Oxford Cambridge and RSA

A Level Computer Science

H446/02 Algorithms and programming

Sample Question Paper

Date – Morning/Afternoon

Time allowed: 2 hours 30 minutes

Do not use:
- a calculator

First name												
Last name												
Centre number							**Candidate number**					

INSTRUCTIONS
- Use black ink.
- Complete the boxes above with your name, centre number and candidate number.
- Answer **all** the questions.
- Write your answer to each question in the space provided.
- Additional paper may be used if required but you must clearly show your candidate number, center number and question number(s).
- Do **not** write in the bar codes.

INFORMATION
- The total mark for this paper is **140**.
- The marks for each question are shown in brackets **[ ]**.
- Quality of extended responses will be assessed in questions marked with an asterisk (*).
- This document consists of **24** pages.

Section A

Answer **all** questions.

1 The map shown below in Fig.1 is of a city and surrounding districts. The map is an *abstraction* made up of a number of component parts.

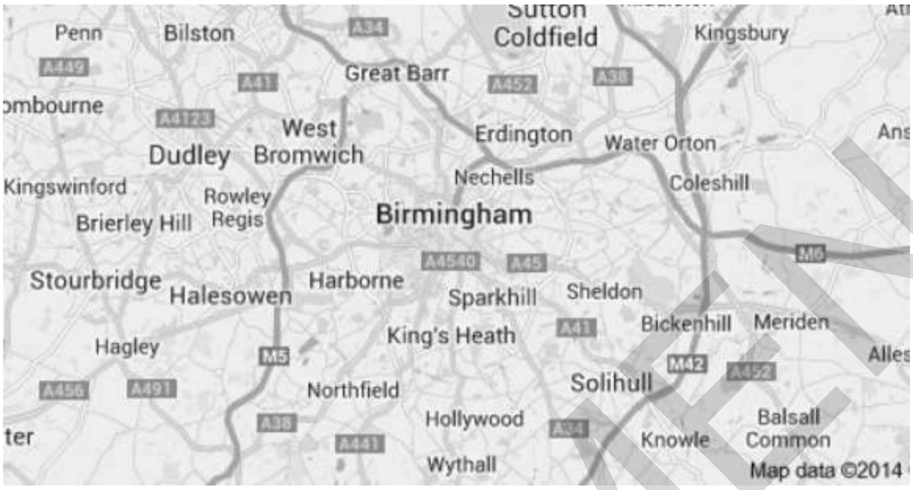


Fig.1

(a) Using the map in Fig.1 as an example, define the term *abstraction*.

……

……

……

……

……**[4]**

(b) Both map making and program development make use of reusable components.

(i) Give **three** examples of how reusable component parts are used in Fig.1.

1……

……..

2……

……..

3……**[3]**

(ii) Explain how programmers make use of reusable components when developing large programs.

……

……

……

……

……**[3]**

(c) Explain **two** advantages to programmers of using reusable components when developing programs.

1……………………………………………………………………………………………………….

……

……

……

……

……

2……………………………………………………………………………………………………….

……

……

……

……

……**[4]**

2 Consider the following algorithm in Fig.2, expressed in pseudocode, as a function S:

```
function S(A[0..N-1], value, low, high)

    if (high < low)then
          return error_message
     endif

     mid = (low + high) / 2

    if (A[mid] > value)then
           return S(A, value, low, mid-1)
    elseif (A[mid] < value)then
          return S(A, value, mid+1, high)
    else
          return mid
    endif

endfunction
```

Fig.2

(a) State the name of the algorithm implemented in Fig.2.

....**[1]**

(b) Describe the purpose of this algorithm.

....

....

....

....**[2]**

(c) Parameters are passed to this function. Complete the following table to identify these parameters and the purpose of each.

Parameter name	**Purpose**

[8]

(d) **(i)** Describe what is meant by recursion.

....

....

....

....**[2]**

(ii) Identify **one** example of where recursion occurs in this algorithm.

....

....**[1]**

(e) Explain how the algorithm in Fig.2 is an example of a divide and conquer approach.

....

....

....

....

....**[3]**

(f) Rewrite the algorithm in Fig.2 without using recursion. Annotate your pseudocode with comments to show how it solves the problem.

[8]

3 Julie wants to earn her living by being a successful app developer.

Before she even writes any code, she thinks it would be sensible to find out some basic facts about app development and the market for apps in order to maximise her chances of being successful.

(a) State **four** items of data that she could obtain in order to make a sensible choice of an app development project.

1……………………………………………………………………………………………………

2……………………………………………………………………………………………………

3……………………………………………………………………………………………………

4……………………………………………………………………………………………………**[4]**

(b)* *'It is possible to use computational methods as a way of predicting the success of an app.'* Discuss the extent to which you agree with this statement.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....**[9]**

4 A supermarket chain stores over 3 petabytes (10^{15} bytes) of data about sales and customers. The supermarket chain carried out a data mining exercise in which they discovered that whenever there was a hurricane warning, sales of fruit pies increased. This had not been noticed before the data mining exercise. The next time there was a hurricane warning, they placed the fruit pies at the end of the aisles and there was a dramatic increase in sales.

Explain how computational methods were able to reveal this unexpected result.

....

....

....

....

....

....**[4]**

5 (a) Describe what is meant by a *heuristic* approach to problem-solving.

....

....**[2]**

(b) Describe how heuristic methods are used when making a decision about when to cross a busy road.

....

....

....

....**[3]**

(c)* Virus checkers work by looking for patterns in the program code. They also use heuristic approaches. Evaluate the effectiveness of heuristic and pattern matching approaches to virus detection.

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……**[9]**

6 Fig.3 shows a plan of a cafeteria. Customers are complaining that it currently takes too long to collect their drinks.

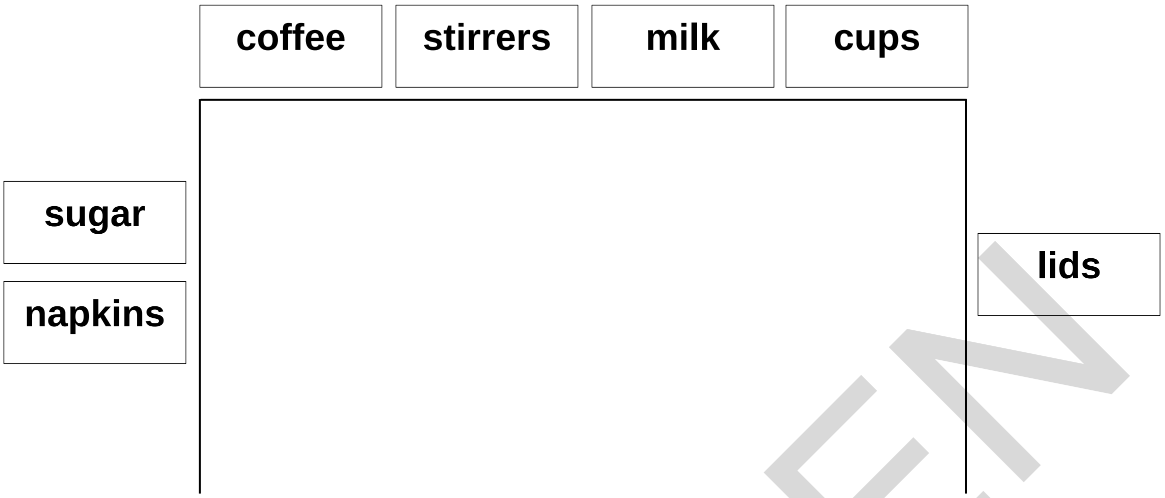


Fig.3

The cafeteria manager is thinking about reorganising the layout of the cafeteria in order to speed up the service to customers.

(a) State **three** data items that could usefully be collected in order to investigate the problem.

1………

………..

2………

………..

3………

………..**[3]**

(b) Explain how computational methods could be used in order to improve the layout of this cafeteria.

