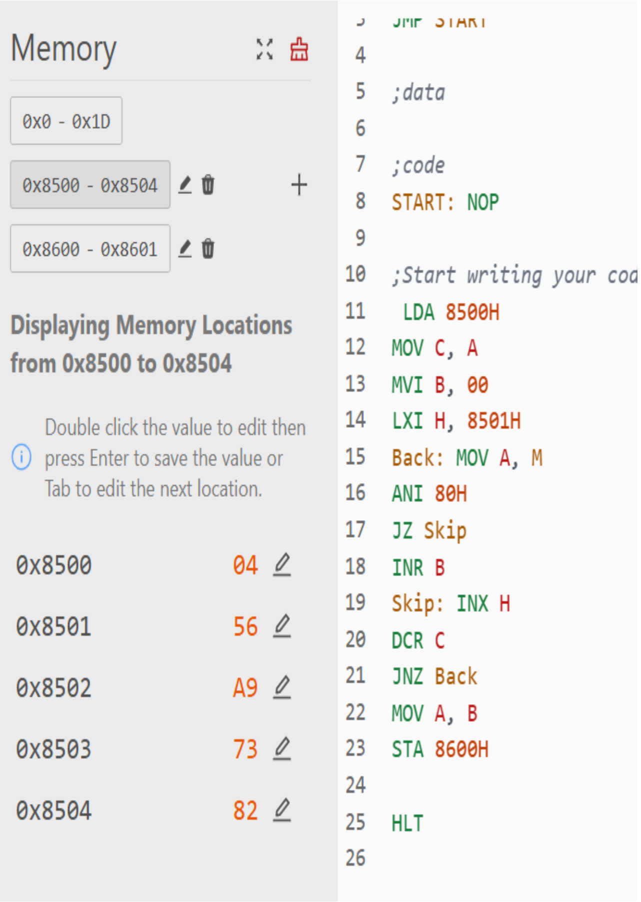
ASSIGNMENT-3

RHYTHM NARANG(102203839) YASHAV AGGARWAL(102203724)

Write a program to find the negative numbers in a block of data.

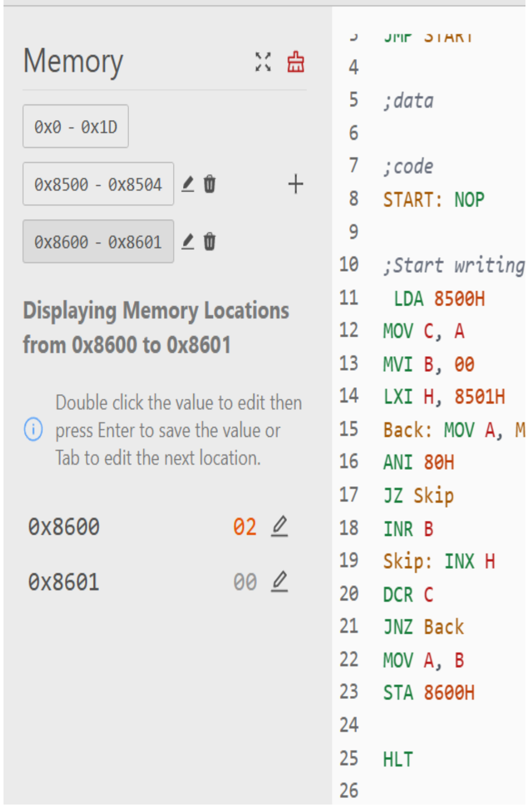


Displaying Memory Locations from 0x8500 to 0x8504

Double click the value to edit then press Enter to save the value or Tab to edit the next location.

0x8500	04
0x8501	56
0x8502	A9
0x8503	73
0x8504	82

```
JMP START

;data

;code
START: NOP

;Start writing your coo
 LDA 8500H
MOV C, A
MVI B, 00
LXI H, 8501H
Back: MOV A, M
ANI 80H
JZ Skip
INR B
Skip: INX H
DCR C
JNZ Back
MOV A, B
STA 8600H

HLT

