Total Pages—15 **PG/IIIS/ELC-305/12(Pr.)**

M.Sc. 3rd Semester Examination, 2012

ELECTRONICS

(*Microprocessor Programming*)

(Practical)

PAPER — ELC-305

Full Marks : 50

Time : 3 hours

Answer any **one** question, selecting it by a lucky draw.

1. A set of ten current readings is stored in memory locations starting at XX50H. The readings are expected to be positive ($<127_{10}$). Write an assembly language program to :

(*a*) Check each reading to determine whether it is positive or negative.

(*b*) Reject all negative readings.

(*c*) Add all positive readings.

(*d*) Store FFH in the memory location XX60H when the sum exceeds eight bits ; otherwise, store the sum.

Data (H) : 28, D8, C2, 21, 24, 30, 2F, 19, F2, 9F.

2. A set of ten bytes are stored in memory starting with the address XX50H. Write an assembly language 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.
Sort the selected bytes in ascending order.

Data (H) : 6F, 28, 5A, 49, C7, 3F, 37, 4B, 78, 64.

3. Write an assembly language program to convert a 2-digit BCD number into its equivalent binary number. The BCD number is available from a memory location NUMB and store the converted binary number into location NUMB+1. Repeat the operation for three different data.

4. An 8-bit binary number is stored in a memory location XX20H. Write an assembly language program to convert the binary number into its equivalent Gray code and store the result in a memory location XX50H. Repeat the operation for 6 different binary numbers.

5. Write an assembly language program to convert one 8-bit Gray code stored in a memory location X200 H, into its binary equivalent number and store the result in a memory location X300 H. Repeat the process for 6 different Gray codes.

6. Write an assembly language program to calculate the square of a given number ($<16_{10}$) using the following algorithm:

Step I : Square ←0, Count ← given number Odd ← 1

Step II : Square ←square + Odd

StepIII : Count ← count – 1

Step IV : If count=0, then go to step VI

Step V : Odd ← Odd + 2

Step VI : Store the current value of square.

The number to be squared is available from a memory location NUMB and store square in a memory location RESULT. Repeat this operation for 5 numbers.

7. Write an assembly language program to find the 2's complement of an 8-bit number using the following algorithm :

"Copy all the bits starting from the LSB of the byte up to the first 1 bit of the number and then complement all the bits up to MSB." Repeat this experiment for 5.

8. Write an assembly language program to compute :

$$S = 2N_1 + 3N_2 + 7N_3$$

where N_1, N_2 and N_3 are three 8-bit numbers available from three consecutive memory locations. St the sum S in another memory location. Perform the experiment for five different sets of N_1, N_2 and N_3.

9. Write an assembly language program to calculate the square root of a given real number. Store the number and FFH if the number is a perfect square ; Otherwise an error message FEH, and the result of the perfect square number in the consecutive memory locations.

10. Write an assembly language program to find the HCF of three 8-bit numbers stored in three consecutive memory locations starting from XX50H. Store the result in a memory location XX60H. Repeat the process with five different sets of data.

11. Write an assembly language program to find the LCM of three 8-bit numbers stored in three consecutive memory locations starting from XX50 H. Store the result in a memory location XX60 H. Repeat the process with five different sets of data.

12. Write an assembly language program to multiply two 8-bit numbers using left-shift and add method. The numbers are stored in consecutive memory locations. Store the product in a suitable memory location. Also find $X = N_1 * N_2 - N_3$, where N_1, N_2 and N_3 are all 8-bit numbers. The number N_3 may be available from a suitable memory location. Store the value of X into the location just after a location where the result of product $N_1 * N_2$ is stored. Repeat the experiment using three sets of N_1, N_2 and N_3.

13. Write an assembly language program to transfer the bytes from a block of 16 bytes stored from location X050H to X05F following the condition stated below :

"The byte should be even but the upper nibble of the byte should be odd." The starting address of the destination should be X300H.

14. Write an assembly language program to transfer the bytes from a block of 16 bytes stored from X050H to X05F following the condition stated below :

"The transferred bytes should have the upper nibble double of the lower nibble" the starting address of the destination should be X300H. Repeat this process for 3 different block of bytes.

15. Write an assembly language program to transfer a block of data bytes stored from memory location X050H to X05F. The data are to be stored from location from X300H to X30FH in reverse order. Repeat this process for 3 different block of bytes.

16. Four bytes P, Q, R and S are stored from a memory location with address X050H onwards. The decimal value of each of the byte does not exceed 15. Write an assembly language program to compute the value of $P \times Q - R \times S$ and to store the result in the memory location. Use the repeated addition method for multiplication as subroutine. Repeat the process for two different combinations of P, Q, R and S.

17. An array 16 bytes are stored from a suitable memory location. Write an assembly language program to search for those bytes whose square root is perfect integer less than 5. Store those bytes from a separates memory location onwards. Repeat this process for at least 3 sets of the arrays.

18. An array of 5 bytes are stored from a suitable memory location. Find the hight common factor for the bytes using an assembly language programme. Use a subroutine for finding the highest common factor for 2 number. Repeat this process for at least 2 sets of arrays.

19. Interfacing two LEDs using common anode technique, write an assembly language program to blink them alternately. Assume 8255A is interfaced in I/O-mapped I/O.

20. Interface 8 switches and a seven segment display to 8085 μP through 8255A. Write an assembly language program to display the switch number when a switch is open. Assume that the 8255A is interfaced in memory-mapped I/O technique.

Distribution of Marks

Flow chart	:	05 Marks
Assembly language program	:	10 Marks
Execution of the program	:	15 Marks
Description of the program	:	05 Marks
Viva -voce	:	10 Marks
Laboratory Note Book	:	05 Marks
	Total :	50 Marks