....

....

....

....

....

....

.... **[4]**

7 When a house is being built, the following activities take place:

- plans are drawn
- foundations are laid
- bricks are ordered
- bricks are delivered
- walls are built
- windows are installed
- electric wiring is installed
- plumbing is installed
- roof rafters are installed
- tiles are put on roof.

(a) Describe the term *pipelining*.

……

……

……**[2]**

(b) Explain how pipelining principles can be used to ensure that a house is built as quickly as possible.

……

……

……

……

……**[4]**

(c) Describe **two** examples of where pipelining is used in any computer system.

1……

……..

……..

2……

……..

……..

……..

[4]

8 It is possible to use XOR together with a key to encrypt a plain text message.

For example, to encrypt the bit pattern 111100001111 using the number 5 as a key, the key is repeated as often as necessary to match the length of the message. An XOR operation is performed to generate the encoded message.

111100001111 <- Message to be encrypted

101101101101 <- Key: repeated as necessary

010001100010 <- Encrypted message

Write an algorithm to accept a text message and a key that would use the method above to generate an encoded message. Annotate your pseudocode with comments to show how it solves the problem.

[8]

9 Linear search and binary search are two different algorithms which can be used for searching arrays.

When comparing linear and binary search it is possible to look at the best, worst and average number of items in the array that need to be checked to find the item being searched for. Assume every item in the array is equally likely to be searched for.

(a) Complete the table below

	Worst Case number of searches	Average Case	Best Case
Binary Search		$\log_2(n)$-1	
Linear Search		n/2	

[4]

(b) As the size of an array increases the average number of checks grows logarithmically. State what is meant by logarithmic growth.

……

……**[1]**

(c) Assuming an array is sorted give a situation when a linear search would perform better than a binary search.

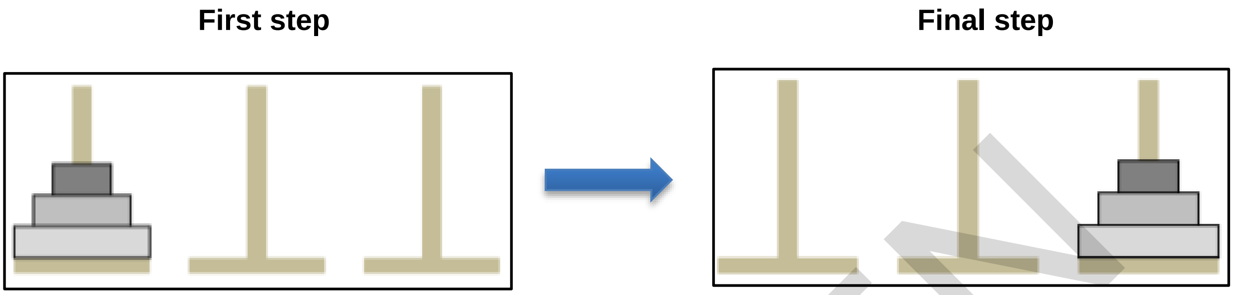
……

……**[1]**

Section B

Answer **all** questions.

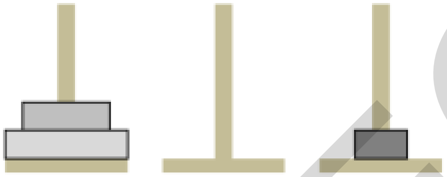
10 The Towers of Hanoi is a classic puzzle. Disks are placed in order on a pole, the biggest disc at the bottom of the pole, the smallest disk at the top of the pole, on the first of three poles. The challenge is to get them to the third pole in the same order.



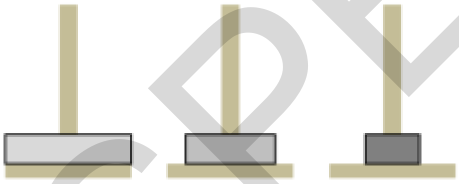
First step **Final step**

The disks can only be moved under the following rules:

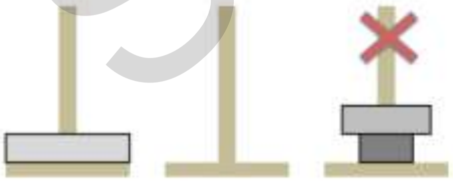
- only one disk can be moved at a time
- a disk can only ever be placed on an empty pole or on top of a larger disk
- a larger disk can never be placed on a smaller disk.



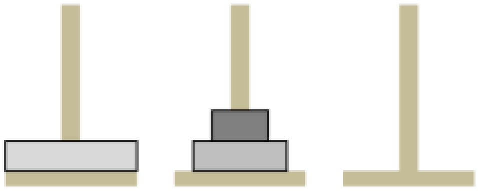
This is a valid move.



This would be a valid second move.



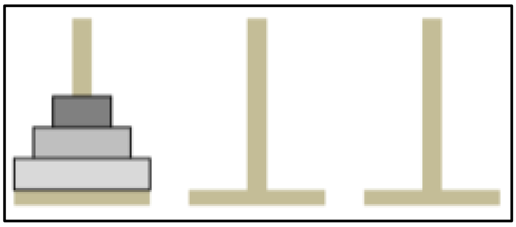
This would **not** be a valid third move (you can't put a bigger disk on a smaller one.)



This **would** be a valid third move.

(a) Each disk can be represented by an integer denoting its size.

So



Can be represented by

1		
2		
3		

(i) Explain why you would use a *stack* rather than a *queue* to store the configuration of disks at each pole.

....

....

....**[2]**

(ii) The tower class has the method push. It takes in the value of the disk to be pushed. It adds it to the top of the stack if it is a valid move. If it is not a valid move, the value of the disc is not added and the message ‘Invalid move’ is printed to the screen.

The stack is implemented using an array called pole and an integer called pointer. Pointer represents the index of the array position at the top of the stack.

```
class Tower
      private array pole[10]
      private pointer

      public procedure new()

            pointer=0

      endprocedure

      public procedure push(diskValue)
            //Code for push method

      endprocedure
endclass
```

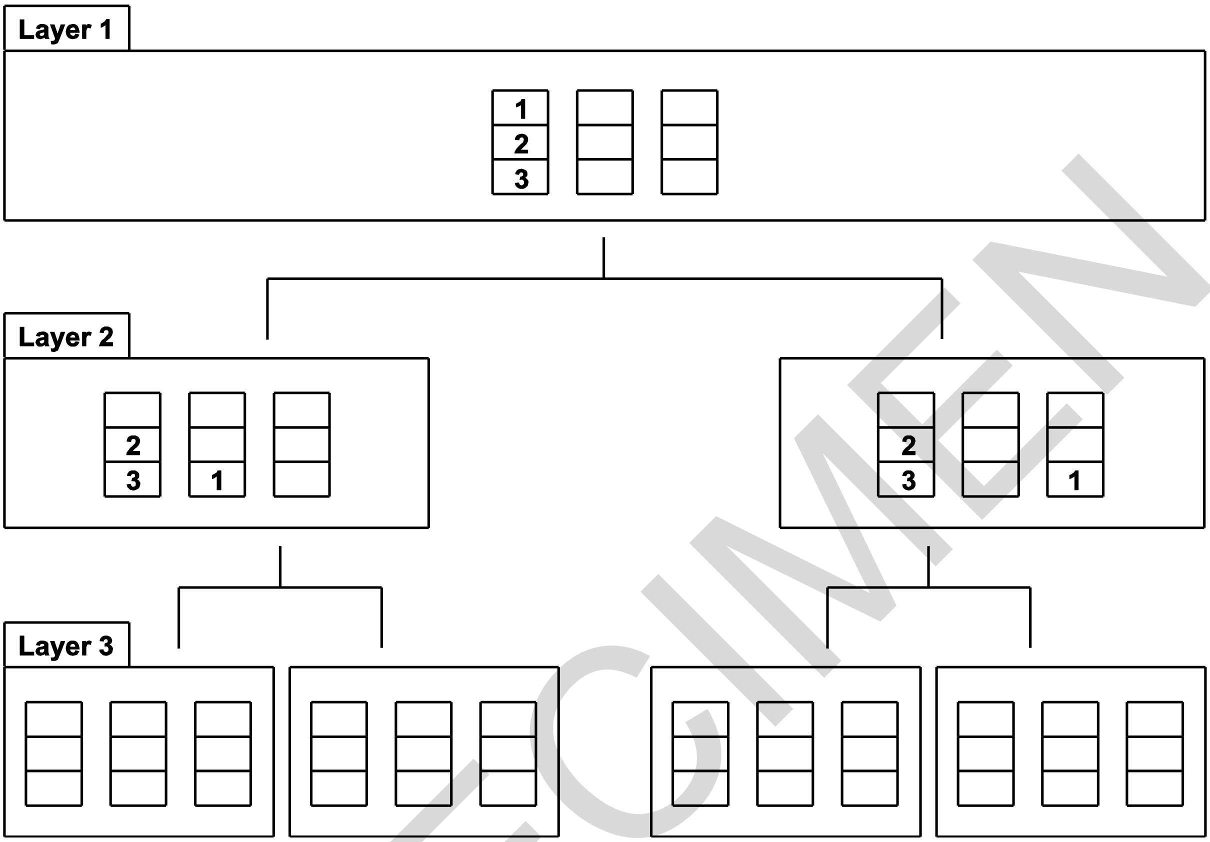
Write the pseudocode to go in the push method. Annotate your pseudocode with comments to show how it solves the problem. You are not expected to test for overflow.

