**PRACTICAL 1**

**AIM: Write a program to execute all data transfer, arithmetic and logical, shift and all rotate instructions.**

**Data Transfer:**

LDA 0002H ; Load value from 0002H into AC

MOV C, A ; Copy from AC to C register

MVI D, 00H ; Store value 00H in D register

MVI E, 06H ; Store value 06H in E register

LDAX D ; Copy the contents pointed at by D to AC

LXI H, 2040H ; Load the value 2040H in HL register

STA 0004H ; Store the value of AC at 0004H

STAX B ; Contents of AC copied to the memory location specified by B

LHLD 000AH ; Copy the data from 000AH into HL register

HLT ; End of Program

**Input:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 0002H | 07 |
| 0006H | 09 |
| 000AH | 11 |
| 000BH | 12 |

**Output:**

|  |  |
| --- | --- |
| Register / Memory Location | Value |
| A | 09 |
| BC | 0007 |
| DE | 0006 |
| HL | 0C0B |
| 0004H | 9 |
| 0007H | 9 |

**Arithmetic:**

MVI A, 04H ; Store the value 04H in accumulator

MVI B, 04H ; Store the value 04H in B register

ADD B ; Add B to A

ADI 04 ; Add 04 to A

MVI C, 05H ; Store the value 05H in C register

SUB C ; Subtract C from A

SUI 03 ; Subtract 03 from A

INR B ; Increment B by 1

DCR C ; Decrement C by 1

INX D ; Increment the register pair DE by 1

DCX B ; Decrement the register pair B by 1

HLT ; End of program

**Input:**

|  |  |
| --- | --- |
| Register | Value |
| A | 04 |
| B | 04H |
| C | 05H |

**Output:**

|  |  |
| --- | --- |
| Register | Value |
| A | 04H |
| B C | 05H, 03H |
| D E | 00H, 01H |

**Logical:**

MVI A, 08H ; Store the value of 08H in A

MVI B, 09H ; Store the value of 09H in B

CMP B ; Compare the value of A and B

CPI 07H ; Compare the value of A with 07H

MVI D, 0FH ; Store the value of 0FH in D

ANA D ; Logical AND

ANI 0FH ; Logical AND with 0FH

MVI C, 07H ; Store the value of 07H in C register

ORA C ; Logical OR

RLC ; Rotate Left

RRC ; Rotate Right

RAL ; Rotate Left through carry flag

RAR ; Rotate Right through carry flag

XRI 0FFH ; Exclusive OR with 0FFH

HLT ; End of program

**Input:**

|  |  |
| --- | --- |
| Register | Value |
| A | 08H |
| B | 09H |
| D | 0FH |
| C | 07H |

**Output:**

|  |  |
| --- | --- |
| Register | Value |
| A | F0H |
| B C | 09H, 07H |
| D | 0FH |

**PRACTICAL 2**

1. **AIM: Write a program to add the content of location 4000h and 4001h and store answer at 4002 h.**

MVI A, 02H ; Move 02h 8-bit data in Accumulator (A)

STA 4000H ; Store Content of Accumulator (02) at Memory Location 2050

MVI A, 03H ; Move 03h 8-bit data in Accumulator (A)

STA 4001H ; Store Content of Accumulator (03) at Memory Location 2051

LDA 4000H ; Load Data (03) from Memory Location 2050 into A

MOV B, A ; Move Content of Accumulator (03) into Register B

LDA 4001h ; Load Data (02) from Memory Location 2051 into A

ADD B ; Add Content of Register B (03) & Content of A (02)

STA 4002h ; Store Content of Accumulator (05) at Memory Location 2052

HLT ; End of Program

**Input:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 4000H | 02 |
| 4001H | 03 |

**Output:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 4000H | 02 |
| 4001H | 03 |
| 4002H | 05 |

1. **AIM: Write an 8085 assembly language program for exchanging two 8-bit numbers stored in memory locations 2050h and 2051h.**

MVI A, 02H ; Move 02h 8-bit data in Accumulator (A)

STA 2050H ; Store Content of Accumulator (02) at Memory Location 2050

MVI A, 03H ; Move 03h 8-bit data in Accumulator (A)

