8085 program tutorial 1

1. Add two numbers stored at locations 8080H and 8090H and save the result at location 80A0H.
2. Write a program in 8085 to load 7DH in register B and 9FH in register C. Add these numbers, complement the result and store the final result in F0F0H.
3. Write a program in 8085 to subtract the value from memory 7070H with value from memory 7071H. Set bit D2 and reset bit D5 of the result and store the final result in memory 7095H.
4. Write a program in 8085 to set Auxiliary Carry flag and reset Zero flag without affecting other flags.
5. Write a program to set auxiliary flag and reset parity flag without affecting other flags.
6. Set bit D2 of register D and reset bit D5 of register E. Then store the final value of register D in memory 8050H and register E in memory 8060H.
7. 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.
8. 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.
9. If the content of memory location 2050H is greater than or equal to 64H, display 0FH else display FFH.
10. WAP for 8085 to swap higher nibble and lower nibble of a number stored at location E050H.
11. 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]
12. Sixteen bytes of data are stored in memory location at 1050H to 105FH. Replace each data byte by FF.
13. Sixteen data are stored in memory location at 1050H to 105FH. Transfer the entire block of data to new location starting at 1070H.
14. 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.
15. 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.
16. WAP for 8085 to transfer 10 8-bit data at location starting from C050H to the location starting from d050H.
17. WAP for 8085 to swap bit D1 and D5 of sixteen numbers stored in memory at 9650H if number is greater than 29H and less than D9H otherwise set D1 and reset D5 of the number stored.
18. Ten 8-bit data are stored from 2000H to 2009H. Add these numbers and store 16-bit result at end of the table. Also display the results at output ports.
19. WAP for 8085 to change the bit D5 of ten numbers stored at address 7600H if the numbers are greater than or equal to 80H.
20. WAP to find the sum of higher nibble and lower nibble of 10 8-bit numbers at 3030H and store the result at the table starting 4040H.

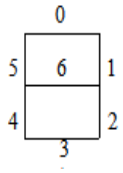
8085 program tutorial 2

21. Write a program in 8080 to transfer ten 8-bit numbers from one table to another if sum of higher nibble and lower nibble is less than 10H else store 00H in another table.
22. WAP for 8085 to find the smallest number among ten numbers stored at memory location 4500H.
23. 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
24. 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)
25. 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.
26. 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.
27. 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.
28. Transfer ten data, which has bit D5 and D0, 0 and 1 respectively from 6430H to 6440H, else store 00 instead of transformation.
29. Transfer ten data with even parity from location 5270H to 5280H else transfer the data by clearing bit D7 and setting bit D2.
30. Ten data are stored from 4080H. Transfer the first 5 numbers at the end of the second table and the rest at the starting of it.
31. Data are stored from 4050H to 405AH. Insert 5 data after 4055H taking from 4040H, but do not lose the previous content.
32. Transfer data from 5050H to 5060H only if data is between 30H and 70H else store 00H in the next table. (Assuming 10 data)
33. WAP for 8085 to swap bit D3 and D6 of sixteen numbers stored in memory at 9650H if number is greater than 70H and less than A0H otherwise set D3 and reset D6 of the number stored.
34. WAP to transfer 8-bit numbers from 9080H to 9090H of bit D5 is 1 and D3 is 0. Otherwise transfer data by changing bit D2 and D6 from 1 to 0 or from 0 to 1. Assume there are ten numbers.
35. 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. WAP 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.
36. WAP for 8085 to add ten 8-bit BCD numbers and store 16-bit result at the end of the table.
37. Ten data are stored in memory location starting at 8345H. Write a program to convert BCD number to binary number and store the result in the second table in the memory location starting at 8365H. Make subroutine for conversion.
38. Ten data are stored in memory location starting at 8345H. WAP to convert binary number to BCD number and store the result in the second table in the memory location starting at 8445H.
39. Write a program for 8085 to find the sum of the following series: x + (x + 5D) + (x + 10D) + (x + 15D) + ……. + to ten terms where x is an 8 bit number stored at location 8085. Store the sixteen bit result at location 8086. [2066 Kartik] (Assuming D is the value stored at register D)

8085 program tutorial 3

40. Ten data are stored in memory location starting at d000H. Write a program to convert single digit hexadecimal number to its equivalent ASCII code and store the result in the second table in the memory location starting at d050H. Make subroutine for conversion.
41. 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)
42. A set of five readings is stored in memory starting at 2050H. Write a program for 8085 to sort the reading in ascending order.
43. Write 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]
44. Registers BC contains 2793H and register DE contain 3182H. Write instructions to add these 16 bit numbers and place the sum in memory location 2050H and 2051H.[2062 Baisakh]
45. 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]
46. 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]
47. 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 store 0.[2065 Chaitra] (Assuming MSB=1 → -ve and MSB=0 → +ve)
48. 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]
49. There are two tables T1, T2 in memory having 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]
50. 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.
51. Write an 8085 program to add ten numbers stored in the consecutive memory locations starting from 4081H and display the 16-bit result in two output ports.
52. 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]
53. 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 has ten eight bit data. [2066 Magh]
54. Add all the numbers with bit D5 and D3, 1 and 0, stored in the memory location 50B1H to 50BAH. Display the 16-bit result in any ports.

55. Write an 8085 program to EX-OR two data stored at 2040H and 2041H without using EX-OR instruction and store the result at 2042H. (Using formula A XOR B=AB'+A'B)
56. 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]



57. A table contains ten 8-bit data starting at 8050H. Write an 8085 program to store the sum of add numbers at 8060H and sum of even numbers at 8070H.
58. Find the number of negative, positive and zero elements in the given series of data. The length is given in memory location 2030H and the series starts from next after this location. Store the results at the end of the table contiguously. (In signed magnitude form if MSB=1, no. is –ve else +ve, i.e. no. greater than 7fH is –ve)
59. 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.
60. The temperature of last 1 hour at an interval of 1 minute, are stored in the memory location starting from 2060H. Transfer all the normal temperatures (20-30 degree celcius) to another location starting from 3060H. Store 00H and FFH in the table for below and above the normal temperatures respectively as well as display the corresponding data in a port to indicate cold and hot.

8085 program tutorial 4

61. 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.
62. Generate the Fibonacci series and store them at memory location starting from 2050H up to 10 terms.
63. A set of three reading is stored in memory starting at 9040H. Write an assembly language program to sort the reading in ascending order. Store the smallest value in address 9054H and so on in higher addresses. [2067 Mangsir]
64. 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)
65. 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 D6 and bit D2 is the number is greater than 90H else transfer by adding 48H. [2068 Jestha]
66. Write a program in 8080 to transfer ten 8-bit numbers from one table to another if sum of higher nibble and lower nibble is less than 10H else store 00H in another table.
67. 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.
68. 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)
69. A set of ten readings is stored in memory location starting at 1160H. The readings are expected to be positive (<127). WAP to
    a. Check each reading to determine whether it is positive or negative.
    b. Reject all negative readings.
    c. Add all positive readings & display sum in Port 1 and carry in Port 2.

70. 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.
71. 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.
72. 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]
73. 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.
74. 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.
75. WAP to convert ASCII at location 1040H to binary and store at location 1050H.
76. 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.
77. 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
78. 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}.
79. 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)