

End of topic quiz – Topic 2.1 Algorithms

1.

- a. Give examples of how **decomposition** can be used when thinking computationally [2 marks].

- b. A theme park uses a 3D computer simulation of a rollercoaster. Riders must wear a virtual reality headset to experience the ride.

Suggest how **abstraction** could be used giving examples for this simulation [2 marks].

2.

- a. (i) The array *people* contains the values:

`["Imogen", "Fletcher", "Kirstie", "Zoe", "Gavin"]`

Why can you not use a binary search on this array? [1 mark]

- a.(ii) Once the issue identified in part a.(i) has been resolved, describe the steps that would be taken to search the array for the value "Fletcher" using a binary search. [2 marks].

- b. (i) The algorithm below uses a different method to search through the array for a name.

Fill in the gaps to complete the algorithm. [4 marks].

```
array people[5]
people = ["Imogen", "Fletcher", "Kirstie", "Zoe", "Gavin"]
found = False
x = 0
searchfor = input("Enter a name to search for : ")
while found == False ..... x < 5
    if people[x] == searchfor then
        found = .....
        print "found at position " + .....
    endif
    x = x + 1
```

.....

- b.(ii) What is the name of this searching algorithm? [2 mark]

- c. A user has a database of 100,000 people and needs to search through to find one particular person.

Compare the use of **both** searching algorithms covered in parts a and b for a data set of this size. [4 marks].

3.

a.(i) A programmer has a list of numbers in an array called *scores*, as shown below:

17	9	4	-12	3	39
-----------	----------	----------	------------	----------	-----------

When setting up a bubble sort algorithm for these numbers, the programmer uses a variable called *swaps* which can either be True or False.

What is the data type of the variable *swaps*? [1 mark]

a.(ii) Describe the use of this variable when implementing the bubble sort. [2 marks].

b. One section of the bubble sort algorithm used by the programmer is shown below:

```
if scores[x] > scores[x + 1] //if scores in wrong order
    scores[x] = scores[x+1]
    scores[x+1] = scores[x] // swap numbers over
endif
```

What is the error that is contained in the code above? Provide a corrected version. [2 marks].

c. How would an insertion sort algorithm arrange the numbers in the *scores* array into order? [2 marks].

d. Name one other sorting algorithm. [1 mark]

e. What is **one advantage** and **one disadvantage** of using a bubble sort? [2 marks].

4. A school divides students into house groups based on the month that they were born in. Students born in January, February, March or April are put into Needwood house. Students born in May, June, July or August are put into Marchington House. All other students are put into Trent house.

Using pseudocode, write an algorithm that will:

- Ask the user to enter a number (1 to 12) relating to their birth month.
- Decide which house they are in and print this out.
- Keep a running total of how many students are in each house.
- Repeat the above for 20 students.
- When 20 students have entered their details, print out how many students are in each house.

[4 marks].

5.

a.

```
num = 3  
for x = 1 to num  
  print x * num  
next x
```

Sketch a flowchart version of this algorithm. [4 marks].





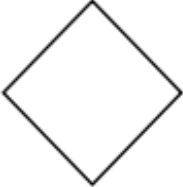

b.

```
num = 3  
for x = 1 to num  
  print x * num  
next x
```

Complete the trace table for the program code. [3 marks].

	num	x	Output
1			
2			
3			

6. Finish the following table to describe the use of each of the following flow chart symbols.
[4 marks].

Symbol	Description of use
	
	
	
	

7. A bicycle dealer uses the following algorithm to determine the price to charge for bicycles.

```

01 p = input("purchase price of bicycle")
02 i = input("number of improvements made")
03 a = input("age of bicycle in years")
04 s = p + (i * 100)
05 if a <= 10 then
06     s = s + s
07 endif
08 print "sale price is " + s

```

Work out the output value with the following inputs:

a. (i) p = 1000, i = 2, a = 12

[1 mark]

a. (ii) $p = 5000, l = 3, a = 10$

[1 mark]

a. (iii) $p = 8000, l = 0, a = 5$

[1 mark]

b. rewrite line 06 so that the + operator is **not** used.

[1 mark]

/46

Answers

1.

- a. Give examples of how **decomposition** can be used when thinking computationally.

Used to break down a (complex) problem...
...into small parts/component parts
these parts are easier to solve/understand than the larger problem.

- b. A theme park uses a 3D computer simulation of a rollercoaster. Riders must wear a virtual reality headset to experience the ride.

