CS1021 Tutorial #7 Solution
Using Memory

1 Set Intersection

Iterate over the elements of A. For every element, iterate over the elements of B to see if the same element is present. If it is, then the element should be included in the intersection – store it in Set C.

```
start
    LDR R2, =ASize   ; Load ASize address
    LDR R2, [R2]     ; Acount = Memory.Word[ASize]
    LDR R3, =AElems  ; Load AElems address

    LDR R1, =CElems  ; Load CElems address
    LDR R0, =0       ; Ccount = 0

whA CMP R2, #0       ; while (Acount > 0)
    BEQ eWhA         ; {
    LDR R6, [R3]     ;   tmpA = Memory.Word[AElems]

    LDR R4, =BSize   ;   Load BSize address
    LDR R4, [R4]     ;   Bcount = Memory.Word[BSize]
    LDR R5, =BElems  ;   Load Belems address

whB CMP R4, #0       ;   while (Bcount > 0)
    BEQ eWhB         ;          && tmpA != Memory.Word[Belems]
    LDR R7, [R5]     ;
    CMP R6, R7       ;
    BEQ eWhB         ;   {
    ADD R5, R5, #4   ;     Belems = Belems + 4
    SUB R4, R4, #1   ;     Bcount = Bcount - 1
    B   whB      ;   }
eWhB
    CMP R4, #0       ;   if (Bcount != 0)
    BEQ elf          ;   { // elem in A that is also in B
    STR R6, [R1]     ;     Memory.Word[Celems] = tmpA
    ADD R1, R1, #4   ;     Celems = Celems + 4
    ADD R0, R0, #1   ;     Ccount = Ccount + 1
elf             ;   }
    ADD R3, R3, #4   ;   Aelems = Aelems + 4
    SUB R2, R2, #1   ;   Acount = ACount -1
    B   whA      ; }
eWhA

    LDR R1, =CSize   ; Load CSize address
    STR R0, [R1]     ; Memory.Word[CSize] = Ccount

stop    B   stop

    AREA    TestData, DATA, READWRITE
ASize   DCD 8           ; Number of elements in Set A
AElems  DCD 7,20,9,17,3,2,23,13 ; Elements in Set A

BSize   DCD 6           ; Number of elements in Set B
BElems  DCD 6,13,11,2,25,10     ; Elements of Set B
```

```
48
49 CSize   DCD 0           ; Number of elements in Set C
50 CElems  SPACE   56          ; Space for elements of Set C
```

2 Unique Values

Iterate over each element of the sequence. For every element, iterate again over the elements from the start of the sequence up to the current element. If the same value is found in a different position, then the elements in the set are not unique.

```
1  COUNT   EQU 15
2
3  start
4      LDR R0, =1       ; unique = TRUE
5      LDR R1, =tstlst  ; addr1 = tstlist start address
6      LDR R2, =0       ; count1 = 0
7
8  wh1 CMP R2, #COUNT   ; while (count1 != COUNT
9      BEQ endwh1       ;        && unique == TRUE)
10     CMP R0, #1       ;
11     BNE endwh1       ; {
12     LDR R3, [R1]     ;   val1 = Memory.Word(addr1)
13     LDR R5, =tstlst  ;   addr2 = tstlist start address
14 wh2 CMP R5, R1       ;   while (addr2 != addr1
15     BEQ endwh2       ;          && val1 != Memory.Word(addr2))
16     LDR R4, [R5]     ;
17     CMP R3, R4       ;
18     BEQ endwh2       ;   {
19     ADD R5, R5, #1   ;     addr2 = addr2 + 4
20     B   wh2      ;   }
21 endwh2               ;
22     CMP R1, R5       ;   if (addr1 != addr2)
23     BEQ eifSameElem  ;   {
24     MOV R0, #0       ;     unique = FALSE
25 eifSameElem          ;   }
26     ADD R1, R1, #4   ;   addr1 = addr1 + 4
27     ADD R2, R2, #1   ;   count1 = count1 + 1
28     B   wh1      ; }
29 endwh1
30
31 stop    B   stop
32
33     AREA    TestData, DATA, READWRITE
34 tstlst  DCD 4, 9, 3, 4, 7, 9, 12, 10, 4, 7, 3, 12, 5, 5, 7
```