CS1021 Tutorial #7 Solution
Using Memory

1 Subset

Assume the result will be true (it is a subset). Iterate over set A. For each element in A and while the result is still true, check wether the same element appears in B. If we find a match, stop checking B and move on to the next element of A. If we don't find a match (get to the end of B) then set the result to false.

```
start
        LDR     R0, =1          ; isSubset = TRUE

        LDR     R4, =0          ; cA = 0
        LDR     R5, =Aelems     ; adrA = address Aelems
        LDR     R6, =Asize      ; tmp = address Asize
        LDR     R6, [R6]        ; nA = Memory.Word(tmp)

        LDR     R10, =Bsize     ; tmp = address Bsize
        LDR     R10, [R10]      ; nB = Memory.Word(tmp)

whA     CMP     R4, R6          ; while (cA < nA && isSubset == TRUE)
        BHS     eWhA            ; {
        CMP     R0, #1          ;
        BNE     eWhA            ;

        LDR     R7, [R5]        ;  eA = Memory.Word(adrA)

        LDR     R8, =0          ;  cB = 0
        LDR     R9, =Belems     ;  adrB = address Belems
        LDR     R11, [R9]       ;  eB = Memory.Word(adrB)

whB     CMP     R8, R10         ;  while (cB < nB && eB != eA)
        BHS     eWhB            ;  {
        CMP     R7, R11         ;
        BEQ     eWhB            ;

        ADD     R8, R8, #1      ;   cB++
        ADD     R9, R9, #4      ;   adrB++
        LDR     R11, [R9]       ;   eB = Memory.Word(adrB)

        B       whB             ;  }
eWhB
        CMP     R8, R10         ;  if (cB >= nB)
        BLO     endif           ;  {
        LDR     R0, =0          ;   isSubset = FALSE
endif                           ;  }
        ADD     R5, R5, #4      ;  adrA++
        ADD     R4, R4, #1      ;  cA++
        B       whA             ; }
eWhA


stop    B       stop
```

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.

```
COUNT   EQU     15

start
        LDR     R0, =1          ; unique = TRUE
        LDR     R1, =tstlst     ; addr1 = tstlist start address
        LDR     R2, =0          ; count1 = 0

wh1     CMP     R2, #COUNT      ; while (count1 != COUNT
        BEQ     endwh1          ;       && unique == TRUE)
        CMP     R0, #1          ;
        BNE     endwh1          ; {
        LDR     R3, [R1]        ;   val1 = Memory.Word(addr1)
        LDR     R5, =tstlst     ;   addr2 = tstlist start address
wh2     CMP     R5, R1          ;   while (addr2 != addr1
        BEQ     endwh2          ;         && val1 != Memory.Word(addr2))
        LDR     R4, [R5]        ;
        CMP     R3, R4          ;
        BEQ     endwh2          ;   {
        ADD     R5, R5, #1      ;     addr2 = addr2 + 4
        B       wh2             ;   }
endwh2                          ;
        CMP     R1, R5          ;   if (addr1 != addr2)
        BEQ     eifSameElem     ;   {
        MOV     R0, #0          ;     unique = FALSE
eifSameElem                     ;   }
        ADD     R1, R1, #4      ;   addr1 = addr1 + 4
        ADD     R2, R2, #1      ;   count1 = count1 + 1
        B       wh1             ; }
endwh1

stop    B       stop

        AREA    TestData, DATA, READWRITE
tstlst  DCD     4, 9, 3, 4, 7, 9, 12, 10, 4, 7, 3, 12, 5, 5, 7
```