[6]

(b) One way to try to find a solution would be to generate a tree of possible moves until a solution is found.

(i) A tree has been started below. Complete **Layer 3** to show 4 possible moves.

[4]



(ii) The search space represented by the tree could be searched using a depth first or breadth first search.

Describe **one** advantage and **one** disadvantage of depth-first search compared with breadth-first search.

Advantage:

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

Disadvantage:

……………………………………………………………………………………………………

……………………………………………………………………………………………………

…………………………………………………………………………………………………… …**[4]**

BLANK PAGE

(c) Rather than using a tree, the following iterative algorithm can be used to play the perfect game where the number of disks is odd. A similar algorithm exists for an even number of disks.

A program is required to solve the Towers of Hanoi puzzle using the iterative algorithm below.

Iterative algorithm to solve Towers of Hanoi for an odd number of disks.

Cycle through the following three steps until the puzzle is solved (which may be after any of the steps):

- make the valid move between tower1 and tower3
- make the valid move between tower1 and tower2
- make the valid move between tower3 and tower2

NB: The valid move might be in either direction but there will only be one possibility each time.

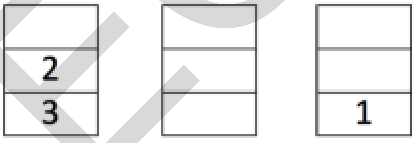
In this program there are three objects of the class Tower; `tower1, tower2` and `tower3`

The Tower class has the methods `push, peek` and `pop`

The method `push` adds a disk to the tower, for example `tower1.push(3)` adds a disk of size 3 to tower1.

The method `peek` returns the value of the disk on top of the tower but does not remove it. It returns the value 999 if the tower is empty.

The method `pop` removes a disk from a tower and returns the value of that disk.



e.g. `x=tower1.peek()` would make x equal to 2 and the towers stay the same.
`x=tower1.pop()` would make x equal to 2 and remove 2 from tower1.

Complete the pseudocode program below so when given an odd number of disks, below 100, on tower1 they will be moved to tower3 using the iterative algorithm. Annotate your pseudocode with comments to show how it solves the problem.

[10]

```
noOfDisks=5 //Can be set to any odd number below 100
      tower1 = new Tower()
      tower2 = new Tower()
      tower3 = new Tower()

      i=noOfDisks
      while i>0
            tower1.push(i)
            i=i-1
      endwhile

//add code to solve puzzle
```

(d) The complexity of solving the Towers of Hanoi can be expressed in Big O notation as $O(2^n)$ where n is the number of disks.

(i) A given computer takes 8 milliseconds (ms) to solve a 3 disk problem. Calculate how long the computer takes to solve a 5 disk problem.

……………………………………………………………………………………..………………….**[1]**

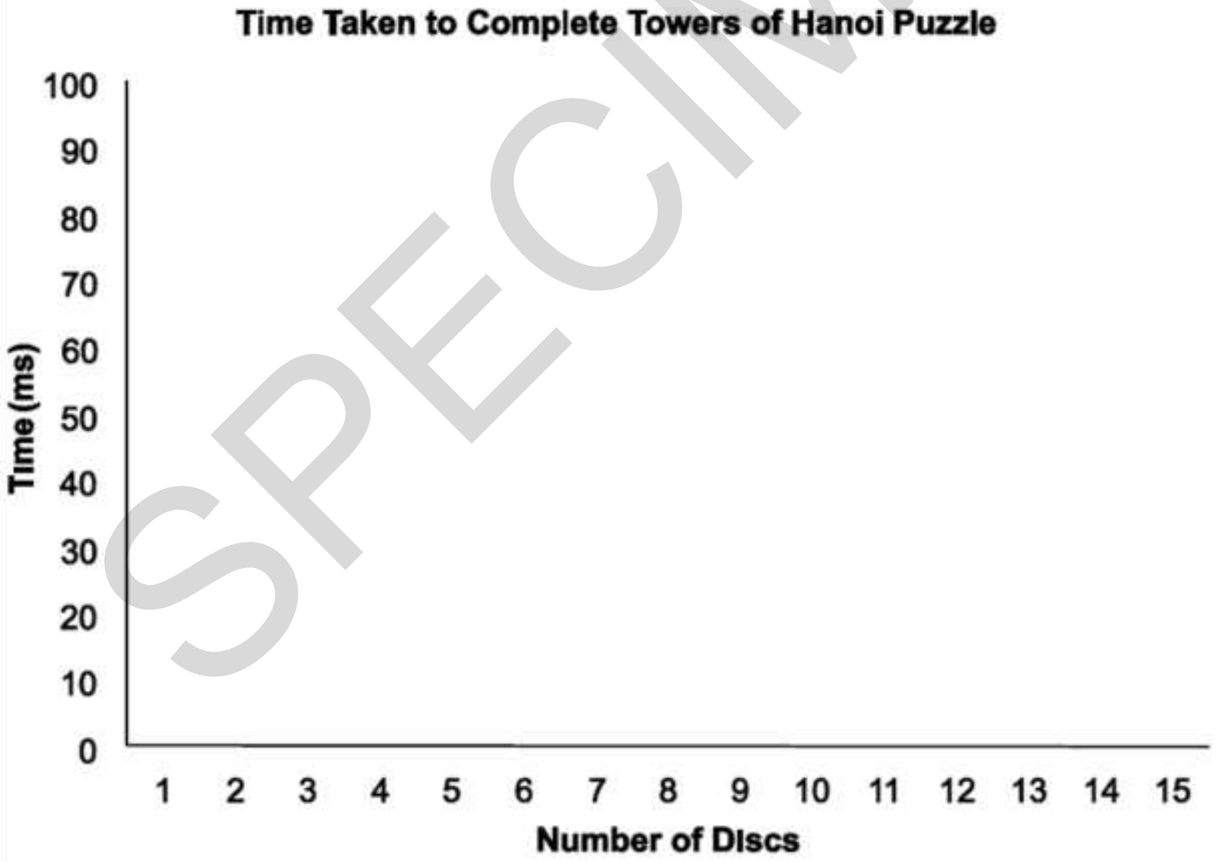
(ii) State **one** reason why the answer given for part **(i)** may only be an estimate.

