Some Solutions of 8085

1. Write a program for 8085 to swap bit D3 and D6 of ten numbers stored in memory at 9650H if any number is greater than 70H and less than A0H. Otherwise set D3 and reset D6 of the number stored. [2063 Ashad]

Label	Instructions	Comments
	LXI H,9650H	;source table
	MVI C,0AH	;counter
UP:	MOV A,M	
	CPI 71H	;check of number is greater or equal to 71H
	JC PASS	;if no goto PASS
	CPI A0H	;check of number is less than A0H
	JNC PASS	;if no goto PASS
	ANI 48H	;mask all bits except D3 and D6
	CPI 00H	;check if result is zero
	JZ PASS1	;if yes goto PASS1(no need to swap, both bits are zero)
	CPI 48H	;check if result is 48H
	JZ PASS1	;if yes goto PASS1(no need to swap, both bits are one)
	MOV A,M	
	XRI 48H	;toggle bit D3 and D6
	MOV M,A	
	JMP PASS1	
PASS:	ORI 08H	;set D3 bit
	ANI BFH	;reset D6 bit
	MOV M,A	
PASS1:	INX H	
	DCR C	
	JNZ UP	
	HLT	

2. Seven status and one control signal of a single microprocessor based instrument are read from data bus and are stored sequentially from memory location 6000H. Control bit is represented by bit D4 and D4=1 represents valid data. Other bits are status signals. Write an assembly language program for 8085 microprocessor which will check control bit of each data and transfer the valid data to new memory location starting from 7000H. The program should count and display the number of valid data and the checking process should stop when all status signals are zero.

Label	Instructions	Comments
	LXI H,6000H	;starting location where the data are stored
	MVI C,00H	; counter for valid data
	LXI D,7000H	;destination table
UP:	MOV A,M	
	ANI 0EFH	;to check the status bits
	CPI 00H	;check if all status bits are zero
	JZ DOWN	;if yes goto DOWN
	MOV A,M	
	ANI 10H	;Mask all bits except D4
	CPI 10H	;check if bit D4=1
	JNZ PASS	;if no goto PASS
	MOV A,M	
	STAX D	;store valid data to new table
	INR C	
	INX D	
PASS:	INX H	

```
        JMP UP
DOWN:   MOV A,C
        OUT PORTA       ;display the no. of valid data through PORTA
        HLT
```

3. Write a program for 8085 to transfer data from a table to another if the number of ones in the data is greater than four else store 00 to next table. [2065 Kartik] (Assuming 10 data)

```
Label       Instructions        Comments
            LXI H,D000H         ;source table
            LXI D,D050H         ;destination table
REPEAT:     MVI B,00H           ;register to count no. of ones
            MVI C,08H           ;counter to rotate number 8 times
            MOV A,M
UP:         RLC
            JNC PASS
            INR B
PASS:       DCR C
            JNZ UP
            MOV A,B
            CPI 05H             ;check if no. of ones is greater or equal to 5
            JNC PASS1           ;if yes goto PASS1
            MVI A,00H
            JMP PASS2
PASS1:      MOV A,M
PASS2:      STAX D
            INX H
            INX D
            MOV A,L
            CPI 0AH
            JNZ REPEAT
            HLT
```

4. Write a program in 8085 to add all the numbers from a table of 8-bit numbers whose higher nibble value is greater than 6 and store the 16-bit result just after the table. [2067 Shrawan]

(Assuming 10 data)

```
Label       Instructions        Comments
            LXI H,3000H         ;source table(assuming starting address of table is 3000H)
            MVI C,0AH           ;counter
            MVI D,00H           ;register to store sum
            MVI E,00H           ;register to store carry
UP:         MOV A,M
            ANI F0H             ;masking lower nibble
            CPI 70H             ;check if higher nibble is greater or equal to 7
            JC PASS             ;if no goto PASS
            MOV A,M
            ADD D
            MOV D,A
            JNC PASS
            INR E
PASS:       INX H
```

```
        DCR C
        JNZ UP
        MOV M,D         ;store sum in memory
        INX H
        MOV M,E         ;store carry in memory
        HLT
```

5. There are two tables holding twenty data whose starting address is 3000H and 3020H respectively. WAP to add the content of first table with the content of second table having same array index. Store sum and carry into the third and fourth table indexing from 3040H and 3060H respectively.

```
Label   Instructions      Comments
        LXI H,3000H     ;first source table
        LXI B,3020H     ;second source table
UP:     LDAX B
        ADD M
        MOV E,A         ;store sum to register E
        JC DOWN         ;if carry is generated goto DOWN
        MVI D,00H       ;store 00H to register D as carry
        JMP PASS
DOWN:   MVI D,01H       ;store 01H to register D as carry
PASS:   MOV A,L
        ADI 40H         ;adding 40H to goto memory location where sum is to be stored
        MOV L,A
        MOV M,E
        ADI 20H         ;adding 20H to goto memory location where carry is to be stored
        MOV L,A
        MOV M,D
        SUI 60H         ;subtracting 60H to return back to first source table
        MOV L,A
        INX H
        INX B
        MOV A,C
        CPI 34H
        JNZ UP
        HLT
```

6. WAP in 8085 to calculate the sum of numbers stored in memory location from 7000H to 700FH only if the number has higher nibble greater than lower nibble. Store the sum at end of the table.

```
Label   Instructions      Comments
        LXI H, 7000H    ;source table
        MVI B, 00H      ;sum register
        MVI C, 00H      ;carry register
        MVI D, 10H      ;counter
UP:     MOV A,M
        ANI 0FH
        MOV E,A
        MOV A,M
        ANI F0H
        RLC
        RLC
        RLC
```

```
        RLC
        CMP E
        JC PASS
        MOV A,M
        ADD B
        MOV B,A
        JNC PASS
        INR C
PASS:   INX H
        DCR D
        JNZ UP
        MOV M,B
        INX H
        MOV M,C
        HLT
```

7. WAP for 8085 to add corresponding data from two tables if the data from the first table is smaller than the second table else subtract data of second table from the first table. Store the result of each operation in the corresponding location of the third table. Assume each has ten 8-bit data.

Label	Instructions	Comments
	LXI B, 2000H	;first source table
	LXI H, 3000H	;second source table
	LXI D, 4000H	;destination table
UP:	LDAX B	
	CMP M	
	JNC SUBTRACT	
	ADD M	
	JMP PASS	
SUBTRACT:	SUB M	
PASS:	STAX D	
	INX H	
	INX B	
	INX D	
	MOV A, L	
	CPI OAH	
	JNZ UP	
	HLT	

8. Two tables with starting location 3000H and 4000H contains 50 bytes of data. WAP in 8085 to find sum of data from the tables and store the result in third table starting from 5000H if the result is in between C0H and FFH, else store 00H to the corresponding location.

Label	Instructions	Comments
	LXI B, 3000H	;first source table
	LXI H, 4000H	;second source table
	LXI D, 5000H	;destination table
UP:	LDAX B	
	ADD M	
	CPI C0H	
	JC PASS	
	CPI FFH	
	JNC PASS	
	JMP DOWN	
PASS:	MVI A,00H	
DOWN:	STAX D	
	INX H	

```
        INX B
        INX D
        MOV A,L
        CPI 32H          ;for 50 bytes(32 in hex) of data
        JNZ UP
        HLT
```

9. Ten 8-bit data are stored in two tables starting at 5050H and 5060H. Transfer the data of 1st table to third table starting at 5070H if data at 1st table is greater than data at 2nd table, else store 00H at third table.

```
Label   Instructions      Comments
        LXI B, 5050H      ;first source table
        LXI H, 5060H      ;second source table
        LXI D, 5000H      ;destination table
UP:     LDAX B
        CMP M
        JNC PASS
        MVI A,00H
PASS:   STAX D
        INX B
        INX H
        INX D
        MOV A,C
        CPI 5AH
        JNZ UP
        HLT
```

10. A table contains ten 8-bit data starting at 8050H. Write an 8085 program to store the sum of odd numbers at 8060H and sum of even numbers at 8070H. Also display the sum of even numbers and odd numbers at two different ports

```
Label   Instructions      Comments
        LXI H, 8050H      ;source table
        MVI B, 00H        ;sum register for even number
        MVI C, 00H        ;sum register for odd number
        MVI D, 0AH        ;Counter
UP:     MOV A, M
        RRC               ;Rotate right to send bit D0 to carry flag
        JC PASS           ;If D0( CY flag) is 1, number is odd
        MOV A, M
        ADD B
        MOV B, A
        JMP DOWN
PASS:   MOV A, M
        ADD C
        MOV C, A
DOWN:   INX H
        DCR D
        JNZ UP
        MOV A, B
        OUT PORT1
        MOV A, C
        OUT PORT2
        HLT
```

Tutorial – 1 (ALP 8085)
Microprocessor (BCT II / II)

1. Add two numbers located at 3030H and 4040H. Display sum on Port 1. If carry is generated, display it on Port 2. Store sum on 5050H.

```
        LDA 3030H
        MOV B, A
        LDA 4040H
        ADD B
        STA 5050H
        OUT PORT 1
        JNC L1
        MVI A, 01H
        OUT PORT 2
L1:     HLT
```

2. Write an Assembly Language Program that retrieves a data located at 2050H and it displays, if it is even and stores FFH on that location if it is odd.

```
        LDA 2050H
        ANI 01H
        JNZ L1
        LDA 2050H
        OUT PORT 1
        HLT
L1:     MVI A, FFH
        STA 2050H
        HLT
```

3. Sixteen bytes of data are stored in memory location at 1050H to 105FH. Replace each data byte by FF.

```
        LXI H, 1050H
        MVI C, 10H
L1:     MVI M, FFH
        INX H
        DCR C
        JNZ L1
        HLT
```

4. Sixteen data are stored in memory location at 1050H to 105FH. Transfer the entire block of data to new location starting at 1070H.

```
        LXI H, 1050H
        MVI C, 10H
        LXI D, 1070H
L1:     MOV A, M
        STAX D
        INX H
        INX D
        DCR C
        JNZ L1
        HLT
```

5. Six bytes are stored in memory locations starting at 2050H. Add all the data bytes, save any carry generated while adding the data bytes. Display entire sum at two output ports and store total carry in 2070H and sum in 2071H.

```
        LXI H, 2050H
        MVI C, 06H
        MVI B, 00H
        MVI D, 00H
L2:     MOV A, M
        ADD B
        MOV B, A
        JNC L1
        INR D
L1:     INX H
        DCR C
        JNZ L2
        HLT
```

6. If the content of memory location 2050H is greater than or equal to 64H, display 0FH else display FFH.

```
        LDA 2050H
        CPI 64H
        JC L1
        MOV A, 0FH
        OUT PORT 1
        HLT
L1:     MOV A, FFH
        OUT PORT 1
        HLT
```

7. We have a list of data stored at memory location starting at 2050H. The end of the data array is indicated by data byte 00H. Add the set of readings. Display the sum at Port 1 and total carry at Port 2.

```
        LXI H, 2050H
        MVI B, 00H
        MVI C, 00H
L3:     MOV A, M
        CPI 00H
        JZ L1
        ADD C
        JNZ L2
        INR B
L2:     MOV C, A
        INX H
        JMP L3
L1:     MOV A, C
        OUT PORT 1
        MOV A, B
        OUT PORT 2
        HLT
```

8. There are two tables holding twenty data whose starting address is 3000H and 3020H respectively. WAP to add the content of first table with the content of second table having same array index. Store sum and carry into the third and fourth table indexing from 3040H and 3060H respectively.

```
        LXI B, 3000H
        LXI H, 3020H
        LXI D, 3040H
NEXT:   LDAX B
        ADD M
        STAX D
        PUSH H
        PUSH D
        JNC L1
        MVI E, 01H
        JMP CSTORE
L1:     MVI E, 00H
CSTORE: LXI H, 3060H
        MOV A, L
        ADD C
        MOV L, A
        MOV M, E
        POP H
        POP D
        INX B
        INX D
        INX H
        MOV A, C
        CPI 14H
        JNZ NEXT
        HLT
```

9. For ten bytes data starting from 1120H, write a program to sort the reading in ascending and in descending order. (Note : For descending, do self)

```
START:  LXI H, 1120H
        MVI D, 00H
        MVI C, 0AH
   L2:  MOV A, M
        INX H
        CMP M
        JC L1
        MOV B, M
        MOV M, A
        DCX H
        MOV M, B
        INX H
        MVI D, 01H
   L1:  DCR C
        JNZ L2
        MOV A, D
        RRC
        JC START
        HLT
```

10. A set of ten readings is stored in memory location starting at 1160H. The readings are expected to be positive (<127). WAP to
    - Check each reading to determine whether it is positive or negative.
    - Reject all negative readings.
    - Add all positive readings & display sum in Port 1 and carry in Port 2.

```
        MVI B, 00H
        MVI C, 00H
        MVI D, 0AH
        LXI H, 1160H
   L2:  MOV A, M
        RAL
        JC NEGLECT
        RAR
        ADD B
        JC L1
        MOV B, A
   L1:  INR D
NEGLECT: INX H
        DCR D
        JNZ L2
        MOV A, B
        OUT PORT 1
        MOV A, D
        OUT PORT 2
        HLT
```

11. A set of six data bytes is stored starting from memory location 2050H. The set includes some blank spaces (bytes with zero values). WAP to eliminate the blanks from the block.

```
        MVI C, 06H
        LXI H, 2050H
        LXI B, 2050H
L2:     MOV A, M
        CPI 00H
        JZ L1
        STAX B
        INX B
L1:     INX H
        DCR C
        JNZ L2
        HLT
```

12. A set of eight data bytes (4 Pairs) are stored in memory locations starting from 1040H. WAP to add two bytes at a time and store the sum in same memory location, sum replacing the first byte and the carry replacing the second byte. If any pair does not generate a carry, the memory location of the second byte should be cleared i.e. store 00H over there.

```
        MVI C, 04H
        LXI H, 1040H
L2:     MOV A, M
        INX H
        ADD M
        DCX H
        MOV M, A
        INX H
        MVI M, 00H
        JNC L1
        MVI M, 01H
L1:     INX H
        DCR C
        JNZ L2
        HLT
```

13. WAP to read BCD number stored at memory location 2020H and converts it into binary equivalent and finally stores that binary pattern into memory location 2030H.
[Note: BCD number is the combination from 0 to 9]

```
        MVI C, 0AH
        LXI H, 2020H
        MOV A, M
        ANI F0H
        RRC
        RRC
        RRC
        RRC
        MOV B, A
        MOV A, 00H
L1:     ADD B
        DCR C
        JNZ L1
        MOV D, A
        MOV A, M
        ANI 0FH
        ADD D
        STA 2030H
        HLT
```

14. A binary number (Suppose FF: 1111 1111_2) is stored in memory location 2020H. Convert the number into BCD and store each BCD as two unpacked BCD digits in memory location from 2030H.

```
        LXI SP, 2000H
        LXI H, 2020H
        MOV A, M
        CALL PWRTEN
        HLT
PWETEN: LXI H, 2030H
        MVI B, 64H
        CALL BINBCD
        MOV M, D
        INX H
        MVI B, 0AH
        CALL BINBCD
        MOV M, D
        INX H
        MOV M, A
        RET
BINBCD: MVI D, 00H
NEXT:   INR D
        SUB B
        JNC NEXT
        DCR D
        ADD B
        RET
```

15. An 8 bit binary number is stored in memory location 1120H. WAP to store ASCII codes of these binary digits (0 to F) in location 1160H and 1161H.

```
LXI SP, 2000H          CODE:  CPI 0AH
LXI H, 1120H                  JC L1
LXI D, 1160H                  ADD 07H
MOV A, M                 L1:  ADD 30H
ANI F0H                       RET
RRC
RRC
RRC
RRC
CALL CODE
STAX D
INX D
MOV A, M
ANI 0FH
CALL CODE
STAX D
HLT
```

16. WAP to convert ASCII at location 1040H to binary and store at location 1050H.

```
LXI SP, 2000H          CODE:  CPI 40H
LXI H, 1040H                  JC L1
LXI D, 1050H                  SUB 07H
MOV A, M                 L1:  SUB 30H
ANI F0H                       RET
RRC
RRC
RRC
RRC
CALL CODE
STAX D
INX D
MOV A, M
ANI 0FH
CALL CODE
STAX D
HLT
```

17. A set of three packed BCD numbers are stored in memory locations starting at 1150H. The seven segment codes of digits 0 to 9 for a common cathode LED are stored in memory locations starting at 1170H and the output buffer memory is reserved at 1190H. WAP to unpack the BCD number and select an appropriate seven segment code for each digit. The codes should be stored in output buffer memory.

```
        LXI SP, 2999H
        LXI H, 1150H
        MVI D, 03H
        LXI B, 1190H
NEXT:   MOV A, M
        ANI F0H
        RRC
        RRC
        RRC
        RRC
        CALL CODE
        INX B
        MOV A, M
        ANI 0FH
        CALL CODE
        INX B
        INX H
        DCR D
        JNZ NEXT
        HLT

CODE:   PUSH H
        LXI H, 1170H
        ADD L
        MOV L, A
        MOV A, M
        STAX B
        POP H
        RET
```

18. A multiplicand is stored in memory location 1150H and a multiplier is stored in location 1151H. WAP to multiply these numbers and store result from 1160H.

```
        MVI B, 08H
        MVI D, 00H
        LXI H, 1150H
        MOV A, M
        MOV E, A
        LXI H, 1151H
        MOV A, M
L2:     RAR
        JNC L1
        LXI H, 0000H
        DAD D
L1:     XCHG
        DAD H
        XCHG
        DCR B
        LNZ L2
        HLT
```

19. A set of ten packed BCD numbers is stored in the memory location starting at 1150H. WAP to add these numbers in BCD. If carry is generated save it in register B and adjust it for BCD. The final sum is less than 9999_{BCD}.

```
        LXI SP, 2000H
        LXI H, 1150H
        MVI C, 0AH
        XRA A
        MOV B, A
L1:     CALL ADD
        INX H
        DCR C
        JNZ L1
        HLT

ADD:    ADD M
        DAA
        RNC
        MOV D, A
        MOV A, B
        ADI 01H
        DAA
        MOV B, A
        MOV A, D
        RET
```

20. A dividend is stored in memory location 2020H and a divisor is stored in 2021H. WAP to divide these numbers and store quotient and remainder from 2040H.

```
        MVI C, 00H
        LXI H, 2021H
        MOV A, M
        MOV D, A
        DCX H
        MOV B, M
L2:     MOV A, B
        SUB D
        JC L1
        MOV B, A
        INR C
        JMP L2
L1:     MOV L, C
        MOV H, B
        SHLD 2040H
        HLT
```

21. Write a program for 8085 to convert and copy the ten lower case ASCII codes to upper case from memory location 9050H to 90A0H if any, otherwise copy as they are. Assume there are fifty codes in the source memory. [Note: ASCII code for A=65 … Z=90, a=97 … z=122].
[2063 Kartik]

```
        LXI H, 9050H
        LXI D, 90A0H
        MVI C, 32H
L2:     MOV A, M
        CPI 60H
        JC L1
        SUI 20H
L1:     STAX D
        DCR C
        JNZ L2
        HLT
```

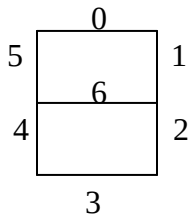
22. Write a program for 8085 to add ten 16-bit BCD numbers from location 4050H and store 24-bit BCD result at the end of the ten given numbers. [2062 Chaitra]

```
        LXI B, 4050H   ; Starting location of the 16-bit BCD Numbers
        LXI D, 0000H
        LXI H, 0000H
        MVI A, 00H

L2:     LDAX B
        ADD L
        INX B
        LDAX B
        ADC H
        JNC L1
        INR E
L1:     INX B
        MOV A, C
        CPI 0AH
        JC L2

        MOV A, L
        STAX B
        INX B
        MOV A, H
        STAX B
        INX B
        MOV A, E
        STAX B
        HLT
```

23. Write an 8085 program to display the BCD digits from 0 to 9 the seven segments as in the following diagram. Use the activating data bits same as the segment number as in figure below. [2059 Shrawan]




```
        LXI SP, 2999H
        LXI H, 2050H
        MOV M, 3FH
        INX H
        MOV M, 06H
        INX H
        MOV M, 5BH
        INX H
        MOV M, 4FH
        INX H
        MOV M, 66H
        INX H
        MOV M, 6DH
        INX H
        MOV M, 7DH
        INX H
        MOV M, 07H
        INX H
        MOV M, 7FH
        INX H
        MOV M, 6FH
        LXI B, 2060H

        LDAX B   ; Where the BCD digit is located
        ANI F0H
        RRC
        RRC
        RRC
        RRC
        CALL CODE
        OUT PORT 1
        LDAX B
        ANI 0FH
        CALL CODE
        OUT PORT 2
        HLT
CODE:   LXI H, 2050H
        ADD L
        MOV L, A
        MOV A, M
        RET
```

24. Write a program for 8085 to change the bit D_5 of ten numbers stored at address 7600H if the numbers are larger than or equal to 80H. [2061 Ashwin]

```
        LXI H, 7600H
        MVI C, 0AH
L2:     MOV A, M
        CPI 80H
        JC L1
        XRI 20H
        MOV M, A
L1:     INX H
        DCR C
        JNZ L2
```

25. Write a program for 8085 to find the smallest number among ten numbers stored at memory location 4500H. [2060 Bhadra]

```
        LXI H, 4500H
        MVI C, 0AH
        MOV A, M
L2:     INX H
        CMP M
        JC L1
        MOV B, A
        MOV A, M
        MOV M, B
L1:     DCR C
        JNZ L2
        OUT PORT 1
        HLT
```

26. Someone has damaged a program written at 4050H for 8085 microprocessor. The damaging is done by changing the bit D_7 and bit D_5 of each byte. The size of the program is 100 bytes. Now write a program for 8085 to correct this damaged program. [2060 Chaitra]

```
        LXI H, 4050H
        MVI C, 64H
L1:     MOV A, M
        ANI 80H   ; 10000000 B
        RRC
        RRC
        MOV B, A
        MOV A, M
        ANI 20H   ; 00100000 B
        RLC
        RLC
        MOV C, A
        MOV A, M
        ANI 5FH    ; 01011111 B
        ORA B
        ORA C
        STAX H
        INX H
        DCR C
        JNZ L1
        HLT
```

27. The temperature of two furnaces being monitored by a microprocessor based system. A set of readings of the first furnace recorded by thermal sensor is stored at memory locations starting at 4050H. Corresponding readings from the second furnace is stored at the memory location starting at 4070H. Each reading from the first furnace is expected to be higher than the corresponding reading from the second furnace. Among the eight bit data bit D_7 is used to test the validity of the data. Write an 8085 program to compare valid data from the two tables, if data from first table is larger than the corresponding data from the second table store 01H in the corresponding memory of the third location starting at 4090H and display 01H to indicate the normal operation else store FFH in the corresponding memory location and display FFH in the port to indicate the emergency. When emergency condition is reached stop the operation. [2060 Jestha]

```
        LXI B, 4050H
        LXI H, 4070H
        LXI D, 4090H
L2:     LDAX B
        CMP M
        JC L1
        JZ L1
        MVI A, 01H
        STAX D
        OUT PORT
        INX B
        IND H
        INX D
        JMP L2
L1:     MVI A, FFH
        STAX D
        OUT PORT
        HLT
```

28. Write a program to transfer eight-bit numbers from 9080H to 9090H if bit D_5 is 1 and D_3 is 0. Otherwise transfer data by changing bit D_2 and D_6 from 1 to 0 or from 0 to 1. Assume there are ten numbers. [2064 Shrawan]

```
        LXI H, 9080H
        LXI D, 9090H
        MVI C, 0AH
L2:     MOV A, M
        ANI 28H
        CPI 20H
        JZ L1
        MOV A, M
        XRI 44H
        MOV M, A
L1:     MOV A, M
        STAX D
        INX H
        INX D
        DCR C
        JNZ L2
        HLT
```

29. There are two tables T1, T2 in memory having ten eight bit data in each. Write a program for 8085 to find the difference of the corresponding element of these two tables. Store the result of each operation on the corresponding element of the third table. Remember that the result should not be negative ; it should be |T1 – T2|. [2064 Poush]

```
        LXI SP, 2999H
        LXI H, 5000H        ; TABLE T1
        LXI D, 6000H        ; TABLE T2
        MVI C, 0AH          ; COUNTER FOR 10 DATA
L1:     LDAX D
        MOV B, A
        MOV A, M
        CMP B
        JNC L2
        MOV A, B
        MOV B, M
L2:     SUB B
        PUSH D
        MVI D, 70H          ; TABLE T3
        STAX D
        POP D
        INX H
        INX D
        DCR C
        JNZ L1
        HLT
```

30. Write a program for 8085 to transfer data from a table to another if the number of ones in the data is greater than four else store 00 in the next table. [2065 Kartik]

```
        LXI H, 5000H        ; SOURCE TABLE
        LXI D, 6000H        ; DESTINATION TABLE
ST:     MVI C, 08H          ; NO OF BITS
        MVI B, 00H          ; NO OF 1'S
        MOV A, M
L1:     RLC
        JNC L2
        INR B
L2:     DCR C
        JNZ L1
        MOV A, B
        CPI 04H
        MVI A, 00H
        JC L3
        JZ L3
        MOV A, M
L3:     STAX D
        INX H
        INX D
        MOV A, E
        CPI 0AH             ; SUPPOSE TABLE FOR 10 DATA
        JNZ ST
        HLT
```

31. Write an assembly language program to count no. of –ve element in a data block containing 16 bytes of data; store the count at the end of the block if the count is greater than 8 otherwise stores 0. [2065 Chaitra]