Suggest how **abstraction** could be used giving examples for this simulation.

a representation of a concept/object/thing...
...in this case, the rollercoaster.

Picks out the important/relevant parts/components/ideas/details...
...in this scenario, the track/rider/car/physics/etc.

Ignores/hides details which are not important/relevant...
...in this scenario, the queues/weather/smells/etc.

2.

- a. (i) The array *people* contains the values:

["Imogen", "Fletcher", "Kirstie", "Zoe", "Gavin"]

Why can you not use a binary search on this array?

Values are not in (alphabetical) order.

- a. (ii) Once the issue identified in part (i) has been resolved, describe the steps that would be taken to search the array for the value "Fletcher" using a binary search.

Array when sorted will be ["Fletcher", "Gavin", "Imogen", "Kirstie", "Zoe"].
Take the middle value and compare it to the item to be searched for...
...in this case, compare "Fletcher" (search data) to "Imogen".
Stop/return value/if data found.
Discard everything above this if the search data is smaller/discard everything below this if the search data is larger.
...in this case, "Fletcher" is smaller (alphabetically) than "Imogen" so discard top half of array.
Repeat/recursively call search routine again on remaining values in array.

- b.(i) The algorithm below uses a different method to search through the array for a name.

Fill in the gaps to complete the algorithm.

```

array people[5]
people = ["Imogen", "Fletcher", "Kirstie", "Zoe", "Gavin"]
found = False
x = 0
searchfor = input("Enter a name to search for : ")
while found == False AND x < 5
    if people[x] == searchfor then
        found = True
        print "found at position " + x
    endif
    x = x + 1

endwhile

```

- b.(ii) What is the name of this searching algorithm?

Linear (search).

- c. A user has a database of 100,000 people and needs to search through to find one particular person.

Compare the use of **both** searching algorithms covered in parts a and b for a data set of this size.

Linear search compares each value in turn.
 Worst case scenario, 100,000 comparisons/all values checked.
 Binary search splits size of list in half each time/uses divide and conquer.
 Worst case scenario, many fewer comparisons (approx. 17 comparisons).
 Binary search is more efficient/linear search is less efficient.

3.

- a.(i) A programmer has a list of numbers in an array called *scores*, as shown below:

17	9	4	-12	3	39
----	---	---	-----	---	----

When setting up a bubble sort algorithm for these numbers, the programmer uses a variable called *swaps* which can either be True or False.

What is the data type of the variable *swaps*?

Boolean.

- a.(ii) Describe the use of this variable when implementing the bubble sort.

Initialised to False at the start of the algorithm.
 If pairs of numbers are swapped, set to True.
 When all numbers have been compared, check again...
 ... if swaps is True, repeat algorithm/process again
 ...setting swaps back to False.

- b. One section of the bubble sort algorithm used by the programmer is shown below:

```
if scores[x] > scores[x + 1] //if scores in wrong order
    scores[x] = scores[x + 1]
    scores[x + 1] = scores[x] // swap numbers over
endif
```

What is the error that is contained in the code above? Provide a corrected version.

Error
 numbers are not swapped correctly/scores[x] will be overwritten
 Correction (eg..)
 temp = scores[x]
 scores[x] = scores[x+1]
 scores[x+1] = temp

- c. How would an insertion sort algorithm arrange the numbers in the *scores* array into order?

Take first number (17) as a sorted list by itself.
 Look at next number (9) and insert into the correct place...
 ...method for doing this (either repeatedly swapping until in right place or comparing against number in sorted list and moving).
 Continue until last number is processed/repeat for each number in the list.

- d. What is the name of one **other** sorting algorithm?

Merge sort.
 Also accept other sorts that are not in the GCSE specification (e.q.Quick sort)

- e. What is **one advantage** and **one disadvantage** of using a bubble sort?

Advantage : easier/simpler/faster to implement (than other sorting algorithms).
 Disadvantage : Less efficient/takes longer to run (than other sorting algorithms).

4. A school divides students into house groups based on the month that they were born in. Students born in January, February, March or April are put into Needwood house. Students born in May, June, July or August are put into Marchington House. All other students are put into Trent house.

Using pseudocode, write an algorithm that will:

- Ask the user to enter a number (1 to 12) relating to their birth month
- Decide which house they are in and print this out.
- Keep a running total of how many students are in each house.
- Repeat the above for 20 students.
- When 20 students have entered their details, print out how many students are in each house.

