

# SL Unit 4 – Problem Solving

## Quiz 3

### Question 1

Objectives: 4.2.5

Exam Reference:

May-16 10

Trace the following fragment, for  $N=139$  and  $L=3$ , by copying and completing the trace table given below.

```
D = N div L
Z = 1
B = false

loop while Z<L
    D = D div L
    Z = Z+1
    B = NOT B
end loop
if (D ≠ 0 AND B) then
    output(D, B)
else
    output(Z, NOT B)
end if
```

D	Z	B	Z<L ?	output
...	...	...	...	...

[4]

**Award [1] for correct output.**

**Award [1] for each of the three correct iterations (three rows).**

**Note: Accept 129/3, 139/9 and 139/27 D.**

D	Z	B	Z<L ?	output
46	1	false	true	
15	2	true	true	
5	3	false	false	
				(3 , true)

## Question 2

Objectives:	4.2.5, 4.2.6, 4.2.7, 4.3.13	Exam Reference:	May-17 12
-------------	-----------------------------	-----------------	-----------

1. (a) By copying the table below, trace the following algorithm using the data in the collection DATA. Note: B and C are also collections and are initially empty.

```

DATA = {2, 4, -1, 3}
loop while DATA.hasNext()
  A = DATA.getNext()
  if A >= 0 then
    if A mod 2 = 0 then
      B.addItem(A)
    else
      C.addItem(A)
    end if
  end if
end loop

```

DATA.hasNext() ?	A	A >= 0?	A mod 2 = 0?	Contents of B	Contents of C

[3]

Award [1] mark if evident from the trace table that the loop executes exactly 4 times

[1] mark for correct contents of B (2 and 4)

[1] mark for the correct contents of C

DATA.hasNext() ?	A	A >= 0 ?	A mod 2 = 0?	Contents of B	Contents of C
true	2	true	true	2	
true	4	true	true	2, 4	
true	-1	false	*true	*2, 4	
true	3	true	false	*2, 4	3
false					

*Note: \* These cells could be left empty*

- (b) Outline the steps involved in performing a binary search on an array of ascending numbers.

**Note:** you can assume that the search value is present in the array and that initially *LOW* is the index of the first value in the array and *HI* is the index of the last value.

[4]

*Award [4] marks as follows:*

A position/index of the middle value calculated from HI and LOW;  
If search value equals to value in the array at this position, then end;  
Otherwise, change HI or LOW according to whether search value is above or below;  
Repeat the process until search value is found;

Example answer 1:

1. Calculate MID (the midpoint of LOW and HI)
2. If ARRAY[MID] is the search value, end, else
3. If search value < than ARRAY[MID], HI = MID-1, else LOW = MID+1
4. Repeat steps 1, 2 and 3 until found

Example answer 2:

Find/calculate the midpoint/median value between HI and LOW;  
If the array value at midpoint equals the search value; then the search value is found;  
If search value is greater than the array value at midpoint, set the midpoint as a new LOW and repeat the process - if search value is less than the array value at midpoint, set midpoint as a new HI;  
Repeat until search value is found;

- (c) A collection called *NUMBERS* is to be searched to see if it contains a specified value. Construct an algorithm in pseudocode to perform the following:

- input the number, *S*, to be searched for
- read in the values from the *NUMBERS* collection into the array *D*.  
**Note:** you can assume that the array is large enough and that the collection is **not** empty
- perform a linear search for *S* on the array *D*
- output the message “*found*” or “*not found*” as appropriate.

[5]

Award up to [5 max] as follows.

Example answer 1:

[1] mark for a loop through collection NUMBERS

[1] mark for the assignment into the array from the collection// do not accept NUMBERS as array

[2] marks for a correct linear search , 1 mark for the loop and 1 mark for the comparison and use of flag

[1] mark for correct input of search value and output message

```
input S // search item
COUNT = 0
loop while NUMBERS.hasNext()      // or ...while NOT
                                   //NUMBERS.isEmpty()
    D[COUNT] = NUMBERS.getNext()
    COUNT = COUNT + 1
end loop
FOUND = false
X = 0
loop while (X<COUNT) and (FOUND == false) //accept for loop
    if S == D[X]
        FOUND = true
        X = X + 1
    end if
end loop
if FOUND == true
    output "found"
else
    output "not found"
end if
```

[5]

A binary search can be performed on the array  $D$ , if the values in  $D$  are in ascending order.

As the values are being read from  $NUMBERS$  into  $D$  they are checked to see if they are in order.

Without writing pseudocode, suggest how this check could be performed.

[3]

Award up to [3 max].

[1] mark for setting and changing a flag

[1] mark for starting from the 2nd entry and checking all values as being entered

[1] mark for comparing the current value with the previous one

**Note:** Do not accept pseudocode/code.

Question 3			
Objectives:	4.3.4	Exam Reference:	Nov-14 4

Outline the need for higher level languages. [2]

Question 4			
Objectives:	4.1.3	Exam Reference:	May-15 3

Outline how a sub-procedure can be considered an example of abstraction. [2]

Question 5			
Objectives:	4.3.3	Exam Reference:	Nov-17 1

Identify **two** essential features of a computer language. [2]