

## **PROGRAMMING EXERCISE 8 (Recursive binary search)**

The following is a pseudocode algorithm for a binary search on an array **data**.

This array stores data of integers sorted in ascending order.

```
## Binary Search Algorithm.
## INPUT:
##     data is an array of integers SORTED in ASCENDING order,
##     toFind is the integer to search for,
##     start is the minimum array index,
##     end is the maximum array index
## OUTPUT:
##     position of the integer toFind within array data, -1 if not found

PROCEDURE Recursive_binary_search(data, toFind, start, end)
    mid = (start + end)/2
    IF (start > end) THEN
        return -1
    ELSEIF (data[mid] = toFind)
        return mid
    ELSEIF (data[mid] > toFind)
        return Recursive_binary_search(data, toFind, start, mid-1)
    ELSE
        return Recursive_binary_search(data, toFind, mid+1, end)
    ENDIF
ENDPROCEDURE
```

### **Task 1**

Write program code for this algorithm and **follow good programming practice**.

**Evidence 1:** Your program code.

### **Task 2**

Write a **main** function that calls the above function, and displays **either** the position of the search item or **“item not found”** statement. An array **score** is given in the file **score.txt**. Copy and paste the array from the file and **search for item 82** from the array **score**.

**Evidence 2:** Your **main** function program code.

**Evidence 3:** One screenshot showing the output from running the program code for array **score**.

### **Task 3**

Amend the program code by deleting the array **score** and read in the array of numbers from the file **newscore.txt** instead. **Search for item 569 from the file**.

**Evidence 4:** Your amended program code.

**Evidence 5:** One screenshot showing the output from running the program code for **newscore.txt**.