```



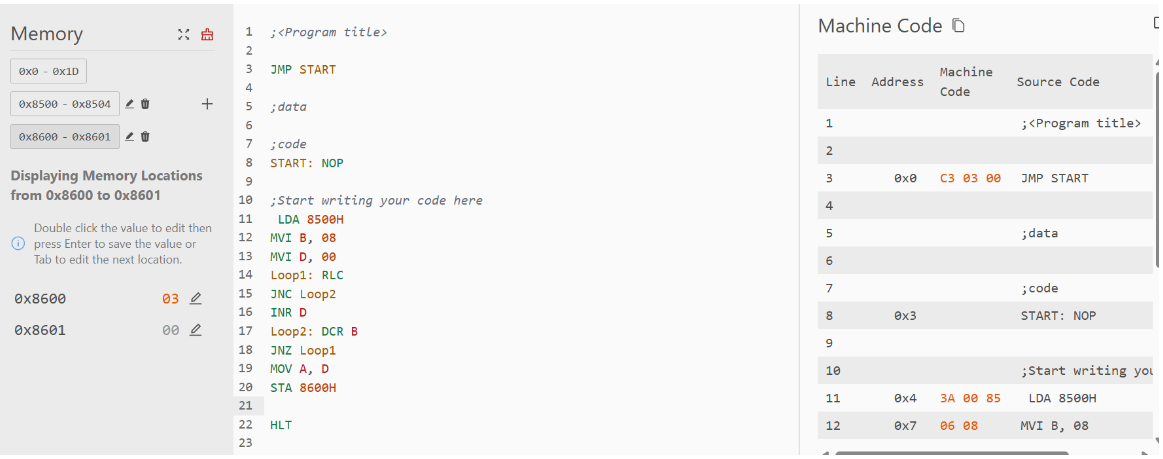
Displaying Memory Locations from 0x8600 to 0x8601

Double click the value to edit then press Enter to save the value or Tab to edit the next location.

0x8600	02
0x8601	00

```
JMP START

;data

;code
START: NOP

;Start writing
 LDA 8500H
MOV C, A
MVI B, 00
LXI H, 8501H
Back: MOV A, M
ANI 80H
JZ Skip
INR B
Skip: INX H
DCR C
JNZ Back
MOV A, B
STA 8600H

HLT

```

- **Write a program to count the number of one's in a number.**



Displaying Memory Locations from 0x8600 to 0x8601

Double click the value to edit then press Enter to save the value or Tab to edit the next location.

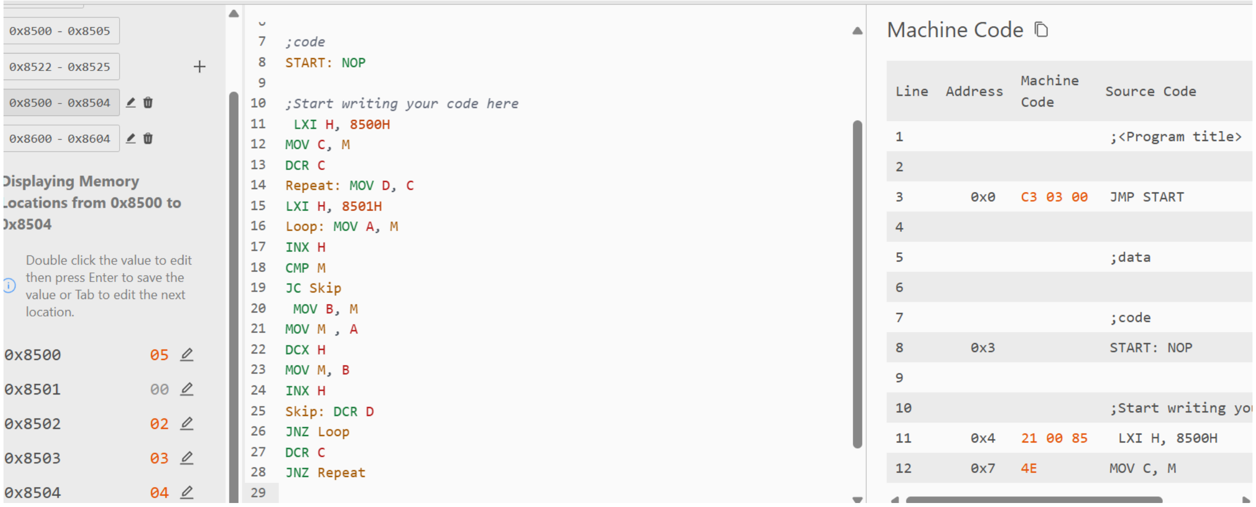
0x8600	03
0x8601	00

```
;<Program title>

JMP START

;data

;code
START: NOP

;Start writing your code here
 LDA 8500H
MVI B, 08
MVI D, 00
Loop1: RLC
JNC Loop2
INR D
Loop2: DCR B
JNZ Loop1
MOV A, D
STA 8600H

HLT