STA 2051H ; Store Content of Accumulator (03) at Memory Location 2051

LDA 2050H ; Load Data (03) from Memory Location 2050 into A

MOV B, A ; Move Content of Accumulator (03) into Register B

LDA 2051H ; Load Data (02) from Memory Location 2051 into A

STA 2050H ; Store Content of Accumulator (02) at Memory Location 2050

MOV A, B ; Move Content of Register B (03) into Accumulator

STA 2051H ; Store Content of Accumulator (03) at Memory Location 2051

HLT ; End of Program

**Input:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 2050H | 02 |
| 2051H | 03 |

**Output:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 2050H | 03 |
| 2051H | 02 |

**PRACTICAL: 3**

## AIM: Write the program to add, subtract two 8 bit and 16 bit nos. without using 16 bit instructions.

### Addition of 8-bit no

MVI A, 02H ; Move 02h 8-bit data in Accumulator (A)

STA 4000H ; Store Content of Accumulator (02) at Memory Location 2050

MVI A, 03H ; Move 03h 8-bit data in Accumulator (A)

STA 4001H ; Store Content of Accumulator (03) at Memory Location2051

LDA 4000H ; Load Data (03) from Memory Location 2050 into A

MOV B, A ; Move Content of Accumulator (03) into Register B

LDA 4001h ; Load Data (02) from Memory Location 2051 into A

ADD B ; Add Content of Register B (03) & Content of A (02)

STA 4002h ; Store Content of Accumulator (05) at Memory Location2052

HLT ; End of Program

**Input:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 4000H | 02 |
| 4001H | 03 |

**Output:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 4000H | 02 |
| 4001H | 03 |
| 4002H | 05 |

### Subtraction of 8-bit no:

MVI A, 03H ; Move 03H 8-bit data in Accumulator (A)

STA 2050H ; Store Content of Accumulator (02) at Memory Location 2050 MVI A, 02H ; Move 02H 8-bit data in Accumulator (A)

STA 2051H ; Store Content of Accumulator (03) at Memory Location 2051

LDA 2050H ; Load Data (03) from Memory Location 2050 into A

MOV B, A ; Move Content of Accumulator (03) into Register B

LDA 2051H ; Load Data (02) from Memory Location 2051 into A

SUB B ; Subtract Content of Register B (03) & Content of A (02).

STA 2052H ; Store Content of Accumulator (01) at Memory Location 2052

HLT ; End of Program

**Input:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 2050H | 03 |
| 2051H | 02 |

**Output:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 2052H | 01 |

### 

### Addition of 16-bit no without using 16-bit Instruction

MVI A, 02H ; Move 02H 8-bit data in Accumulator (A)

STA 2050H ; Store Content of Accumulator (02) at Memory MVI A, 03H ; Move 03H 8-bit data in Accumulator (A)

STA 2051H ; Store Content of Accumulator (03) at MemoryLocation 2051

MVI A, 04H ; Move 04H 8-bit data in Accumulator (A)

STA 2052H ; Store Content of Accumulator (04) at Memory Location 2052

MVI A, 05H ; Move 05H 8-bit data in Accumulator (A)

STA 2053H ; Store Content of Accumulator (05) at Memory Location 2053

LHLD 2050H ; Load Data (03) from Memory Location 2051 into Register H & Data (02) from Memory Location 2050 into Register L

XCHG ; Move Content of Register H (03) into Register D & Content of Register L (02) into Register E

LHLD 2052H ; Load Data (05) from Memory Location 2053 into Register H & Data (04) from Memory Location 2052 into Register L

MOV A, H ; Move Content of Register H (05) into Accumulator

ADD D ; Adds Content of Accumulator (05) with Content of Register D (03) and result (08) Stores in Accumulator

STA 2054H ; Store Content of Accumulator (08) at Memory Location 2054

MOV A, L ; Move Content of Register L (04) into Accumulator

ADC E ; Add Content of Accumulator (04) with Content of Register E (02) as well as Carry (1= if it’s generated) and result (06) Stores in Accumulator

