly Language Editor
Assembler
Disassembler
Assemble
Registers :
Simulate :
Run all At a Time
Step By Step
No. Converter Tool :
Created by : Jubin Mitra

```
                JNZ ILOOP


                DCR D
                JNZ OLOOP


END :
                HLT
```

| * | Addre... | Label | Mnemonics | Hexco... | Bytes | M-Cycl... | T-Stat... |
|---|---|---|---|---|---|---|---|
| √ | 1000 | | LDA 204F | 3A | 3 | 4 | 13 |
| | 1001 | | | 4F | | | |
| | 1002 | | | 20 | | | |
| √ | 1003 | | CPI 00 | FE | 2 | 2 | 7 |
| | 1004 | | | 00 | | | |
| √ | 1005 | | JZ END | CA | 3 | 3 | 10 |
| | 1006 | | | 25 | | | |
| | 1007 | | | 10 | | | |
| √ | 1008 | | MOV D,A | 57 | 1 | 1 | 4 |
| √ | 1009 | OLOOP | LXI H,2050 | 21 | 3 | 3 | 10 |
| | 100A | | | 50 | | | |
| | 100B | | | 20 | | | |
| √ | 100C | | MOV C,D | 4A | 1 | 1 | 4 |
| √ | 100D | | DCR C | 0D | 1 | 1 | 4 |
| √ | 100E | | JZ END | CA | 3 | 3 | 10 |
| | 100F | | | 25 | | | |
| | 1010 | | | 10 | | | |
| √ | 1011 | ILOOP | MOV A,M | 7E | 1 | 2 | 7 |
| √ | 1012 | | INX H | 23 | 1 | 1 | 6 |
| √ | 1013 | | MOV B,M | 46 | 1 | 2 | 7 |
| √ | 1014 | | DCX H | 2B | 1 | 1 | 6 |
| √ | 1015 | | CMP B | B8 | 1 | 1 | 4 |
| √ | 1016 | | JC SKIP | DA | 3 | 3 | 10 |
| | 1017 | | | 1D | | | |

| 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 | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register H | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register L | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Memory(M) | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Resister | Value | S | Z | * | AC | * | P | * | CY |
|---|---|---|---|---|---|---|---|---|---|
| Flag Resister | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Type | Value |
|---|---|
| Stack Pointer(SP) | 0000 |
| Memory Pointer (HL) | 0000 |
| Program Status Word(PSW) | 0000 |
| Program Counter(PC) | 1000 |
| Clock Cycle Counter | 4 |
| Instruction Counter | 1 |

| 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 | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

For RIM instruction

| SID | I7.5 | I6.5 | I5.5 | IE | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Start From → 0000

| Hexadecimal | Decimal | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

You have exceeded the memory range

Memory Range: 1000 ---- FFFF

Memory Editor

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1000 | 3A | 4F | 20 | FE | 00 | CA | 25 | 10 | 57 | 21 | 50 | 20 | 4A | 0D | CA | 25 |
| 1010 | 10 | 7E | 23 | 46 | 2B | B8 | DA | 1D | 10 | 70 | 23 | 77 | 2B | 0D | C2 | 11 |
| 1020 | 10 | 15 | C2 | 09 | 10 | 76 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1030 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1040 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1090 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10C0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10E0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

● Show entire memory content
○ Show only loaded memory location
○ Store directly to specified memory location