```

Machine Code

Line	Address	Machine Code	Source Code
1			;<Program title>
2			
3	0x0	C3 03 00	JMP START
4			
5			;data
6			
7			;code
8	0x3		START: NOP
9			
10			;Start writing you
11	0x4	3A 00 85	LDA 8500H
12	0x7	06 08	MVI B, 08

- **Write a program to arrange numbers in Ascending order. Calculate the sum of series of even numbers.**



0x8500 - 0x8505

0x8522 - 0x8525

0x8500 - 0x8504

0x8600 - 0x8604

Displaying Memory Locations from 0x8500 to 0x8504

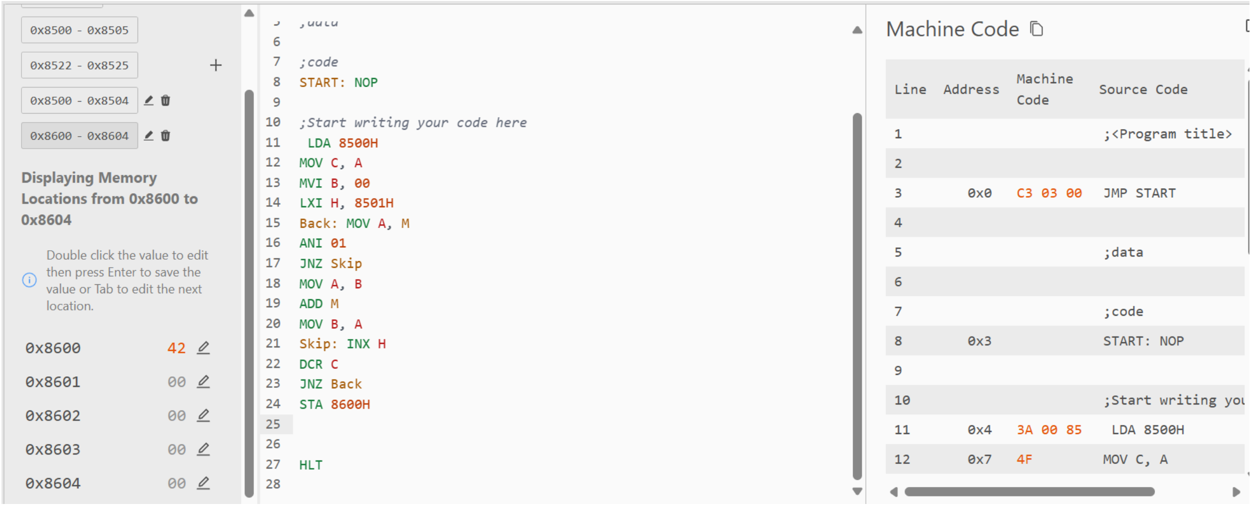
Double click the value to edit then press Enter to save the value or Tab to edit the next location.

0x8500	05
0x8501	00
0x8502	02
0x8503	03
0x8504	04

```
;code
START: NOP

;Start writing your code here
 LXI H, 8500H
MOV C, M
DCR C
Repeat: MOV D, C
LXI H, 8501H
Loop: MOV A, M
INX H
CMP M
JC Skip
 MOV B, M
MOV M , A
DCX H
MOV M, B
INX H
Skip: DCR D
JNZ Loop
DCR C
JNZ Repeat

```

Machine Code

Line	Address	Machine Code	Source Code
1			;<Program title>
2			
3	0x0	C3 03 00	JMP START
4			
5			;data
6			
7			;code
8	0x3		START: NOP
9			
10			;Start writing yo
11	0x4	21 00 85	LXI H, 8500H
12	0x7	4E	MOV C, M



0x8500 - 0x8505

0x8522 - 0x8525

0x8500 - 0x8504

0x8600 - 0x8604

Displaying Memory Locations from 0x8600 to 0x8604

Double click the value to edit then press Enter to save the value or Tab to edit the next location.