STA 2055H ; Store Content of Accumulator (06) at Memory Location 2055

HLT ; End of Program

**Input:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 2050H | 02 |
| 2051H | 03 |
| 2052H | 04 |
| 2053H | 05 |

**Output:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 2054H | 08 |
| 2055H | 06 |

### Addition of 16-bit no without using 16-bit Instruction

MVI B, 04H ; Moves 04 H into register B

MVI C, 0FFH ; Moves FF H into register C

MVI D, 01H ; Moves 01 H into register D

MVI E, 0EEH ; Moves EE H into register E

MOV A, C ; Moves content from C Register to AC

SUB E ; Subtract content of AC content with E Register

MOV L, A ; Moves content from AC to L Register AC

MOV A, B ; Moves content from B Register to AC

SUB D ; Add content of AC with D Register with borrow

MOV H, A ; Moves content from AC to H L pair Register

HLT ; End of program

**Input:**

|  |  |
| --- | --- |
| Register | Value |
| B | 04 |
| C | FF |
| D | 05 |
| E | EE |

**Output:**

|  |  |
| --- | --- |
| Register | Value |
| H | 03 |
| L | 11 |

1. **AIM: Write an 8085 assembly language program to add two 16-bit numbers stored in memory.**

MVI A, 02h ; Move 02h 8-bit data in Accumulator (A)

STA 2050h ; Store Content of Accumulator (02) at Memory Location 2050

MVI A, 03h ; Move 03h 8-bit data in Accumulator (A)

STA 2051h ; Store Content of Accumulator (03) at Memory Location 2051

MVI A, 04h ; Move 04h 8-bit data in Accumulator (A)

STA 2052h ; Store Content of Accumulator (04) at Memory Location 2052

MVI A, 05h ; Move 05h 8-bit data in Accumulator (A)

STA 2053h ; Store Content of Accumulator (05) at Memory Location 2053

LHLD 2050h ; Load Data (03) from Memory Location 2051 into Register H & Data (02) from Memory Location 2050 into Register L

XCHG ; Move Content of Register H (03) into Register D & Content of Register L (02) into Register E

LHLD 2052h ; Load Data (05) from Memory Location 2053 into Register H & Data (04) from Memory Location 2052 into Register L

DAD D ; Add Content of Register H (05) with Content of Register D (03) as well as Content of Register L (04) with Content of Register E (02) (including carry if it’s generated) and result (08) into Register H and result (06) into Register L

SHLD 2054h ; Store Content of Register H (08) at Memory Location 2054 and Register L (06) at Memory Location 2055

HLT ; End of Program

**Input:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 2050H | 02 |
| 2051H | 03 |
| 2052H | 04 |
| 2053H | 05 |

**Output:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 2054H | 08 |
| 2055H | 06 |

1. **AIM: Write a program to find 2’s Compliment of given no.**

LDA 8000H ; Load the number form memory

CMA ; Complement the AC

STA 8001H ; Store the 1’s Compliment

INR A ; Increased A by 1

STA 8002H ; Store 2’s Compliment

HLT ; End of Program

**Input:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 8000H | 01 |

**Output:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 8001H | 10 |
| 8002H | 11 |

**Practical 4**

1. **AIM: Write an 8085 assembly language program to add two decimal numbers using DAA instruction.**

MVI A, 0AH ; Move 0Ah 8-bit data (Hexadecimal) in Accumulator (A)

DAA ; Convert Content of Accumulator (0A- Hexadecimal) into 10 (Decimal), Store results (10) in AC

MOV B, A ; Move Content of Accumulator (10) into Register B

MVI A, 0BH ; Move 0Bh 8-bit data (Hexadecimal) in Accumulator (A)

DAA ; Convert Content of Accumulator (0B- Hexadecimal) into 11 (Decimal), Store results (11) in Accumulator

ADD B ; Add Content of Register B (10) with Content of Accumulator (10), Result (21) Store in AC

HLT ; End of Program

**Input:**

|  |  |
| --- | --- |
| Register | Value |
| A | 0AH |
| B | 0BH |

**Output:**

|  |  |
| --- | --- |
| Register | Value |
| A | 21H |

1. **AIM: Write an 8085 assembly language program to add two 16-bit numbers using DAD instruction.**

