Lab B4

TT0L - GROUP 0

Person 1	1111111111
Person 2	1111111111

---

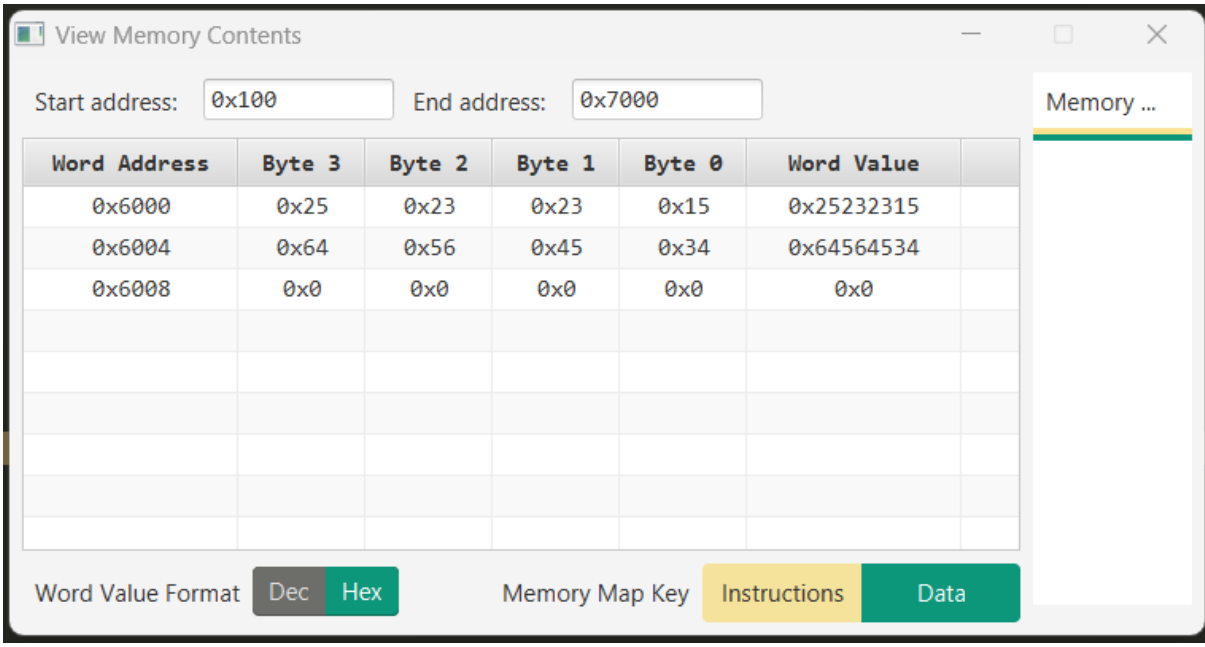
1. Perform the following operations by writing programs using ARM instruction.

   34,25,15,23,45,64,56,23

   a. Sort the numbers above in ascending order using bubble sort.

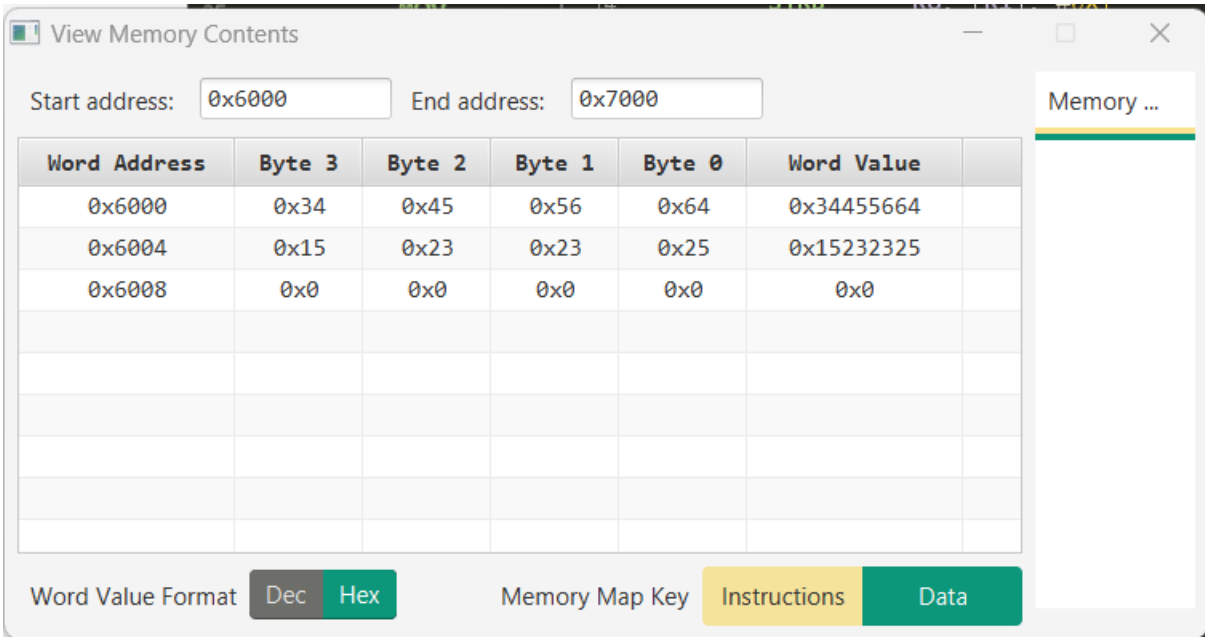
```
1           MOV     R1, #0X6000
2           MOV     R2, #0X34
3           MOV     R3, #0X25
4           MOV     R4, #0X15
5           MOV     R5, #0X23
6           MOV     R6, #0X45
7           MOV     R7, #0X64
8           MOV     R8, #0X56
9           MOV     R9, #0X23
10          STRB    R2, [R1], #0X1
11          STRB    R3, [R1], #0X1
12          STRB    R4, [R1], #0X1
13          STRB    R5, [R1], #0X1
14          STRB    R6, [R1], #0X1
15          STRB    R7, [R1], #0X1
16          STRB    R8, [R1], #0X1
17          STRB    R9, [R1], #0X1
18          MOV     R2, #0X6000
19          MOV     R3, #0
20          BL      SORT
21          END
```

```
22
23 SORT
24          MOV       R10, R2
25          MOV       R7,R3
26          MOV       R4, #01
27
28 NEXT
29          LDRB      R5,[R10],#1
30          LDRB      R6,[R10]
31          CMP       R5,R6
32          BLE       SKIP
33          STRB      R5,[R10],#-1
34          STRB      R6,[R10]
35          ADD       R10, R10,#1
36          MOV       R7, #1
37
38 SKIP
39          ADD       R4,R4,#1
40          CMP       R4, #8
41          BNE       NEXT
42          CMP       R7, #1
43          BEQ       SORT
44
45          MOV       PC, LR
```