………

…………………………………………………………………………………………………..…….**[1]**

(iii) Complete the graph below to show an estimate of how long a computer would take to solve the Towers of Hanoi with a variable number of discs.

[2]



(e)* Concurrency can be used to speed up the processing of some problems.

Discuss to what extent concurrency might be of use when solving the Towers of Hanoi puzzle. You should consider both the search space tree and iterative algorithm approaches in your answer.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....**[9]**

Copyright Information:

Q1: Map of Birmingham © taken from Google: https://services.google.com

…day June 2014 – Morning/Afternoon

A Level Computer Science
H446/02 Algorithms and programming

SAMPLE MARK SCHEME

Duration: 2 hours 30 minutes

MAXIMUM MARK 140

This document consists of 28 pages

MARKING INSTRUCTIONS

PREPARATION FOR MARKING

SCORIS

1. Make sure that you have accessed and completed the relevant training packages for on-screen marking: *scoris assessor Online Training*; *OCR Essential Guide to Marking*.

2. Make sure that you have read and understood the mark scheme and the question paper for this unit. These are posted on the RM Cambridge Assessment Support Portal http://www.rm.com/support/ca

3. Log-in to scoris and mark the **required number** of practice responses (“scripts”) and the **required number** of standardisation responses.

   YOU MUST MARK 10 PRACTICE AND 10 STANDARDISATION RESPONSES BEFORE YOU CAN BE APPROVED TO MARK LIVE SCRIPTS.

TRADITIONAL

Before the Standardisation meeting you must mark at least 10 scripts from several centres. For this preliminary marking you should use **pencil** and follow the **mark scheme**. Bring these **marked scripts** to the meeting.

MARKING

1. Mark strictly to the mark scheme.

2. Marks awarded must relate directly to the marking criteria.

3. The schedule of dates is very important. It is essential that you meet the scoris 50% and 100% (traditional 50% Batch 1 and 100% Batch 2) deadlines. If you experience problems, you must contact your Team Leader (Supervisor) without delay.

4. If you are in any doubt about applying the mark scheme, consult your Team Leader by telephone, email or via the scoris messaging system.

5. Work crossed out:
   a. where a candidate crosses out an answer and provides an alternative response, the crossed out response is not marked and gains no marks
   b. if a candidate crosses out an answer to a whole question and makes no second attempt, and if the inclusion of the answer does not cause a rubric infringement, the assessor should attempt to mark the crossed out answer and award marks appropriately.

6. Always check the pages (and additional objects if present) at the end of the response in case any answers have been continued there. If the candidate has continued an answer there then add a tick to confirm that the work has been seen.

7. There is a NR (No Response) option. Award NR (No Response)
   - if there is nothing written at all in the answer space
   - OR if there is a comment which does not in any way relate to the question (e.g. ‘can’t do’, ‘don’t know’)
   - OR if there is a mark (e.g. a dash, a question mark) which isn’t an attempt at the question.

   Note: Award 0 marks – for an attempt that earns no credit (including copying out the question).

8. The scoris **comments box** is used by your Team Leader to explain the marking of the practice responses. Please refer to these comments when checking your practice responses. **Do not use the comments box for any other reason.**
   If you have any questions or comments for your Team Leader, use telephone, email or the scoris messaging system.

9. Assistant Examiners will send a brief report on the performance of candidates to their Team Leader (Supervisor) via email by the end of the marking period. The report should contain notes on particular strengths displayed as well as common errors or weaknesses. Constructive criticism of the question paper/mark scheme is also appreciated.

10. For answers marked by levels of response:
    a. **To determine the level –** start at the highest level and work down until you reach the level that matches the answer
    b. **To determine the mark within the level**, consider the following:

Descriptor	Award mark
On the borderline of this level and the one below	At bottom of level
Just enough achievement on balance for this level	Above bottom and either below middle or at middle of level (depending on number of marks available)
Meets the criteria but with some slight inconsistency	Above middle and either below top of level or at middle of level (depending on number of marks available)
Consistently meets the criteria for this level	At top of level

11. **Annotations**

Annotation	Meaning

12. **Subject-specific Marking Instructions**

INTRODUCTION

Your first task as an Examiner is to become thoroughly familiar with the material on which the examination depends. This material includes:

- the specification, especially the assessment objectives
- the question paper and its rubrics
- the mark scheme.

You should ensure that you have copies of these materials.

You should ensure also that you are familiar with the administrative procedures related to the marking process. These are set out in the OCR booklet **Instructions for Examiners**. If you are examining for the first time, please read carefully **Appendix 5 Introduction to Script Marking: Notes for New Examiners**.

Please ask for help or guidance whenever you need it. Your first point of contact is your Team Leader.

USING THE MARK SCHEME

Please study this Mark Scheme carefully. The Mark Scheme is an integral part of the process that begins with the setting of the question paper and ends with the awarding of grades. Question papers and Mark Schemes are developed in association with each other so that issues of differentiation and positive achievement can be addressed from the very start.

This Mark Scheme is a working document; it is not exhaustive; it does not provide ‘correct’ answers. The Mark Scheme can only provide ‘best guesses’ about how the question will work out, and it is subject to revision after we have looked at a wide range of scripts.

The Examiners’ Standardisation Meeting will ensure that the Mark Scheme covers the range of candidates’ responses to the questions, and that all Examiners understand and apply the Mark Scheme in the same way. The Mark Scheme will be discussed and amended at the meeting, and administrative procedures will be confirmed. Co-ordination scripts will be issued at the meeting to exemplify aspects of candidates’ responses and achievements; the co-ordination scripts then become part of this Mark Scheme.

Before the Standardisation Meeting, you should read and mark in pencil a number of scripts, in order to gain an impression of the range of responses and achievement that may be expected.

In your marking, you will encounter valid responses which are not covered by the Mark Scheme: these responses must be credited. You will encounter answers which fall outside the ‘target range’ of Bands for the paper which you are marking. Please mark these answers according to the marking criteria.

Please read carefully all the scripts in your allocation and make every effort to look positively for achievement throughout the ability range. Always be prepared to use the full range of marks.

LEVELS OF RESPONSE QUESTIONS:

The indicative content indicates the expected parameters for candidates' answers, but be prepared to recognise and credit unexpected approaches where they show relevance.

Using 'best-fit', decide first which set of BAND DESCRIPTORS best describes the overall quality of the answer. Once the band is located, adjust the mark concentrating on features of the answer which make it stronger or weaker following the guidelines for refinement.

- **Highest mark:** If clear evidence of all the qualities in the band descriptors is shown, the HIGHEST Mark should be awarded.
- **Lowest mark:** If the answer shows the candidate to be borderline (i.e. they have achieved all the qualities of the bands below and show limited evidence of meeting the criteria of the band in question) the LOWEST mark should be awarded.
- **Middle mark:** This mark should be used for candidates who are secure in the band. They are not 'borderline' but they have only achieved some of the qualities in the band descriptors.

Be prepared to use the full range of marks. Do not reserve (e.g.) high Band 3 marks 'in case' something turns up of a quality you have not yet seen. If an answer gives clear evidence of the qualities described in the band descriptors, reward appropriately.