MVI A, 02H ; Move 02H 8-bit data in Accumulator (A)

STA 2050H ; Store Content of Accumulator (02) at Memory Location 2050

MVI A, 03H ; Move 03H 8-bit data in Accumulator (A)

STA 2051H ; Store Content of Accumulator (03) at Memory Location 2051

MVI A, 04H ; Move 04H 8-bit data in Accumulator (A)

STA 2052H ; Store Content of Accumulator (04) at Memory Location 2052

MVI A, 05H ; Move 05H 8-bit data in Accumulator (A)

STA 2053H ; Store Content of Accumulator (05) at Memory Location 2053

LHLD 2050H ; Load Data (03) from Memory Location 2051 into Register H & Data (02) from Memory Location 2050 into Register L

XCHG ; Move Content of Register H (03) into Register D & Content of Register L (02) into Register E

LHLD 2052H ; Load Data (05) from Memory Location 2053 into Register H &

Data (04) from Memory Location 2052 into Register L

DAD D ; Add Content of Register H (05) with Content of Register D (03) as well as Content of Register L (04) with Content of Register E (02) (including carry if it’s generated) and result (08) into Register H and result (06) into Register L

SHLD 2054H ; Store Content of Register H (08) at Memory Location 2054 and Register L (06) at Memory Location 2055

HLT ; End of Program

**Input:**

|  |  |
| --- | --- |
| Memory Location | Value(HEX) |
| 2050H | 02 |
| 2051H | 03 |
| 2052H | 04 |
| 2053H | 05 |

**Output:**

|  |  |
| --- | --- |
| Memory Location | Value(HEX) |
| 2054H | 08 |
| 2055H | 06 |

1. **AIM: Write a program to multiply two nos. (Hint: 2X3= 2+2+2)**

LHLD 2050 ; Loads content of 2051 in H and content of 2050 in L

XCHG ; Exchanges contents of H with D and contents of L with E

MOV C, D ; Copies content of D in C

MVI D, 00 ; Assigns 00 to D

LXI H, 0000 ; Assigns 00 to H and 00 to L

DAD D ; Adds HL and DE and assigns the result to HL

DCR C ; Decrement C by 1

JNZ 200A ; Jumps program counter to 200A if zero flag=0

SHLD 3050 ; Stores value of H at memory location 3051 and L at 3050

HLT ; End of program

**Input:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 2050H | 43 |
| 2051H | 07 |

**Output:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 3050H | D5 |
| 3051H | 01 |

1. **AIM: Write a program to mask (or clear) the lower 4-bit of the contents of the memory location 2050H.**

LDA 2050H ; Load the content of memory location 2050 in accumulator A

MOV B, A ; Move the content of A to B

ANI 0FH ; Perform AND operation of A with 0F and store the result back to A

STA 3050H ; Store content of A in memory location 3050

MOV A, B ; Moves the content of B in A

ANI 0F0H ; Perform AND operation of A with 0F and store the result back to A

RLC ; Rotate content of A left by 1 bit without carry. Use this instruction 4 times to reverse the content of A

RLC

RLC

RLC

STA 3051H ; Store the content of A in memory location 3051

HLT ; End of program

**Input:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 2050H | 14 |

**Output:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 3050H | 04 |
| 3051H | 01 |

1. **AIM: Write a program to set the higher 4 bits of the contents of memory location 2050H to 1.**

LDA 2050H ; Loads the content from location 2050H into AC

ANI 1FH ; Ands accumulator content with 1FH

STA 2050H ; Stores the accumulator content at 2050H

HLT ; End of program

**Input:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 2050H | 219 |

**Output:**

|  |  |
| --- | --- |
| Memory Location | Value |
| 2050H | 27 |

**PRACTICAL: 5**

1. **AIM: Write an 8085 assembly language program to find the minimum from two 8-bit numbers and store minimum no at 2000h location.**