I/O Port Editor

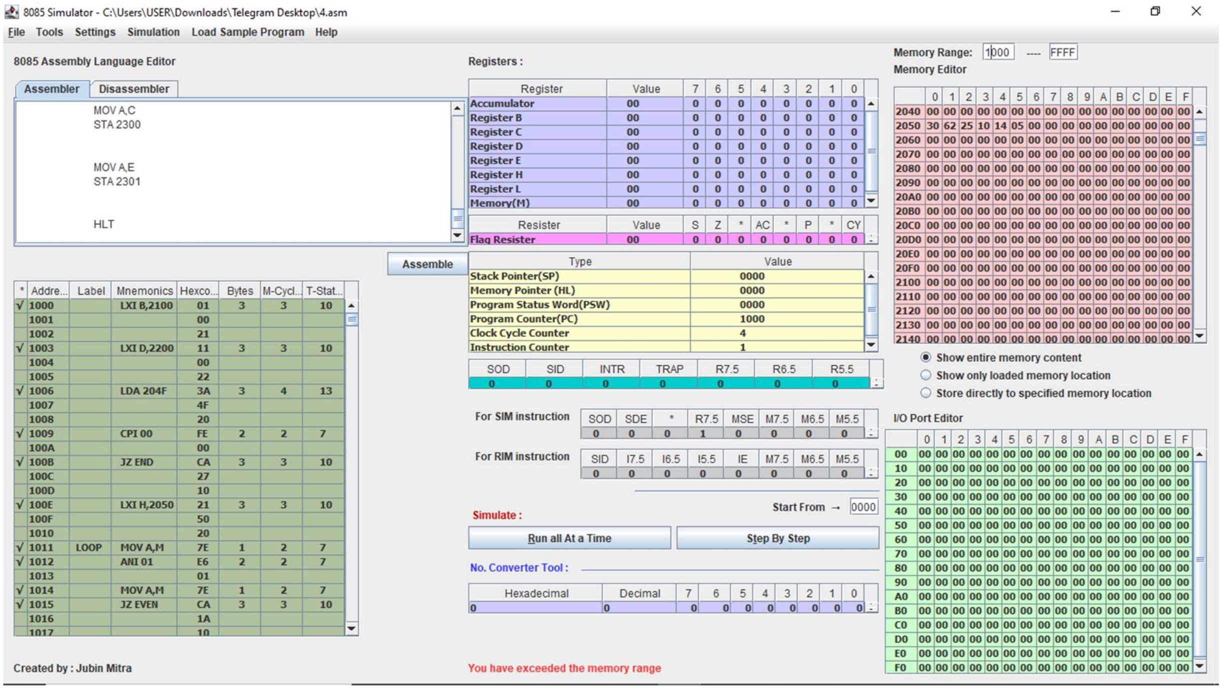
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 30 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 40 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 50 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 60 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 70 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 80 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 90 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| C0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| E0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

4.N numbers are stored in consecutive m/m location startingfrom 2050 H. The value N is stored in 204F H. Write a program to copy the even and odd numbers starting from 2100 H and 2200 H ,respectively. Store the total num of even and odd numbers in 2300 H and 2301 H , respectively

| SL NO. | ADDRESS | OPCODE IN HEX CODE | LABEL | INSTRUCTIONS | COMMENTS |
|---|---|---|---|---|---|
| 1 | 1000 | 01 | | LXI B , 2100 | Starting address of even numbers |
| | 1001 | 00 | | | |
| | 1002 | 21 | | | |
| 2 | 1003 | 11 | | LXI D,2200 | Starting address of odd numbers |
| | 1004 | 00 | | | |
| | 1005 | 22 | | | |
| 3 | 1006 | 3A | | LDA 204F | Get N |
| | 1007 | 4F | | | |
| | 1008 | 20 | | | |
| 4 | 1009 | FE | | CPI 00 H | Is N =0? |
| | 100A | 00 | | | |
| 5 | 100B | CA | | JZ END | If N=0, nothing to do |