b. Sort the numbers in descending order by changing BLE to BGE.

```
23 SORT
24              MOV         R10, R2
25              MOV         R7,R3
26              MOV         R4, #01
27
28 NEXT
29              LDRB        R5,[R10],#1
30              LDRB        R6,[R10]
31              CMP         R5,R6
32              BGE         SKIP
33              STRB        R5,[R10],#-1
34              STRB        R6,[R10]
35              ADD         R10, R10,#1
36              MOV         R7, #1
37
38 SKIP
39              ADD         R4,R4,#1
40              CMP         R4, #8
41              BNE         NEXT
42              CMP         R7, #1
43              BEQ         SORT
44
45              MOV         PC, LR
```



2. Assume there is a five-stages instruction pipeline - Fetch (F), Decode (D), Fetch Operand (FO), Execute (E) and Write (W) running in a microprocessor. Assume that each stage requires one-time unit and no branch instruction is involved

a. By using formula, how many time units are needed to complete these FOUR instructions with pipelining?

$$T_{k,n} = [k + (n - 1)]\tau$$
$$T_{k,n} = [5 + (4 - 1)]$$
$$= 8\ time\ units$$

1	2	3	4	5	6	7	8
F	D	FO	E	W			
	F	D	FO	E	W		
		F	D	FO	E	W	
			F	D	FO	E	W

b. By using formula, calculate the total time required to execute FOUR instructions without pipelining.

$$T_{1,n} = nk\tau$$
$$T_{1,n} = 4 \cdot 5$$
$$= 20\ time\ units$$

c. Calculate the speedup factor for the same number of instructions

$$\frac{nk}{k+(n-1)}$$
$$(4 * 5)/5 + (4 - 1) = 2.5$$

3. Write a program to evaluate the arithmetic expression A = [(B+C) - D)] / E, using one address instructions, two address instructions and three address instructions. The instructions available for use are as follows

One Address	Two Address	Three Address
LOAD X		
STORE X	MOVE X ,Y	
ADD X	ADD X, Y	ADD X,Y,Z
SUB X	SUB X, Y	SUB X,Y,Z
MUL X	MUL X ,Y	MUL X,Y, Z
DIV X	DIV X, Y	DIV X, Y, Z

One Address

LOAD B	ACC = B
ADD C	ACC = B + C
SUB D	ACC = (B + C) - D
DIV E	ACC = [(B + C) - D] / E

STORE A A = ACC

Two Address

MOV A, B A = B
ADD A,C A = B + C
SUB A,D A = (B + C) - D
DIV A,E A = [(B + C) - D] / E

Three Address

ADD A, B, C A = B + C
SUB A, A, D A = (B + C) - D
DIV A, A, E A = [(B + C) - D] / E

4. Suppose an 8-bit data word stored in memory is 1111 1000. Using the Hamming algorithm, determine what is the value of the four check bits (Check bit 8, Check bit 4, Check bit 2 and Check bit 1) that would be stored in memory with the data word. Show how you got your answer.

Hamming Code

	12	11	10	9	8	7	6	5	4	3	2	1
Bit	D8	D7	D6	D5	C8	D4	D3	D2	C4	D1	C2	C1
Word	1	1	1	1		1	0	0		0		
Check Bit												

The check bits are in bit number 8, 4, 2 and 1

Check bit 8 (Bit positions: 12, 11, 10, 9)

$$1 \oplus 1 \oplus 1 \oplus 1 = 0$$

Check bit 4 (Bit positions: 12, 7, 6, 5)

$$1 \oplus 1 \oplus 0 \oplus 0 = 0$$

Check bit 2 (Bit positions: 11, 10, 7, 6, 3)

$$1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 1$$

Check bit 1 (Bit positions: 11, 9, 7, 5, 3)

$$1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 1$$

The check bits are: 0 0 1 1