0x8600	42
0x8601	00
0x8602	00
0x8603	00
0x8604	00

```
;data

;code
START: NOP

;Start writing your code here
 LDA 8500H
MOV C, A
MVI B, 00
LXI H, 8501H
Back: MOV A, M
ANI 01
JNZ Skip
MOV A, B
ADD M
MOV B, A
Skip: INX H
DCR C
JNZ Back
STA 8600H


HLT

```

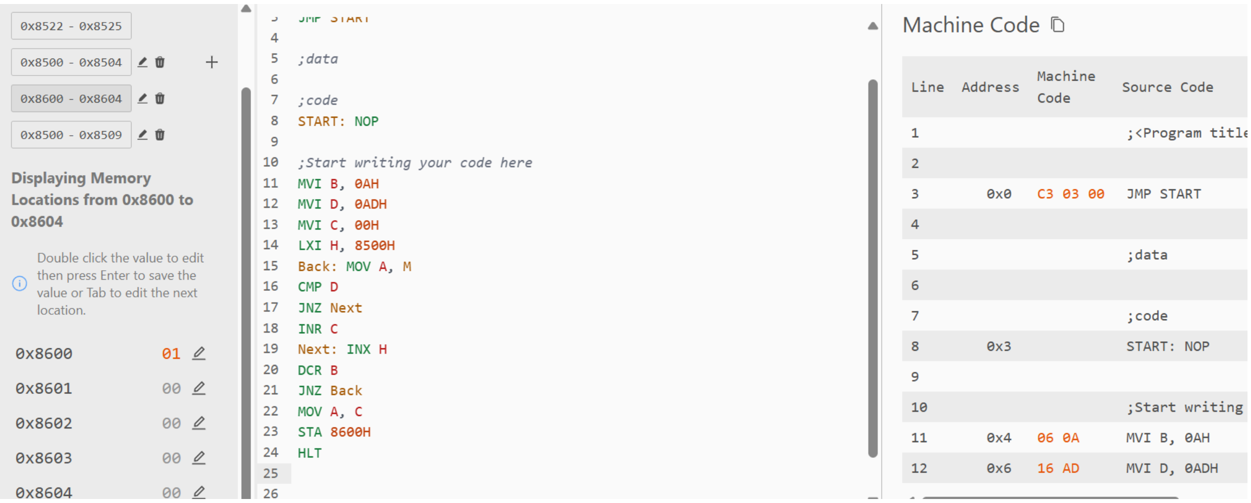
Machine Code

Line	Address	Machine Code	Source Code
1			;<Program title>
2			
3	0x0	C3 03 00	JMP START
4			
5			;data
6			
7			;code
8	0x3		START: NOP
9			
10			;Start writing you
11	0x4	3A 00 85	LDA 8500H
12	0x7	4F	MOV C, A

Input - [8500] – 04, [8501] – 20,
[8502] – 15 , [8503] – 13, [8504] – 22

- **Write an assembly language program to verify how many bytes are present in a given set, which resembles 10101101 in 8085.**

Input - [8500] – AD, [8501] – 01, [8502] – 01, [8503] – 01, [8504] – 01, [8505] – 01, [8506] – 01, [8507] – 01, [8508] – 01, [8509] – 01



0x8522 - 0x8525

0x8500 - 0x8504

0x8600 - 0x8604

0x8500 - 0x8509

Displaying Memory Locations from 0x8600 to 0x8604

ⓘ Double click the value to edit then press Enter to save the value or Tab to edit the next location.