| | | | | | |
|---|---|---|---|---|---|
| | 100C | 27 | | | |
| | 100D | 10 | | | |
| 6 | 100E | 21 | | LXI H ,2050 | Contents of memory location 2050 H into HL register pair/input |
| | 100F | 50 | | | |
| | 1010 | 20 | | | |
| 7 | 1011 | 7E | LOOP | MOV A, M | Get current numbers |
| 8 | 1012 | E6 | | ANI 01 | Check odd or not |
| | 1013 | 01 | | | |
| 9 | 1014 | 7E | | MOV A, M | Restore the number |
| 10 | 1015 | CA | | JZ EVEN | If 0, then even |
| | 1016 | 1A | | | |
| | 1017 | 10 | | | |
| 11 | 1018 | 12 | | STAX D | This is an odd number |
| 12 | 1019 | 13 | | INX D | One odd number added |
| 13 | 101A | 02 | EVEN | STAX B | This is an even number |
| 14 | 101B | 03 | | INX B | One even number added |
| 15 | 101C | 23 | LEND | INX H | Go to the next address |
| 16 | 101D | 3A | | LDA 204F | Get n |
| | 101E | 4F | | | |
| | 101F | 20 | | | |
| 17 | 1020 | 3D | | DCR A | One number checked |
| 18 | 1021 | 32 | | STA 204F | Store N for later use |
| | 1022 | 4F | | | |
| | 1023 | 20 | | | |
| 19 | 1024 | C2 | | JNZ LOOP | If numbers left ,continue |
| | 1025 | 11 | | | |
| | 1026 | 10 | | | |
| 20 | 1027 | 79 | END | MOV A,C | A=C=number of even number because BC started from 2100 H |
| 21 | 1028 | 32 | | STA 2300 H | Store numbers of even number |
| | 1029 | 00 | | | |
| | 102A | 23 | | | |
| 22 | 102B | 7B | | MOV A,E | A=E=number of odd numbers because DE started from 2200 H |
| 23 | 102C | 32 | | STA 2301 H | Store number of odd numbers |
| | 102D | 01 | | | |
| | 102E | 23 | | | |
| 24 | 102F | 76 | | HLT | Stop the program |

Simulator with loaded machine code:


8085 Simulator - C:\Users\USER\Downloads\Telegram Desktop\4.asm
File Tools Settings Simulation Load Sample Program Help
8085 Assembly Language Editor
Assembler
Disassembler
Assemble
Registers :
Simulate :
Run all At a Time
Step By Step
No. Converter Tool :
Created by : Jubin Mitra

```
        MOV A,C
        STA 2300


        MOV A,E
        STA 2301


        HLT
```

| * | Addre... | Label | Mnemonics | Hexco... | Bytes | M-Cycl... | T-Stat... |
|---|---|---|---|---|---|---|---|
| √ | 1000 | | LXI B,2100 | 01 | 3 | 3 | 10 |
| | 1001 | | | 00 | | | |
| | 1002 | | | 21 | | | |
| √ | 1003 | | LXI D,2200 | 11 | 3 | 3 | 10 |
| | 1004 | | | 00 | | | |
| | 1005 | | | 22 | | | |
| √ | 1006 | | LDA 204F | 3A | 3 | 4 | 13 |
| | 1007 | | | 4F | | | |
| | 1008 | | | 20 | | | |
| √ | 1009 | | CPI 00 | FE | 2 | 2 | 7 |
| | 100A | | | 00 | | | |
| √ | 100B | | JZ END | CA | 3 | 3 | 10 |
| | 100C | | | 27 | | | |
| | 100D | | | 10 | | | |
| √ | 100E | | LXI H,2050 | 21 | 3 | 3 | 10 |
| | 100F | | | 50 | | | |
| | 1010 | | | 20 | | | |
| √ | 1011 | LOOP | MOV A,M | 7E | 1 | 2 | 7 |
| √ | 1012 | | ANI 01 | E6 | 2 | 2 | 7 |
| | 1013 | | | 01 | | | |
| √ | 1014 | | MOV A,M | 7E | 1 | 2 | 7 |
| √ | 1015 | | JZ EVEN | CA | 3 | 3 | 10 |
| | 1016 | | | 1A | | | |
| | 1017 | | | 10 | | | |

| 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 | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register H | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register L | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Memory(M) | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Resister | Value | S | Z | * | AC | * | P | * | CY |
|---|---|---|---|---|---|---|---|---|---|
| Flag Resister | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Type | Value |
|---|---|
| Stack Pointer(SP) | 0000 |
| Memory Pointer (HL) | 0000 |
| Program Status Word(PSW) | 0000 |
| Program Counter(PC) | 1000 |
| Clock Cycle Counter | 4 |
| Instruction Counter | 1 |

| 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 | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