Initialising variables at the start for count of students in three houses
 Suitable loop that repeats 20 times
 Inputting the birth month as a number
 Printing out message AND adding 1 to counter if birth month is between 1 and 4
 Printing out message AND adding 1 to counter if birth month is between 5 and 8
 Printing out message AND adding 1 to counter if birth month is between 9 and 12
 Printing out totals for all three counters at the end.

Example:

```
N = 0
M = 0
T = 0
for x = 1 to 20
    input birthnum
    if birthnum >= 1 and birthnum <= 4 then
        print "Needwood house"
        N = N + 1
    elseif birthnum >= 5 and birthnum <= 8 then
        print "Marchington house"
        M = M + 1
    elseif birthnum >= 9 and birthnum <= 12 then
        print "Trent house"
        T = T + 1
    endif
next x
print N, M, T
```

Must be pseudocode, not flowchart (asked in).

Mark for loop only given if correct code is inside/outside the loop. Accept WHILE/DO loop as long as it repeats 20 times.

Only allow use of ELSE for between 9 and 12 if suitable validation is used to restrict answers to between 1 and 12.

Allow ECF for 4th and 5th bullet points if something is not correct with 3rd bullet point.

5.

a.

```
num = 3
  for x = 1 to num
    print x * num
  next
```

Draw a flowchart version of this algorithm.

Start and end symbols (both present and correct shape)

num initially given value 3
x initially given value 1
outputting the result of $x * \text{num}$ (even if previous steps missed)
Deciding whether to loop again correctly (ie if not yet looped 7 times)
...incrementing x by 1 if appropriate
... looping to the correct position in the algorithm if appropriate
...stopping if at end of loop/already repeated 7 times

correct shapes for flowchart symbols must be used.
Decision box must be labelled up with at least one of YES/NO (allow BOD if only 1 labelled).
Allow different variable names as long as the algorithm would logically work.

```

graph TD
    Start([Start]) --> Num[num = 3]
    Num --> X[x = 1]
    X --> Print[/print x * num/]
    Print --> Decision{Is x >= num?}
    Decision -- NO --> Inc[x = x + 1]
    Inc --> Print
    Decision -- YES --> End([End])
    
```



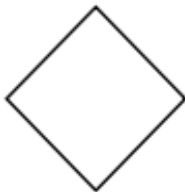

b.

```
num = 3
for x = 1 to num
  print x * num
next
```

Complete the trace table for the program code.

	Num	X	print
1	7	1	7
2	7	2	14
3	7	3	21

6. Finish the following table to describe the use of each of the following flow chart symbols.

Symbol	Explanation of use
	Process Used to work out calculations/assignment/instructions that have no input or output Suitable example (eg $x = x + 7$)
	Input/output Used to get information from /give information to <u>the user</u> Suitable example (eg print "hello")
	Decision Used to make yes/no decisions/choices Suitable example (eg IF $x > 10$)
	Terminator/start/stop symbol Used at the very beginning/very end of the program/algorithm.

7. A bicycle dealer uses the following algorithm to determine the price to charge for cars.

```

01 p = input("purchase price of car"           02 i =
input("number of improvements made")
    03 a = input("age of car in years")
    04 s = p + (i*100)
    05 if a <= 10 then
    06   s = s + s
    07 endif
    08 print "sale price is " + s

```

Work out the output value with the following inputs:

a. (i) p = 1000, i = 2, a = 12

1200

a. (ii) p = 5000, i = 3, a = 10

10600

a. (iii) p = 8000, i = 0, a = 5

16000

b. rewrite line 06 so that the + operator is **not** used.

S=s*2
S=s^2 (accept s=s**2, Python syntax for this)

OCR Resources: the small print

OCR's resources are provided to support the delivery of OCR qualifications, but in no way constitute an endorsed teaching method that is required by the Board, and the decision to use them lies with the individual teacher. Whilst every effort is made to ensure the accuracy of the content, OCR cannot be held responsible for any errors or omissions within these resources.

Our documents are updated over time. Whilst every effort is made to check all documents, there may be contradictions between published support and the specification, therefore please use the information on the latest specification at all times. Where changes are made to specifications these will be indicated within the document, there will be a new version number indicated, and a summary of the changes. If you do notice a discrepancy between the specification and a resource please contact us at: resources.feedback@ocr.org.uk.

© OCR 2020 - This resource may be freely copied and distributed, as long as the OCR logo and this message remain intact and OCR is acknowledged as the originator of this work. OCR acknowledges the use of the following content: n/a
Please get in touch if you want to discuss the accessibility of resources we offer to support delivery of our qualifications: resources.feedback@ocr.org.uk