	AO1	**AO2**	**AO3 - Only AO3.3 is assessed in the external assessment**
High (thorough)	Precision in the use of question terminology. Knowledge shown is consistent and well-developed. Clear appreciation of the question from a range of different perspectives making extensive use of acquired knowledge and understanding.	Knowledge and understanding shown is consistently applied to context enabling a logical and sustained argument to develop. Examples used enhance rather than detract from response.	Concerted effort is made to consider all aspects of a system / problem or weigh up both sides to an argument before forming an overall conclusion. Judgements made are based on appropriate and concise arguments that have been developed in response resulting in them being both supported and realistic.
Middle (reasonable)	Awareness of the meaning of the terms in the question. Knowledge is sound and effectively demonstrated. Demands of question understood although at times opportunities to make use of acquired knowledge and understanding not always taken.	Knowledge and understanding applied to context. Whilst clear evidence that an argument builds and develops through response there are times when opportunities are missed to use an example or relate an aspect of knowledge or understanding to the context provided.	There is a reasonable attempt to reach a conclusion considering aspects of a system / problem or weighing up both sides of an argument. However the impact of the conclusion is often lessened by a lack of supported judgements which accompany it. This inability to build on and develop lines of

			argument as developed in the response can detract from the overall quality of the response.
Low (basic)	Confusion and inability to deconstruct terminology as used in the question. Knowledge partial and superficial. Focus on question narrow and often one-dimensional.	Inability to apply knowledge and understanding in any sustained way to context resulting in tenuous and unsupported statements being made. Examples if used are for the most part irrelevant and unsubstantiated.	Little or no attempt to prioritise or weigh up factors during course of answer. Conclusion is often dislocated from response and any judgements lack substance due in part to the basic level of argument that has been demonstrated throughout response.

The breakdown of Assessment Objectives for A Level in Computer Science:

	Assessment Objectives
AO1	Demonstrate knowledge and understanding of the principles and concepts of computer science, including abstraction, logic, algorithms and data representation.
AO1.1	Demonstrate **knowledge** of the principles and concepts of abstraction, logic, algorithm, data representation or other as appropriate.
AO1.2	Demonstrate **understanding** of the principles and concepts of abstraction, logic, algorithm, data representation or other as appropriate.
AO2	Apply knowledge and understanding of the principles and concepts of computer science including to analyse problems in computational terms.
AO2.1	Apply knowledge and understanding of the principles and concepts of computer science.
AO2.2	Analyse problems in computational terms.
AO3	Design, program and evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions.
AO3.1	Design computer systems that solve problems.
AO3.2	Program computer systems that solve problems.
AO3.3	Evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions.

Question			Answer	Marks	Guidance
1	(a)		• Abstraction is the process of separating ideas (1 – AO 1.1) from particular instances/reality (1 – AO 1.1). It is a means of hiding detail/only using relevant detail (1 – AO 1.1), it is a representation of reality (1 – AO 1.1), using symbols to show real-life features (1 – AO 2.1) or irrelevant features (e.g. such as buildings) left out (1 – AO 2.1).	**4** **AO1.1** **(2)** **AO2.1** **(2)**	Up to 4 marks for a valid definition. Up to 2 marks for demonstrating knowledge (AO1.1) Up to 2 marks for demonstrating application of knowledge and understanding (AO2.1).
	(b)	(i)	Examples: • Road/type of road symbol (1). • Road number symbol (1). • Colour coding (1). • Place labels (1).	**3** **AO1.2** **(3)**	1 mark for each correct identification up to a maximum of three identifications.
		(ii)	• Software is modular (1), an example being an object/function (1). Modules can be transplanted into new software (1) or can be shared at run time (1) through the use of program libraries (1).	**3** **AO1.2** **(3)**	Up to 3 marks for a valid explanation.
	(c)		• Modules already tested (1 – AO 1.2) so more reliable programs (1 – AO 2.1). • Less development time (1 – AO 1.2) as programs can be shorter (1 – AO 2.1) and modules can be shared (1 – AO 2.1).	**4** **AO1.2** **(2)** **AO2.1** **(2)**	1 mark for each correct identification (AO1.2) up to a maximum of two identifications 1 mark for each of two valid explanations (AO2.1).

Question			Answer	Marks	Guidance
2	(a)		• Binary search (1).	**1** **AO1.1** **(1)**	For 1 mark.
	(b)		• To locate an item (1) in a list (1). The list is in some order (1).	**2** **AO1.2** **(2)**	Up to 2 marks for a valid description.
	(c)		• A (1 AO – 1.2) the list to be searched (1 – AO 2.1). • Value (1– AO 1.2) the item being searched for (1 – AO 2.1). • Low (1 AO – 1.2) the lower end of the list/sublist (1 – AO 2.1). • High (1 AO – 1.2) the upper end of the list/sublist (1 – AO 2.1).	**8** **AO1.2** **(4)** **AO2.1** **(4)**	Up to 4 marks for each correct identification (AO1.2). Up to 4 marks for each purpose identified (AO2.1).
	(d)	(i)	• The function calls itself (1) from within the function.	**2** **AO1.1** **(2)**	Up to 2 marks for a valid description.
		(ii)	• Return S(A, value, low, mid-1) (1) return S(A, value, mid+1, high) (1).	**1** **AO1.2** **(1)**	For 1 mark (either point). Accept if point in the algorithm is unambiguously referenced. There are two recursive calls in the program. Either is acceptable.
	(e)		• Divide and conquer splits a big problem into smaller parts (1). This algorithm repeatedly splits the list (1) in half (1) which reduces the amount of searching (1).	**3** **AO1.2** **(3)**	Up to 3 marks for a valid explanation.
	(f)		Individual steps in pseudocode: • Function declaration • Parameters all given correctly • Found flag used • Mid-point found • Check if value found is greater than looked for	**8** **AO2.2** **(4)** **AO3.2** **(4)**	Up to four marks for valid pseudocode (AO3.2). Up to four marks for annotated comments used (A02.2).

Question				Answer	Marks	Guidance

				value
				- Check if value found is less than mid value
				- Mid-point adjusted
				
				Programming marks to be awarded as follows:
				
				- Function declaration with parameters all given correctly (1 – AO 3.2).
				- Found flag and midpoint (1 – AO 3.2)
				- Checks if value found is greater or less than looked for value (1 – AO 3.2).
				- Mid-point adjusted within a loop (1 – AO 3.2)
				
				Possible annotated comments:
				
				- An If is used to catch an unordered list (1 – AO 2.2).
				- A WHILE loop deals with the logic that tests the binary search conditions without needing to use recursion (1 – AO 2.2) by reiterating through the conditions (1 – AO 2.2).
				- A found flag is used as the exit condition of the loop so that when the number is found the loop ends (1 – AO 2.2).
				
				Accept other valid annotations.
						Example pseudocode:
						
						`function S(A, value, low, high):`
						
						`    found = false`
						
						`    if (high < low) then`
						`        return error message`
						`    endif`
						
						`    while found == false`
						`        mid = (low+high)/2`
						`        if A[mid] > value then`
						`            high=mid-1`
						`        elseif A[mid] < value then`
						`            low=mid+1`
						`        else`
						`            found=true`
						`        endif`
						`    endwhile`
						`    return mid`
						` endfunction`

Question			Answer	Marks	Guidance
3	(a)		• Popularity data (1). • Platforms available (1). • Sales of existing similar apps (1). • Prices charged (1). • Does it exist already? (1).	**4** **AO1.2** **(4)**	1 mark for each correct identification up to a maximum of four identifications.
	(b)*		**Mark Band 3–High Level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of a wide range of considerations in relation to the statement; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate is able to weigh up both sides of the argument which results in a supported and realistic judgment as to whether it is possible to use computational methods as a way of predicting the success of an ap. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.* **Mark Band 2-Mid Level** **(4-6 marks)** The candidate demonstrates reasonable knowledge and understanding of a range of considerations in relation to the statement; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation.	**9** **AO1.1** **(2)** **AO1.2** **(2)** **AO2.1** **(2)** **AO3.3** **(3)**	**AO1: Knowledge and Understanding** The following is indicative of possible factors/evidence that candidates may refer to but is not prescriptive or exhaustive: **Indicative Content:** • Some issues can be quantified - these are easier to process computationally - simple statistics can help identify likely sales. • Examples given of quantifiable data such as sales figures. • Past performance does not guarantee future. • Could easily make errors in choosing likely quantifiable attributes. • A new app might have no relevant predecessors. • Difficult to process whether app will appeal - might depend on reviews – these are unpredictable. • App might fail for unexpected reasons/examples given. • Issues could include platform limitations/restrictions. • Marketing of apps plays an important role. • App features such as “in app purchases” and cost play an important role in popularity. **AO2.1: Application** The selected knowledge/examples should be directly related to the specific question.

