Problem 1:

Write an assembly language program to find the minimum number in an array of ***signed numbers***? Create a variable of name “minimum”.

Let array of size byte:

Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Value	0xBA	0X12	0XEF	0X87	0XAA	0X77	0X7F	0XF3	0X3F	0X09

Problem 2:

Convert the following high level language program to its equivalent assembly language program. Use loop to read the contents of an array.

int arr[] = {32,-84,39,56,23,56,33,-1,17,30}; (decimal signed)

int sum = 0; (save the sum of array in variable sum)

Hint : **Declare byte type array and word type sum.**

sum=sum+ arr[0]+1;

sum= sum + arr[2] +2 ;

sum= sum + arr[4] +3 ;

sum= sum + arr[6] +4 ;

sum= sum + arr[8] +5 ;

NOTE: Assembly language do not support memory to memory mov,add,sub instructions.

Problem # 3:

Write an assembly language program that reverses the whole array of numbers. Use temporary variable to perform the desired operation. Do not declare two arrays declare only one array and can use temp variable for swapping purpose. Hardcoding is not allowed.

Let Array 1 (byte size):

Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Value	5A	6B	24	8D	35	0F	90	34	73	66

After reverse Array 1 (byte size) should be:

Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Value	66	73	34	90	0F	35	8D	24	6B	5A