For RIM instruction

| SID | I7.5 | I6.5 | I5.5 | IE | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Start From → 0000

| Hexadecimal | Decimal | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

You have exceeded the memory range

Memory Range: 1000 ---- FFFF

Memory Editor

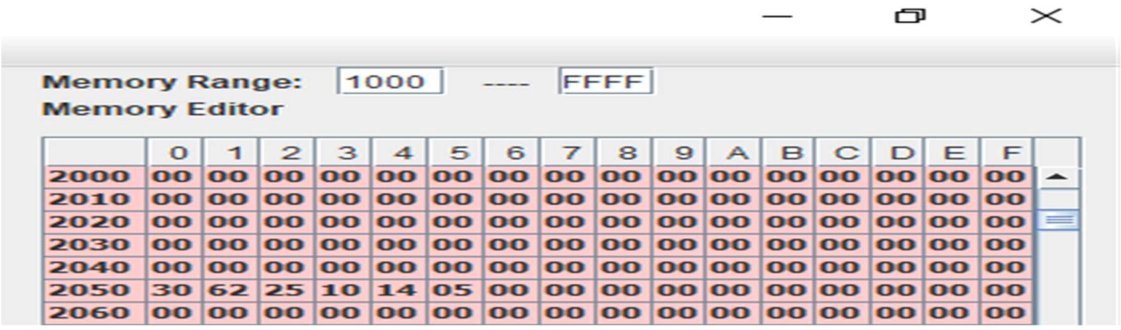
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2040 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2050 | 30 | 62 | 25 | 10 | 14 | 05 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2090 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20C0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20E0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2110 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2120 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2130 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2140 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

● Show entire memory content
○ Show only loaded memory location
○ Store directly to specified memory location

I/O Port Editor

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 30 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 40 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 50 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 60 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 70 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 80 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 90 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| C0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| E0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

Sample input (M[204F H]=06 H, 6 numbers 30 H,62 H,25 H,10 H,24 H and 05 H starting from M[2050 H]):

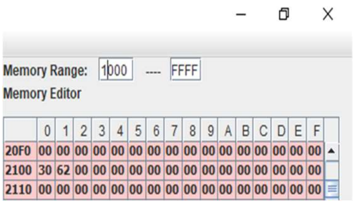


Memory Range: 1000 ---- FFFF

Memory Editor

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2000 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2010 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2020 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2030 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2040 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2050 | 30 | 62 | 25 | 10 | 14 | 05 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

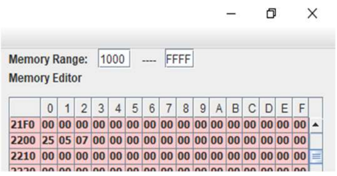
Sample output (2 even numbers 30 H, 62 H starting from M[2100 H], 3 odd numbers 25 H,05 H, 07 H starting from M[2200 H], M[2300 H]=02 H,M[2301]=03 H):



Memory Range: 1000 ---- FFFF

Memory Editor

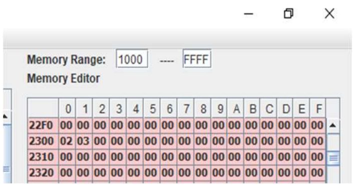
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 20F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2100 | 30 | 62 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2110 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |



Memory Range: 1000 ---- FFFF

Memory Editor

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 21F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2200 | 25 | 05 | 07 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2210 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |



Memory Range: 1000 ---- FFFF

Memory Editor

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 22F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2300 | 02 | 03 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2310 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2320 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