Question	Answer	Marks	Guidance
	The candidate makes a reasonable attempt to come to a conclusion showing some recognition of influencing factors that would determine whether it is possible to use computational methods as a way of predicting the success of an app. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence.* **Mark Band 1-Low Level** **(1-3 marks)** The candidate demonstrates a basic knowledge of considerations with limited understanding shown; the material is basic and contains some inaccuracies. The candidate makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides nothing more than an unsupported assertion. *The information is basic and communicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear.* **0 marks** No attempt to answer the question or response is not worthy of credit.		The following is indicative of possible factors/evidence that candidates may refer to but is not prescriptive or exhaustive: • Discussions of abstraction in predicting behaviour. • Discussions of the difficulties of predicting markets, users, trends and other technical influences. • Issues surrounding validity of data gathered and how it is gathered. • Considerations regarding the components of the problems and how they are sub divided. • Discussions around how decisions are taken and the conditions that affect the flow through a program. • Discussions around concurrency and how this affects the outputs. **AO3.3: Evaluation** Having considered the different sides to the argument candidates will need to reach a supported judgment based on the evidence included in their response. There should be no bias in marks as to the degree to which the candidate agrees with the statement but especially in the top mark band there must be a clear link between the points candidates have made and justification. The following is indicative of possible factors/evidence that candidates may refer to but is not prescriptive or exhaustive: • Discussions around how the problem is solvable using computational methods • Discussions around relevant stakeholders • Discussions around solutions and essential features and any inherent limitations • An explanation/description of how the problem can be decomposed • A description of the solution potentially describing algorithms, variables, data structures or the limitations there of.

Question			Answer	Marks	Guidance
4			• Data mining or description (1) which involves searching through unconnected data (1), pattern matching (1) and calculation of correlation (1). There may be no predetermined matching criteria (1); a brute force approach is possible with high speed computers (1).	**4** **AO2.2** **(4)**	Up to 4 marks for a valid explanation. Allow for other examples.
5	(a)		• Rule of thumb/educated guess approach (1 – AO 1.1) which is used when unfeasible to analyse **all** eventualities (1 – AO 1.1). • This leads to a "good enough" result (1 – AO 1.2) although it is not 100% reliable (1 – AO 1.2).	**2** **AO1.1** **(1)** **AO1.2** **(1)**	Up to 2 marks for a valid description. 1 mark for demonstrating knowledge (AO1.1). 1 mark for demonstrating application of knowledge and understanding (AO1.2).
	(b)		• We don't gather/measure all the data (1) about vehicle speeds (1). We scan for the data most likely to help us (1) and make a judgement based on experience (1).	**3** **AO1.2** **(3)**	Up to 3 marks for a valid description.

Question			Answer	Marks	Guidance
	(c)*		**Mark Band 3–High Level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of heuristic and pattern matching approaches to virus detection; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate provides a thorough discussion which is well-balanced. Evaluative comments are consistently relevant and well-considered. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.* **Mark Band 2-Mid Level** **(4-6 marks)** The candidate demonstrates reasonable knowledge and understanding of heuristic and pattern matching approaches to virus detection; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate provides a reasonable discussion, the majority of which is focused. Evaluative comments are for the most part appropriate, although one or two opportunities for development are missed. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and*	**9** **AO1.1** **(2)** **AO1.2** **(2)** **AO2.1** **(2)** **AO3.3** **(3)**	**If only heuristic or pattern matching virus detection considered – MAX 5 marks.** **AO1: Knowledge and Understanding** **Indicative Content:** • Pattern matching requires a database of known viruses - needs to be updated regularly. • Should be successful at catching known viruses - similar viruses form a “family” should be picked up - all data about the viruses can be considered. • New viruses may be completely unlike existing ones - reliance on analytics may miss a totally new mechanism. • Heuristic looks at behaviour rather than structure - can uncover suspicious activity even if produced in a novel way. • Heuristic methods examine the susceptibility of the system to possible attacks. • Heuristic methods simulate the possible effects of a suspected virus. • Heuristic methods sometimes decompile the suspicious program, then analyse the resulting source code. **AO2.1: Application** The selected knowledge/examples should be directly related to the specific question. The following is indicative of possible factors/evidence that candidates may refer to but is not prescriptive or exhaustive: • Discussion on different types of virus checkers • Issues around virus attack and protection • Discussion on heuristic and pattern matching approaches to virus detection

Question	Answer	Marks	Guidance
	supported by some evidence. **Mark Band 1-Low Level** **(1-3 marks)** The candidate demonstrates a basic knowledge of approaches to virus detection with limited understanding shown; the material is basic and contains some inaccuracies. The candidate makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides a limited discussion which is narrow in focus. Judgments if made are weak and unsubstantiated. *The information is basic and communicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear.* **0 marks** No attempt to answer the question or response is not worthy of credit.		• Discussion on anti-virus proactive scan for unknown viruses **AO3.3: Evaluation** Candidates will need to consider a variety of strategies taken in relation to virus detection and will make some evaluative comments about the issues and solutions they are discussing. The following is indicative of possible factors/evidence that candidates may refer to but is not prescriptive or exhaustive: • A comparison of the different types of pattern matching methods that can be used to identify viruses. • How effective are the approaches of virus detection using heuristic and different pattern matching methods? • The differences between the heuristic and pattern matching methods, explaining which one is better? • Likelihood of 'false positives' occurring for the different methods? How this problem could be solved? • Consideration of how fuzzing can affect the results of each method. • How updates for antivirus software alter the effectiveness of the methods. • Discussion of other factors that alter the effectiveness of antivirus software.

Question			Answer	Marks	Guidance
6	(a)		• Time to complete transaction (1). • The number of customers (1). • The distance travelled by one/each customer (1). • The number of collection actions per customer (1). • The number of changes of direction of customer (1).	**3** **AO2.2** **(3)**	1 mark for each correct identification up to a maximum of three identifications.
	(b)		• Computational methods could be used to break the problem down into sections (1) e.g. distances (1), times (1), number of stops (1). Using computational methods, statistics can be compiled (1), models of new situations produced (1), simulations run by computer (1) variables used to represent data items (1) and algorithms devised to test possible layouts under different circumstances (1).	**4** **AO2.2** **(4)**	Up to 4 marks for a valid explanation.
7	(a)		• Data/processes arranged in a series output of one is input of next.	**2** **AO1.1** **(2)**	Up to 2 marks for a valid description.
	(b)		• Look for processes that can be processed at the same time (1) processes that must be sequential (1). Example of possible parallel processes, e.g. windows installed at same time as electrics/plumbing (1). Example of an obligatory sequence e.g. rafters installed before tiles put on/walls before roof (1).	**4** **AO2.1** **(4)**	Up to 4 marks for a valid explanation. Allow any correct examples for the last 2 mark points.
	(c)		• Instruction processing (1) - some processors allow parts of instructions to be processed (1) without waiting to complete the whole instruction cycle (1). • Pipes to pass data between programs (1) from programs to peripherals/to programs from peripherals (1), example such as | symbol in Unix, or • Popen() or pipe() in C (1).	**4** **AO2.1** **(4)**	1 mark for each correct identification up to a maximum of two identifications plus up to a further 1 mark for each of two valid descriptions.