0x8600	01
0x8601	00
0x8602	00
0x8603	00
0x8604	00

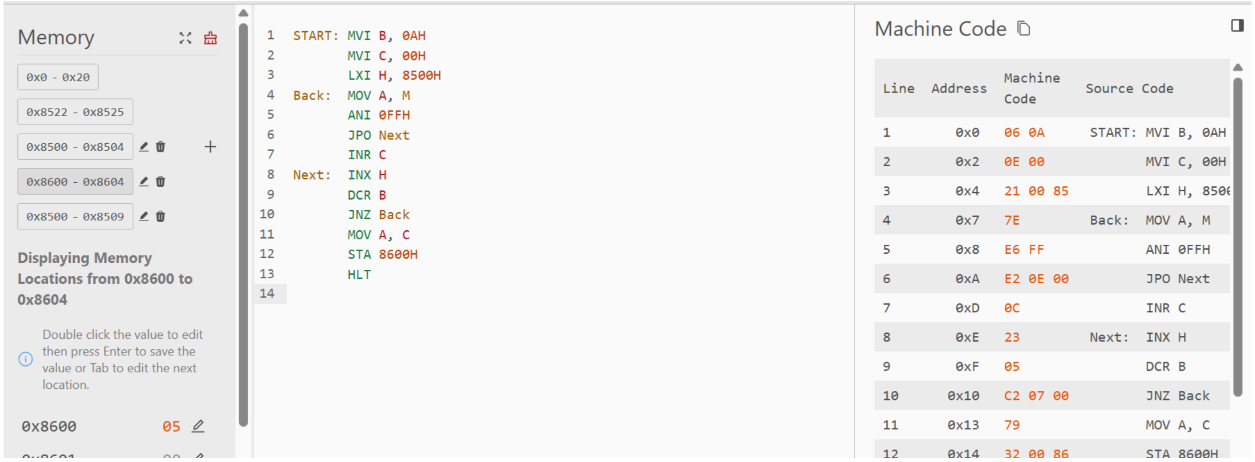
```
JMP START

;data

;code
START: NOP

;Start writing your code here
MVI B, 0AH
MVI D, 0ADH
MVI C, 00H
LXI H, 8500H
Back: MOV A, M
CMP D
JNZ Next
INR C
Next: INX H
DCR B
JNZ Back
MOV A, C
STA 8600H
HLT
```

Machine Code

Line	Address	Machine Code	Source Code
1			;<Program title
2			
3	0x0	C3 03 00	JMP START
4			
5			;data
6			
7			;code
8	0x3		START: NOP
9			
10			;Start writing
11	0x4	06 0A	MVI B, 0AH
12	0x6	16 AD	MVI D, 0ADH

- **Write an assembly language program to find the numbers of even parity in ten consecutive memory locations in 8085.**

Input - [8500] – 01, [8501] – 03, [8502] – 01, [8503] – 03, [8504] – 01, [8505] – 03, [8506] – 01, [8507] – 03, [8508] – 01, [8509] – 03



Memory

0x0 - 0x20

0x8522 - 0x8525

0x8500 - 0x8504

0x8600 - 0x8604

0x8500 - 0x8509

Displaying Memory Locations from 0x8600 to 0x8604

ⓘ Double click the value to edit then press Enter to save the value or Tab to edit the next location.

0x8600	05

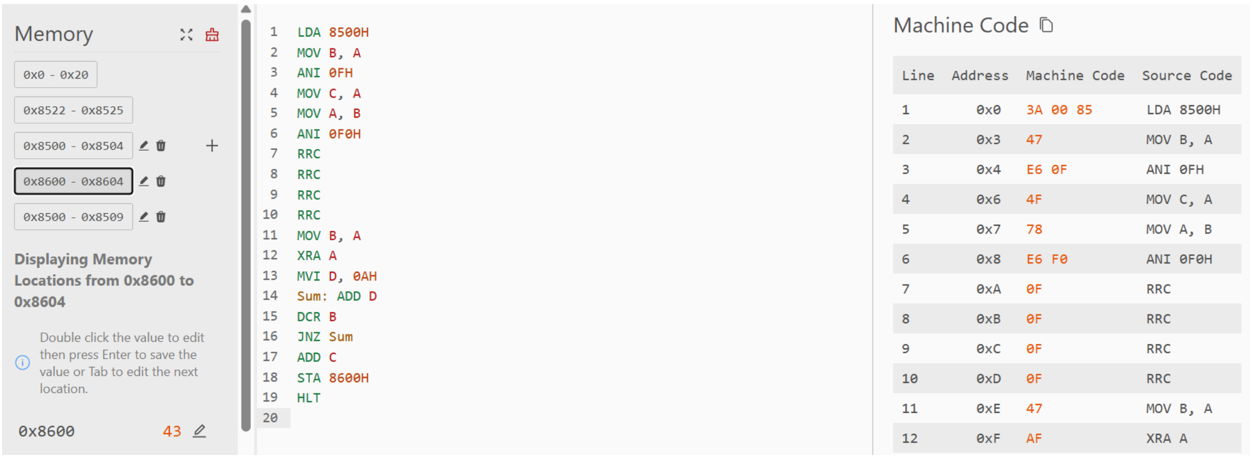
```
START: MVI B, 0AH
       MVI C, 00H
       LXI H, 8500H
Back:  MOV A, M
       ANI 0FFH
       JPO Next
       INR C
Next:  INX H
       DCR B
       JNZ Back
       MOV A, C
       STA 8600H
       HLT
```

