Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Registration: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Question #1: [10 points]

You required to solve it on paper/assembler, and paste its image as solution in Word/pdf document.

Consider the following assembly program. What changes are made in the affected memory locations after execution of following code? Draw the relevant memory locations with values.

Value of DS = 0xA000

|  |  |
| --- | --- |
| **MOV AX, [0XBCD8]**  **MOV BX, [0XBCDA]**  **ADD AX, BX**  **MOV [0XBCD4], AX** |  |

Question#2: [10 points]

Consider an array of any size with random values of word type. You need to calculate the sum of it’s first & last elements and place it in AX register.

**Note that, if we change the array size, your program should work accordingly.**

Question#3: [10+10 points]

1. Consider three arrays of arbitrary size, say 10 elements. You need to add their adjacent entries and place their sum in third array adjacently.
2. In part (a), whenever a sum is calculated, you need to check the carry Flag status. Whenever Carry is generated, save the index value in 4th array called **IndexCarry** array.

Question#4: [10 points]

Consider the following program and implement it’s equivalent in assembly language.

Assume, Operand1=0; Operand2=10 initially,

While (Operand1 < Operand2)

{

Operand1++;

If (operand1==operand2)

X=2;

Else

X=3;

}