MVI A, 1EH ; Move the value 1E into A

CPI 19H ; Compare direct with 19H

JC T ; Jump to label T if carry

MVI A, 19H ; Move the value 19H into A

T: STA 02H ; Store the value of A at 02H

HLT ; End of Program

|  |  |  |
| --- | --- | --- |
| **Input:** |  |  |

|  |  |
| --- | --- |
| Register | Value |
| A | 1EH |

**Output:**

|  |  |
| --- | --- |
| Register | Value |
| A | 19H |

1. **AIM: Write a program to transfer the block of data stored at 2000h location to 4000h location. Transfer any 5 nos.**

LXI H, 2000H ; Store the source address in H

LXI D, 4000H ; Store the destination address in D

MVI B, 05H ; Set the counter value to 5

NEXT: MOV A, M ; Move the value from memory to A STAX D ; Store the value at destination

INX H ; Increment H

INX D ; Increment D

DCR B ; Decrement B

JNZ NEXT ; Jump to NEXT if not zero

HLT ; End of program

**Input:**

|  |  |
| --- | --- |
| Register/  Memory address | Value |
| H | 2000H |
| D | 4000H |
| B | 05H |
| 2000H | 1 |
| 2001H | 2 |
| 2002H | 3 |
| 2003H | 4 |
| 2004H | 5 |

**Output:**

|  |  |
| --- | --- |
| Register/  Memory address | Value |
| H | 0007H |
| D | 0013H |
| B | 00H |
| 4000H | 1 |
| 4001H | 2 |
| 4002H | 3 |
| 4003H | 4 |
| 4004H | 5 |

1. **AIM: Write a program to transfer the block of data stored at 2000h location to 4000h location in the reverse order. Transfer any 5 nos.**

LXI H, 2000H ; Load starting address in H

LXI D, 4004H ; Load destination address in D

MVI B, 05H ; Initialize the value of counter

NEXT: MOV A, M ; Move the value from Memory to A

STAX D ; Store at the destination

INX H ; Decrement H by 1

DCX D ; Increment D by 1

DCR B ; Decrement B by 1

JNZ NEXT ; Jump to label NEXT if not zero

HLT ; End of program

**Input:**

|  |  |
| --- | --- |
| Register/  Memory address | Value |
| H | 2000H |
| D | 4000H |
| B | 05H |
| 2000H | 1 |
| 2001H | 2 |
| 2002H | 3 |
| 2003H | 4 |
| 2004H | 5 |

**Output:**

|  |  |
| --- | --- |
| Register/  Memory address | Value |
| H | 1999H |
| D | 4005H |
| B | 00H |
| 4000H | 5 |
| 4001H | 4 |
| 4002H | 3 |
| 4003H | 2 |
| 4004H | 1 |

1. **AIM: Write a program to add all the nos given in array and store the answer at 5000h location.**

XRA A ; Clean A to save sum

MOV B, A ; Clean B to save carry

MVI C, 05H ; Counter

LXI H, 0005H ; Load H with 0005H

NXTBYT: ADD M ; Add

JNC NXTMEM ; If no carry do not increment carry register

INR B ; Increment B by 1

NXTMEM: INX H ; Increment H by 1

DCR C ; Decrement C by 1

JNZ NXTBYT ; Jump if not zero

STA 0002H ; Store in 0002H

MOV A, B ; Move value from B to A

STA 0001H ; Store in 0001H

HLT ; End

LXI H, 000AH ; Store 000AH in H register

MOV A, M ; Store the value from A in memory

INX H ; Increment H by 1

MOV A, B ; Store the value from B in memory

HLT ; End of Program

|  |  |  |
| --- | --- | --- |
| **Input:** |  |  |

|  |  |
| --- | --- |
| Memory address | Value |
| 0005H | 1 |
| 0006H | 1 |
| 0007H | 1 |
| 0008H | 1 |
| 0009H | 1 |

**Output:**

|  |  |
| --- | --- |
| Memory address | Value |
| 0002H | 5 |

**PRACTICAL: 6**

1. **AIM: Write an 8085 assembly language program to get the minimum from block of N 8-bit numbers.**