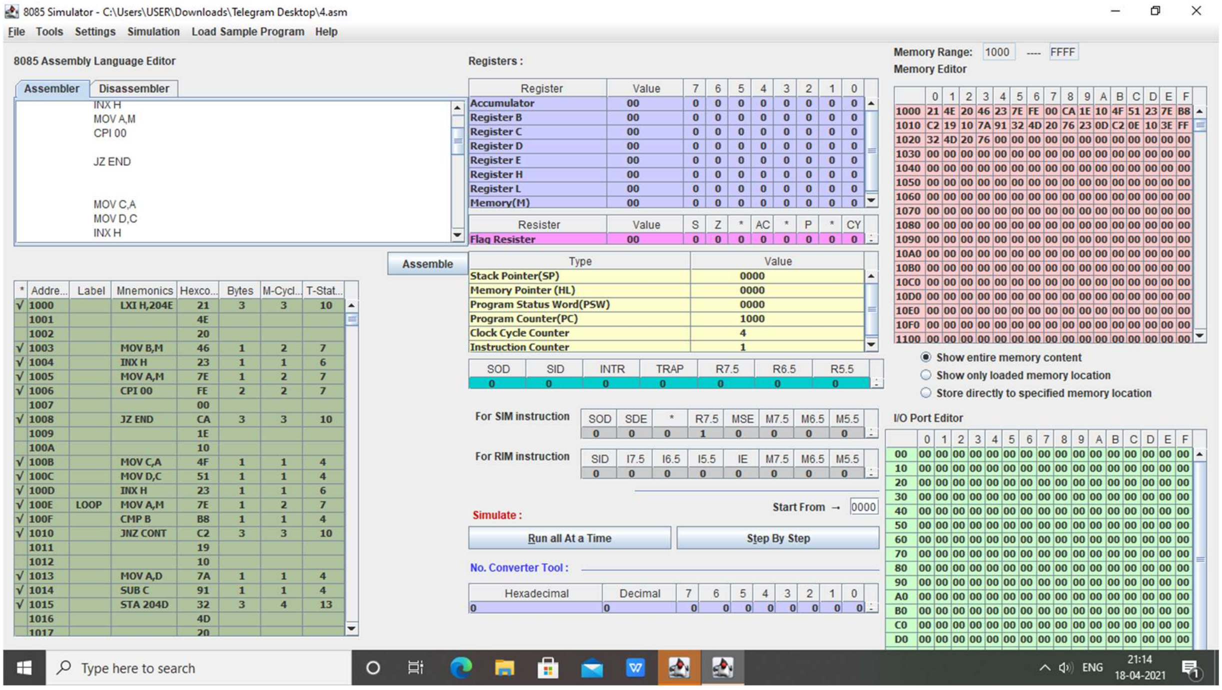
5. N numbers are stored in consecutive m/m location starting from 2050 H. The value N is stored in 204F H. Write a program to test whether a number stored in 204E H is present in the list. If present , store its position in the list at 204D H ;otherwise store FF H.

We will give position by a zero – based index.

| SL NO. | ADDRESS | OPCODE IN HEX CODE | LABEL | INSTRUCTIONS | COMMENTS |
|---|---|---|---|---|---|
| 1 | 1000 | 21 | | LXI H, 204E | B=Numbers to search (key) |
| | 1001 | 4E | | | |
| | 1002 | 20 | | | |
| 2 | 1003 | 46 | | MOV B,M | Move memory location to B register |
| 3 | 1004 | 23 | | INX H | HL=204F H |
| 4 | 1005 | 7E | | MOV A,M | Get N |
| 5 | 1006 | FE | | CPI 00 H | Is N=0? |
| | 1007 | 00 | | | |
| 6 | 1008 | CA | | JZ END | If N=0,do not search |
| | 1009 | 1E | | | |
| | 100A | 10 | | | |
| 7 | 100B | 4F | | MOV C,A | Move accumulator to C register [C=A=Counter] |
| 8 | 100C | 51 | | MOV D,C | Move C register to D register[D=total no of items] |
| 9 | 100D | 23 | | INX H | HL=2050 H=Starting address of input |
| 10 | 100E | 7E | LOOP | MOV A,M | Move memory address to accumulator, load current number |
| 11 | 100F | B8 | | CMP B | Compare against key |
| 12 | 1010 | C2 | | JNZ CONT | If unequal ,continue loop |
| | 1011 | 19 | | | |
| | 1012 | 10 | | | |
| 13 | 1013 | 7A | | MOV A,D | A=D=Totals number of items |
| 14 | 1014 | 91 | | SUB C | A=position of key in list |
| 15 | 1015 | 32 | | STA 204D | Store found position |
| | 1016 | 4D | | | |
| | 1017 | 20 | | | |
| 16 | 1018 | 76 | | HLT | We have nothing to do anymore |
| 17 | 1019 | 23 | | INX H | Go to next number |

