Section 10.2 - Arrays

Layer 6: High-Order Language

Syllabus Content Section 10: Data Types and Structures

S10.2.1 Use the technical terms associated with arrays

- Including index, upper and lower bound

`Array index`: row or column number of an individual array element
`Upper bound`: the highest number index of an array dimension
`Lower bound`: the smallest number index of an array dimension

S10.2.2 Select a suitable data structure (1D or 2D array) to use for a given task

---

```
// A One-dimensional array
DECLARE <identifier>:ARRAY[<lower>:<upper>] OF <data type>
// A Two-dimensional array
DECLARE <identifier>:ARRAY[<lower1>:<upper1>,<lower2>:<upper2>] OF <data type>
```

S10.2.3 Write pseudocode for 1D and 2D arrays

```
// array declarations
DECLARE StudentNames : ARRAY[1:30] OF STRING
DECLARE NoughtsAndCrosses : ARRAY[1:3,1:3] OF CHAR
// Accessing individual array elements
```

```
StudentNames[1] ← "Ali"
NoughtsAndCrosses[2,3] ← 'X'
StudentNames[n+1] ← StudentNames[n]
```

S10.2.4 Write pseudocode to process array data

- Sort using a bubble sort
- Search using a linear search

> Bubble sort

```
DECLARE array:ARRAY[1:5] OF INTEGER
DECLARE size:INTEGER

Arr[1] <- 1
Arr[2] <- 6
Arr[3] <- 2
Arr[4] <- 5
Arr[5] <- 9


PROCEDURE BublesSort(arr:ARRAY,n:INTEGER) RETURN INTEGER
    FOR i <- 1 TO n
        FOR j <- 1 TO n-i
            IF arr[j]>arr[j+1]
                DECLARE temp:INTEGER
                temp <- arr[j]
                arr[j] <- arr[j+1]
                arr[j+1] <- temp
            ENDIF
        ENDFOR
    ENDFOR
ENDPROCEDURE


size <- SIZEOF(Arr) / SIZEOF(Arr[1])
BublesSort(array,size)
FOR i <- 1 TO size
    OUTPUT array[i]," "
NEXT i
```

> Linear Search

```
DECLARE Arr:ARRAY[1:5] OF INTEGER
DECLARE size:INTEGER
DECLARE num:INTEGER
DECLARE result:INTEGER

FUNCTION linearSearch(arr:ARRAY,s:INTEGER,n:INTEGER) RETURN INTEGER
    FOR i <- 1 to n
        IF arr[i]=n
            RETURN i
        ENDIF
    NEXT i
    RETURN -1
ENDFUNCTION

Arr[1] <- 3
Arr[2] <- 4
Arr[3] <- 1
Arr[4] <- 7
Arr[5] <- 5

size <- SIZEOF(Arr) / SIZEOF(Arr[1])

OUTPUT "what number you want to search?"
INPUT num

result <- linearSearch(Arr,size,num)

IF result=-1
    OUTPUT "Element is not present in the array."
ELSE
    OUTPUT "Element is present at index ", result

//时间复杂度O(n)
//空间复杂度O(1)
```