LXI H, 2000H ; Load the value 0002H in H register

MOV C, M ; Store the value of memory pointed by HL at C

INX H ; Increment H by 1

DCR C ; Decrement C by 1

MOV A, M ; Store the value of memory pointed by HL at A

NEXT: INX H ; Increment H by 1

CMP M ; Compare memory with AC

JC SKIP ; Jump to SKIP if carry

MOV A, M ; Store the contents of memory in A

SKIP: DCR C ; Decrement the value of C

JNZ NEXT ; Jump to NEXT if not zero

INX H ; Increment H by 1

MOV M, A ; Store the contents of A in the value pointed by the HL register

HLT ; End of program

**Input:**

|  |  |
| --- | --- |
| Memory address | Value |
| 0002H | 5 |
| 0003H | 11 |
| 0004H | 6 |
| 0005H | 40 |
| 0006H | 32 |
| 0007H | 19 |

**Output:**

|  |  |
| --- | --- |
| Memory address | Value |
| 0008H | 5 |

1. **AIM: Write a program to find highest readings from array and display the reading at an output port.**

LXI H, 0002H ; Load the value of 0002H in H register

MOV C, M ; Store the value of memory in C

MVI A, 00H ; Clear A

NEXT: INX H ; Increment H by 1

CMP M ; Compare A with M

JNC SKIP ; Jump to SKIP if not carry

MOV A, M ; Move the content of memory into A

SKIP: DCR C ; Decrement C

JNZ NEXT ; Jump to NEXT if not zero

INX H ; Increment H by 1

OUT 01H ; Show output at port 01H

HLT ; End of program

**Input:**

|  |  |
| --- | --- |
| Memory address | Value |
| 0002H | 5 |
| 0003H | 66 |
| 0004H | 2 |
| 0005H | 34 |
| 0006H | 14 |
| 0007H | 53 |

**Output:**

|  |  |
| --- | --- |
| Port | Value |
| 01 | 66 |

1. **AIM: A string of readings is stored at memory location starting from 3000h and the end of strings is indicated by 0Dh. Write a program to check each byte in the string and save the bytes between the range of 30H to 39H both inclusive in memory location starting from 5000H.**

LXI H, 3000H ; Set the source pointer

LXI D, 5000H ; Set the destination pointer

LOOP: MOV A, M ; Move the value from memory to A

CPI 0DH ; Compare if 0DH or not

JZ END ; Jump to END if zero

CPI 30H ; Compare if less than 30H

JC REJECT ; Jump to REJECT if zero

CPI 3AH ; Compare if greater than 3AH

JNC REJECT ; Jump to REJECT if not zero

STAX D ; Store at the destination

INX D ; Increment D

REJECT: INX H ; Increment H by 1

JMP LOOP ; Jump to LOOP

END: HLT ; End of Program

|  |  |  |
| --- | --- | --- |
| **Input:** |  |  |

|  |  |
| --- | --- |
| Memory address | Value |
| 3000H | 35 |
| 3001H | 49 |
| 3002H | 30 |
| 3003H | 57 |
| 3004H | 50 |
| 3005H | 13 |

**Output:**

|  |  |
| --- | --- |
| Memory Address | Value |
| 5000H | 49 |
| 5001H | 57 |
| 5002H | 50 |

1. **AIM: Write a program to count no of zero values in the given block of data.**

LXI H, 0002H ; Set source pointer

MOV C, M ; Move the content from memory to C

MVI B, 00H ; Store the value 00H in B

NEXT: INX H ; Increment H by 1

MOV A, M ; Store the value from memory to A

CPI 00H ; Compare with 00H

JNZ SKIP ; Jump to SKIP if not zero

INR B ; Increment B by 1

SKIP: DCR C ; Decrement C by 1

JNZ NEXT ; Jump to NEXT if not zero

INX H ; Increment H by 1

MOV M, B ; Store the value of B in memory

HLT ; End of program

**Input:**

|  |  |
| --- | --- |
| Memory address | Value |
| 0002H | 5 (Size) |
| 0003H | 1 |
| 0004H | 0 |
| 0005H | 3 |
| 0006H | 0 |
| 0007H | 5 |

**Output:**

|  |  |
| --- | --- |
| Memory Address | Value |
| A | 5 |
| B | 2 |
| 0008H | 2 |