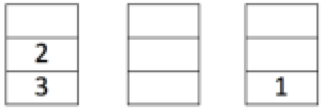
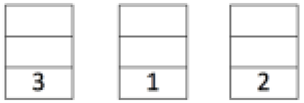
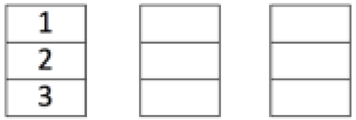
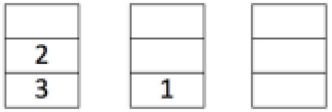
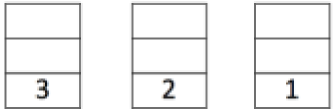
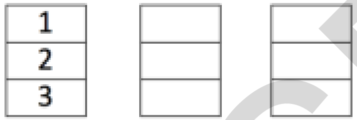
Question			Answer	Marks	Guidance
			• Graphics pipelines (1) separate processor renders graphics from data supplied by other processes (1), parts (vertices) of the image are pipelined at the same time as custom software (shader) that renders the display (1).		

8			Individual steps in pseudocode:
			• Input text and key
			• Get a length of text
			• Method to convert text to binary such as
			• ASCII/lookup table
			• Convert key to binary or accept key as binary
			• Use of loop to iterate through message duplicate key element
			• Perform XOR operation
			• Convert back to text
			Programming marks to be awarded as follows
			• pseudocode shows the input of text and the length being established (1-AO3.2)
			• pseudocode shows a method to convert text and key to binary and binary to text with the need for the key to be duplicated (1-AO3.2)
			• pseudocode shows a loop iterating through the message (1-AO3.2)
			• pseudocode shows the XOR operation being correctly applied (1-AO3.2)
			Possible annotated comments:
			• The pseudocode will prompt for an input for a message and key and store it as a variable. (1 – AO 2.2)
				8
				AO2.2
				(4)
				AO3.2
				(4)
					Up to 4 marks for valid pseudocode (AO3.2).
					Up to 4 marks for annotated comments used (AO2.2).
					Example pseudocode:
					`msg=(message)`
					`key=int(key)`
					
					`msglength=len(msg)`
					
					`keystring=str(bin(key))`
					`keystring_length=len(keystring)`
					
					`for i=1 to msglength`
					`asc_msg=asc_msg+asc(msg(i,1))`
					`end for`
					
					`asc_msg_length=len(asc_msg)`
					`for i=1 to asc_msg_length/8`
					`complete_key=complete_key+keystring`
					`end for`
					
					`for i=1 to len(asc_msg)`
					`encryptmsg=encryptmsg+complete_key(1,i) XOR asc_msg(1,i)`
					`end for`

<table>
<tr><th colspan="3">Question</th><th>Answer</th><th>Marks</th><th>Guidance</th></tr>
<tr><td></td><td></td><td></td><td>• The length of the message variable is checked and then converted to binary (1 – AO 2.2).
• The key is then converted to binary if needed (1 – AO 2.2).
• The loop then duplicates the key and performs a XOR operation, the output of this is then converted back to text which is now encrypted (1 – AO 2.2).</td><td></td><td></td></tr>
<tr><td><b>9</b></td><td><b>(a)</b></td><td></td><td><table><tr><th></th><th>Worst Case</th><th>Average Case</th><th>Best Case</th></tr><tr><td>Binary Search</td><td>log2(n)</td><td></td><td>1</td></tr><tr><td>Linear Search</td><td>n</td><td></td><td>1</td></tr></table></td><td><b>4
AO2.1
(4)</b></td><td>For 4 marks – 1 mark for each correct entry.</td></tr>
<tr><td></td><td><b>(b)</b></td><td></td><td>• As x (or the size of the array) increases, the rate at which y (or the number of checks needed) increases more slowly (1).
• The inverse of exponential growth (1).
• The rate of increase is a logarithmic function of the size of the array (1).</td><td><b>1
AO2.1
(1)</b></td><td>For 1 mark.</td></tr>
<tr><td></td><td><b>(c)</b></td><td></td><td>• If the array is very small. (1)
• If the item being searched for is very close to the start of the array. (1)</td><td><b>1
AO2.1
(1)</b></td><td>For 1 mark.</td></tr>
<tr><td><b>10</b></td><td><b>(a)</b></td><td><b>(i)</b></td><td>• A disk can only be put onto the top of the pole (1) and a stack is a last in first out structure (1) whereas a queue is first in first out (1).</td><td><b>2
AO2.1
(2)</b></td><td>Up to 2 marks for a valid explanation.</td></tr>
</table>

Question			Answer	Marks	Guidance

		(ii)	Individual steps in pseudocode:
			• Checks if pointer is 0 (i.e. pointer is 0)
			• Checks if the disk being added is smaller than the one at the top of the pole
			• Puts diskValue at the top of stack
			• Move the pointer up one
			• If it’s not a valid move, no disk is added
			• …and prints Invalid move
			
			Programming marks to be awarded as follows:
			• Checks if pointer is 0 and checks if the disk being added is smaller than the one at the top of the pole (1 – AO 3.2).
			• Puts diskValue at the top of stack and moves the pointer up one (1 – AO 3.2).
			• If it’s not a valid move, no disk is added and prints Invalid move (1 – AO 3.2).
			
			Possible annotated comments:
			• The IF statement checks the pointer value is 0 so we can assume the disk is available to be moved (1 – AO 2.2).
			• If these conditions are met then pole value is set to the diskValue and incremented by 1 so the disk can move to the next pole (1 – AO 2.2).
			• If these conditions are not met then the else prints Invalid move (1 – AO 2.2).
				6
				AO2.2
				(3)
				AO3.2
				(3)
					Up to 3 marks for valid pseudocode (AO3.2).
					
					Up to 3 marks for annotated comments used (AO2.2).
					
					Example pseudocode:
					
					`If pointer==0 or pole[pointer]>diskValue then`
					`    pole[pointer]=diskValue`
					`    pointer=pointer+1`
					`else`
					`    print(“Invalid move”)`
					`endif`
					
					Example of pseudocode with comments in code for guidance:
					
					`If pointer==0 or`
					`pole[pointer]>diskValue then`
					
					`    pole[pointer]=diskValue`
					`    pointer=pointer+1`
					
					`else`
					`    print(“Invalid move”)`
					`endif`

Question			Answer	Marks	Guidance
	(b)	**(i)**	Left hand boxes, any 2 of:  or  or  (1 Mark each, Max 2) Right hand boxes, any 2 of:  or  or  (1 Mark each, Max 2)	**4** **AO2.2** **(4)**	For 4 marks as indicated.

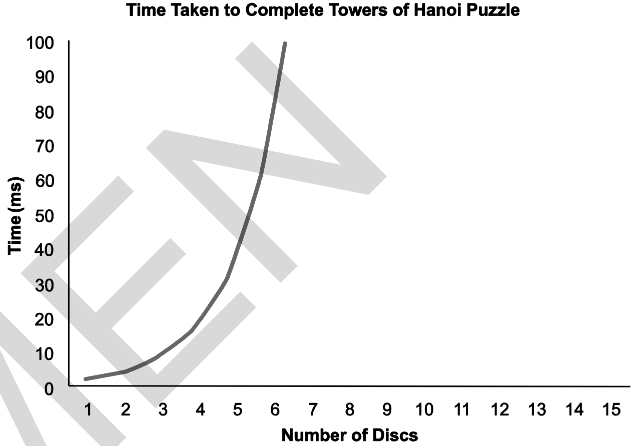
Question			Answer	Marks	Guidance
		(ii)	**Depth first advantage:** • Depth first requires less memory than breadth first search (1). It is quicker if you are looking at deep parts of the tree (1). **Depth first disadvantage:** • Depth first isn't guaranteed to find the quickest solution (1) and possibly may never find the solution (1) if we don't take precautions not to revisit previously visited states (1).	**4 AO1.2 (4)**	Up to 4 marks for a valid description.