| | | | | | |
|---|---|---|---|---|---|
| 18 | 101A | 0D | | DCR C | Checked one number |
| 19 | 101B | C2 | | JNZ LOOP | If number left,continue |
| | 101C | 0E | | | |
| | 101D | 10 | | | |
| 20 | 101E | 3E | END | MVI A,FF | We haven’t found key |
| | 101F | FF | | | |
| 21 | 1020 | 32 | | STA 204D | Store the number |
| | 1021 | 4D | | | |
| | 1022 | 20 | | | |
| 22 | 1023 | 76 | | HLT | Stop the program. |

Simulator with loaded machine code:



```
        INX H
        MOV A,M
        CPI 00

        JZ END


        MOV C,A
        MOV D,C
        INX H
```

| * | Addre... | Label | Mnemonics | Hexco... | Bytes | M-Cycl... | T-Stat... |
|---|---|---|---|---|---|---|---|
| √ | 1000 | | LXI H,204E | 21 | 3 | 3 | 10 |
| | 1001 | | | 4E | | | |
| | 1002 | | | 20 | | | |
| √ | 1003 | | MOV B,M | 46 | 1 | 2 | 7 |
| √ | 1004 | | INX H | 23 | 1 | 1 | 6 |
| √ | 1005 | | MOV A,M | 7E | 1 | 2 | 7 |
| √ | 1006 | | CPI 00 | FE | 2 | 2 | 7 |
| | 1007 | | | 00 | | | |
| √ | 1008 | | JZ END | CA | 3 | 3 | 10 |
| | 1009 | | | 1E | | | |
| | 100A | | | 10 | | | |
| √ | 100B | | MOV C,A | 4F | 1 | 1 | 4 |
| √ | 100C | | MOV D,C | 51 | 1 | 1 | 4 |
| √ | 100D | | INX H | 23 | 1 | 1 | 6 |
| √ | 100E | LOOP | MOV A,M | 7E | 1 | 2 | 7 |
| √ | 100F | | CMP B | B8 | 1 | 1 | 4 |
| √ | 1010 | | JNZ CONT | C2 | 3 | 3 | 10 |
| | 1011 | | | 19 | | | |
| | 1012 | | | 10 | | | |
| √ | 1013 | | MOV A,D | 7A | 1 | 1 | 4 |
| √ | 1014 | | SUB C | 91 | 1 | 1 | 4 |
| √ | 1015 | | STA 204D | 32 | 3 | 4 | 13 |
| | 1016 | | | 4D | | | |
| | 1017 | | | 20 | | | |

| 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 | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register H | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Register L | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Memory(M) | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Resister | Value | S | Z | * | AC | * | P | * | CY |
|---|---|---|---|---|---|---|---|---|---|
| Flag Resister | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Type | Value |
|---|---|
| Stack Pointer(SP) | 0000 |
| Memory Pointer (HL) | 0000 |
| Program Status Word(PSW) | 0000 |
| Program Counter(PC) | 1000 |
| Clock Cycle Counter | 4 |
| Instruction Counter | 1 |

| 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 | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

For RIM instruction

| SID | I7.5 | I6.5 | I5.5 | IE | M7.5 | M6.5 | M5.5 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Hexadecimal | Decimal | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Memory Range: 1000 ---- FFFF

Memory Editor

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1000 | 21 | 4E | 20 | 46 | 23 | 7E | FE | 00 | CA | 1E | 10 | 4F | 51 | 23 | 7E | B8 |
| 1010 | C2 | 19 | 10 | 7A | 91 | 32 | 4D | 20 | 76 | 23 | 0D | C2 | 0E | 10 | 3E | FF |
| 1020 | 32 | 4D | 20 | 76 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1030 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1040 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1090 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10C0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10E0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 1100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

● Show entire memory content
○ Show only loaded memory location
○ Store directly to specified memory location

I/O Port Editor

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 20 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 30 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 40 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 50 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 60 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 70 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 80 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 90 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| C0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |