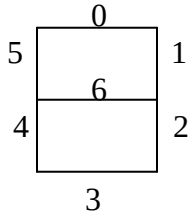
1. Add two numbers located at 3030H and 4040H. Display sum on Port 1. If carry is generated, display it on Port 2. Store sum on 5050H.
2. Write an Assembly Language Program that retrieves a data located at 2050H and it displays, if it is even and stores FFH on that location if it is odd.
3. Sixteen bytes of data are stored in memory location at 1050H to 105FH. Replace each data byte by FF.
4. Sixteen data are stored in memory location at 1050H to 105FH. Transfer the entire block of data to new location starting at 1070H.
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.
6. If the content of memory location 2050H is greater than or equal to 64H, display 0FH else display FFH.
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.
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.
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)
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.
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.
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.
13. WAP to read BCD number (Suppose 70_{10}: $0111\ 0000_{BCD}$) stored at memory location 2020H and converts it into binary equivalent and finally stores that binary pattern into memory location 2030H.
14. A binary number (Suppose FFH: $1111\ 1111_2$) 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.
15. An 8 bit binary number is stored in memory location 1120H. WAP to store the ASCII codes of the binary digits in location 1160H and 1161H.
16. WAP to convert ASCII code stored at memory location 1040H to binary equivalent and store the result at location 1050H.
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.
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.

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}.
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.
21. Design a counter to count continuously from 00H to FFH with 500ms delay between each count. Display the count at PORT1 and PORT2 (one digit per port) with a common-anode seven-segment –LED code.
22. Load the data byte 8EH in register D and F7H in register E. Mask the high order bits(D_7-D_4) from both the data bytes. Exclusive-OR the low-order bits (D_3-D_0) and display the answer at PORT1.
23. WAP to add two Hex numbers 7AH and 46H and store the sum at memory location 9601H and the flag status at location 9600H.
24. A set of 10 bytes are stored in memory starting with the address XX50H. Write a Program to check each byte, and save the bytes that are higher than 60_{10} and lower than 100_{10} in memory locations starting from XX60H. Data: 6FH,28H,5AH,49H,C7H,3FH,4BH,78H,64H
25. 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]



26. 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]
27. Write a program for 8085 to find the smallest number among ten numbers stored at memory location 4500H. [2060 Bhadra]
28. 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]
29. 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]
30. Registers BC contain 2793H and registers DE contain 3182H. Write instructions to add these 16-bit numbers and place the sum in memory locations 2050H and 2051H. [2062 Baishakh]
31. Write a program for 8085 to add ten 16-bit BCD numbers and store 24-bit BCD result at the end of the ten given numbers. [2062 Bhadra]
32. 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]
33. Write a program for 8085 to swap bit D_3 and D_6 of ten programs stored in memory at 9650H if any number is greater than 70H and less than A0H otherwise set D_3 and reset D_6 of the numbers stored. [2063 Ashad]

34. 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]
35. 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]
36. 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]
37. 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]
38. 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]
39. Write a program for 8085 to find the sum of the following series.
    X + (X+5D) + (X+10D) + (X+15D) + … + to ten terms [2066 Kartik]
40. Write a program 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 table has ten eight bit data. [2066 Magh]
41. Write a program in 8085 to add all the numbers from a table of 8-bit numbers whose higher nibble value is greater than 06H and store the 16-bit result just after the table. [2067 Shrawan]
42. A set of three reading is stored in memory starting at 9040H. Write an assembly language program to sort the readings in ascending order. Store the smallest value in address 9054H and so on in higher addresses. [2067 Mangsir]
43. There is a table in memory which has ten eight bit numbers starting at 9350H. Write a program for 8085 to transfer the numbers from this table to another table that starts at location 9540H by swapping bit D_6 and D_2 if the number is greater than 90H else transfer by adding 48H. [2068 Jestha]
44. Write a program to convert ten BCD numbers stored at 4350H to binary and store the result at 4360H. [2069 Bhadra]
45. Write a program in 8085 to transfer bytes of data with odd parity from location 9205H to A200H, else transfer the data by clearing bit D_5 and setting bit D_3. The end of data bytes is indicated by 51H in the data. [2069 Poush]
46. Write an assembly language program for 8085 to exchange the bits D6 and D2 of every byte of a program. Suppose there are 200 bytes in the program starting from memory location 8090H. [2070 Bhadra]
47. Write an assembly language program in 8085 to divide a byte stored in memory location 9070H by byte stored in 9071H and store the remainder and quotient at 9072H and 9073H respectively. [2070 Magh]