Machine Code

Line	Address	Machine Code	Source Code
1	0x0	06 0A	START: MVI B, 0AH
2	0x2	0E 00	MVI C, 00H
3	0x4	21 00 85	LXI H, 850
4	0x7	7E	Back: MOV A, M
5	0x8	E6 FF	ANI 0FFH
6	0xA	E2 0E 00	JPO Next
7	0xD	0C	INR C
8	0xE	23	Next: INX H
9	0xF	05	DCR B
10	0x10	C2 07 00	JNZ Back
11	0x13	79	MOV A, C
12	0x14	32 00 86	STA 8600H

- **Write an assembly language program to convert a BCD number into its equivalent binary in 8085.**

Input - [8500] – 67
Input - [8600] – 43



Displaying Memory Locations from 0x8600 to 0x8604

Double click the value to edit then press Enter to save the value or Tab to edit the next location.

```
0x8600    43
```

```
LDA 8500H
MOV B, A
ANI 0FH
MOV C, A
MOV A, B
ANI 0F0H
RRC
RRC
RRC
RRC
MOV B, A
XRA A
MVI D, 0AH
Sum: ADD D
DCR B
JNZ Sum
ADD C
STA 8600H
HLT
```

Machine Code

Line	Address	Machine Code	Source Code
1	0x0	3A 00 85	LDA 8500H
2	0x3	47	MOV B, A
3	0x4	E6 0F	ANI 0FH
4	0x6	4F	MOV C, A
5	0x7	78	MOV A, B
6	0x8	E6 F0	ANI 0F0H
7	0xA	0F	RRC
8	0xB	0F	RRC
9	0xC	0F	RRC
10	0xD	0F	RRC
11	0xE	47	MOV B, A
12	0xF	AF	XRA A

- **Write an assembly language program for exchange the contents of memory location.**

LXI H, 8500H

MOV A, M

INX H

MOV B, M

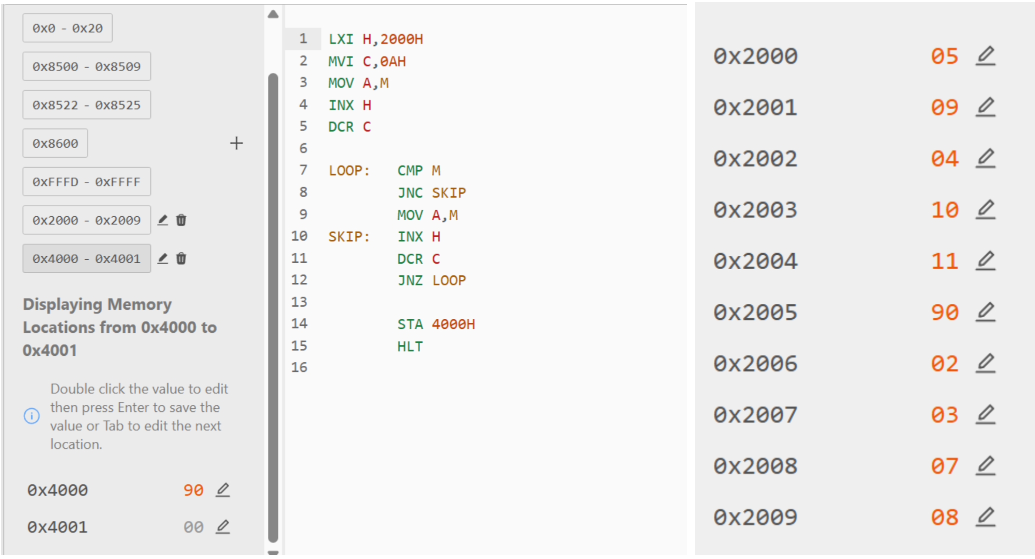
MOV M, A

DCX H

MOV M, B

HLT

- **Write a program to find the largest number in an array of 10 elements.**



Displaying Memory Locations from 0x4000 to 0x4001

Double click the value to edit then press Enter to save the value or Tab to edit the next location.

0x4000	90
0x4001	00

```
LXI H,2000H
MVI C,0AH
MOV A,M
INX H
DCR C

LOOP:   CMP M
        JNC SKIP
        MOV A,M
SKIP:   INX H
        DCR C
        JNZ LOOP

        STA 4000H
        HLT
```

0x2000	05
0x2001	09
0x2002	04
0x2003	10
0x2004	11
0x2005	90
0x2006	02
0x2007	03
0x2008	07
0x2009	08