	(c)		Individual steps in pseudocode:
			Makes a swap between 1 and 3
			…..and always makes a valid swap.
			Then makes a swap between 1 and 2.
			…. And always makes a valid swap
			….only if the problem is not solved.
			Repeats if not solved.
			Will not attempt to make a swap between 2 empty towers.
			Programming marks to be awarded as follows:
			• Makes a swap between pole 1 and 3 (1 – AO 3.2).
			• Checks for a valid move throughout (1 – AO 3.2).
			• Makes a swap between pole 1 and 2 (1 – AO 3.2).
			• Checks whether problem is solved throughout (1 – AO 3.2).
			• Makes a swap between 3 and 2 (1 – AO 3.2).
			• Repeats if not solved and will not attempt a swap between two empty towers. (1 – AO 3.2).
			•
				**10
				AO2.2
				(4)
				AO3.2
				(6)**
					Up to 6 marks for valid pseudocode (AO3.2)
					Up to 4 marks for annotated comments used (AO2.2).
					Example pseudocode:
					`while tower1.peek()!=999 or`
					`tower2.peek()!=999`
					`    if tower1.peek()<tower3.peek() then`
					`         disk=tower1.pop()`
					`         tower3.push(disk)`
					`    elseif tower3.peek()<tower1.peek() then`
					`         disk=tower3.pop()`
					`         tower1.push(disk)`
					`    endif`
					
					`if  tower1.peek()!=999 or tower2.peek()!=999`
					`then`
					`         if tower1.peek()<tower2.peek() then`
					`             disk=tower1.pop()`
					`             tower2.push(disk)`
					`         elseif tower2.peek()<tower1.peek()`
					`then`

Question			Answer	Marks	Guidance

			Possible annotated comments:
			• The loop checks if towers 1 or 2 have anything on them. If they do the problem is not solved as all the disks need to be on pole 3. (1 – AO 2.2).
			• The first selection (if) makes a swap between 1 and 2 and always makes a valid swap only if the problem is not solved. (1 – AO 2.2).
			• The second selection (if) makes a swap between 3 and 2 and always makes a valid swap only if the problem is not solved. (1 – AO 2.2).
			• A swap is only valid if the pole is either empty or the disk on which it is to be put is a larger number (1 – AO 2.2).
					`disk=tower2.pop()`
					`tower1.push(disk)`
					`endif`
					`endif`
					`if  tower1.peek()!=999 or tower2.peek()!=999 then`
					`if tower3.peek()<tower2.peek() then`
					`disk=tower3.pop()`
					`tower2.push(disk)`
					`elseif tower2.peek()<tower3.peek() then`
					`disk=tower2.pop()`
					`tower3.push(disk)`
					`endif`
					`endif`
					`endwhile`

	(d)	**(i)**	32 ms (1)	**1** **AO2.1** **(1)**	For one mark.
		(ii)	Big O notation shows the limiting behaviour of an algorithm (to classify its complexity) (1). • Other processes may be taking up some of the processor time (1).	**1** **AO1.2** **(1)**	For one mark.
		(iii)	Exponential curve drawn (1). Curve passes 100 on the y axis before x reaches 8 (1).	**2** **AO2.1** **(2)**	For two marks.

Question			Answer	Marks	Guidance
					
	(e)		**Mark Band 3–High Level** **(7-9 marks)** The candidate demonstrates thorough knowledge and understanding of concurrency including considerations of the search space tree and iterative algorithm approaches; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate provides a thorough discussion which is well-balanced. Evaluative comments are consistently relevant and well-considered. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.*	**9** **AO1.1** **(2)** **AO1.2** **(2)** **AO2.1** **(2)** **AO3.3** **(3)**	**If only search space tree or iterative algorithm approaches considered – MAX 5 marks.** **AO1: Knowledge and Understanding** **Indicative Content:** The following is indicative of possible factors/evidence that candidates may refer to but is not prescriptive or exhaustive: • Concurrency could be used with the search tree. • Different processors could generate the subtrees of different nodes until the correct answer is found. • With the iterative algorithm each stage depends on the previous one meaning it is not possible to use concurrent processing to speed up the process. • Because of the exponential complexity as n increases concurrent processing becomes less practical.

Question	Answer	Marks	Guidance
	Mark Band 2-Mid Level **(4-6 marks)** The candidate demonstrates reasonable knowledge and understanding of concurrency including considerations of the search space tree and iterative algorithm approaches; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate provides a reasonable discussion, the majority of which is focused. Evaluative comments are for the most part appropriate, although one or two opportunities for development are missed. *There is a line of reasoning presented with some structure The information presented is in the most part relevant and supported by some evidence.* **Mark Band 1-Low Level** **(1-3 marks)** The candidate demonstrates a basic knowledge of concurrency with limited understanding shown; the material is basic and contains some inaccuracies. The candidate makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides a limited discussion which is narrow in focus. Judgments if made are weak and unsubstantiated. *The information is basic and communicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be*		**AO2.1: Application** The selected knowledge/examples should be directly related to the specific question. The following is indicative of possible factors/evidence that candidates may refer to but is not prescriptive or exhaustive: • Discussions around abstraction and modelling. • Discussions around the preconditions and how these will be applied. • Discussions around identifying the components of the problem and how concurrency might be used. • Discussions around order of steps and sub processes. • Discussion around the flow of the program and the decisions taken and how this effects the use of concurrency. **AO3.3: Evaluation** Candidates will need to consider a variety of factors in relation to the question and will make some evaluative comments about the issues and solutions they are discussing. The following is indicative of possible factors/evidence that candidates may refer to but is not prescriptive or exhaustive: • Discussions of the features and how the different approaches are more or less suitable for a solution. • Discussions of the essential features of the solutions and the differences between them with discussion of the limitations.

Question	Answer	Marks	Guidance
	clear. **0 marks** No attempt to answer the question or response is not worthy of credit.		

Assessment Objectives (AO) Grid

Question	Assessment Objectives							Total
	AO1.1	AO1.2	AO2.1	AO2.2	AO3.1	AO3.2	AO3.3	
1a	2	0	2	0	0	0	0	**4**
1bi	0	3	0	0	0	0	0	**3**
1bii	0	3	0	0	0	0	0	**3**
1c	0	2	2	0	0	0	0	**4**
2a	1	0	0	0	0	0	0	**1**
2b	0	2	0	0	0	0	0	**2**
2c	0	4	4	0	0	0	0	**8**
2di	2	0	0	0	0	0	0	**2**
2dii	0	1	0	0	0	0	0	**1**
2e	0	3	0	0	0	0	0	**3**
2f	0	0	0	4	0	4	0	**8**
3a	0	4	0	0	0	0	0	**4**
3b*	2	2	2	0	0	0	3	**9**
4	0	0	0	4	0	0	0	**4**
5a	1	1	0	0	0	0	0	**2**
5b	0	3	0	0	0	0	0	**3**
5c*	2	2	2	0	0	0	3	**9**
6a	0	0	0	3	0	0	0	**3**
6b	0	0	0	4	0	0	0	**4**
7a	2	0	0	0	0	0	0	**2**
7b	0	0	4	0	0	0	0	**4**
7c	0	0	4	0	0	0	0	**4**
8*m*	0	0	0	4	0	4	0	**8**
9a*m*	0	0	4	0	0	0	0	**4**
9b*m*	0	0	1	0	0	0	0	**1**
9c*m*	0	0	1	0	0	0	0	**1**
10ai	0	0	2	0	0	0	0	**2**
10aii	0	0	0	3	0	3	0	**6**
10bi	0	0	0	4	0	0	0	**4**
10bii	0	4	0	0	0	0	0	**4**
10c	0	0	0	4	0	6	0	**10**
10di*m*	0	0	1	0	0	0	0	**1**
10dii*m*	0	1	0	0	0	0	0	**1**
10diii*m*	0	0	2	0	0	0	0	**2**
10e*	2	2	2	0	0	0	3	**9**
Totals	**14**	**37**	**33**	**30**	**0**	**17**	**9**	**140**

*** = extended response**

***m* = mathematical content**

BLANK PAGE

BLANK PAGE