GCE AS/A LEVEL

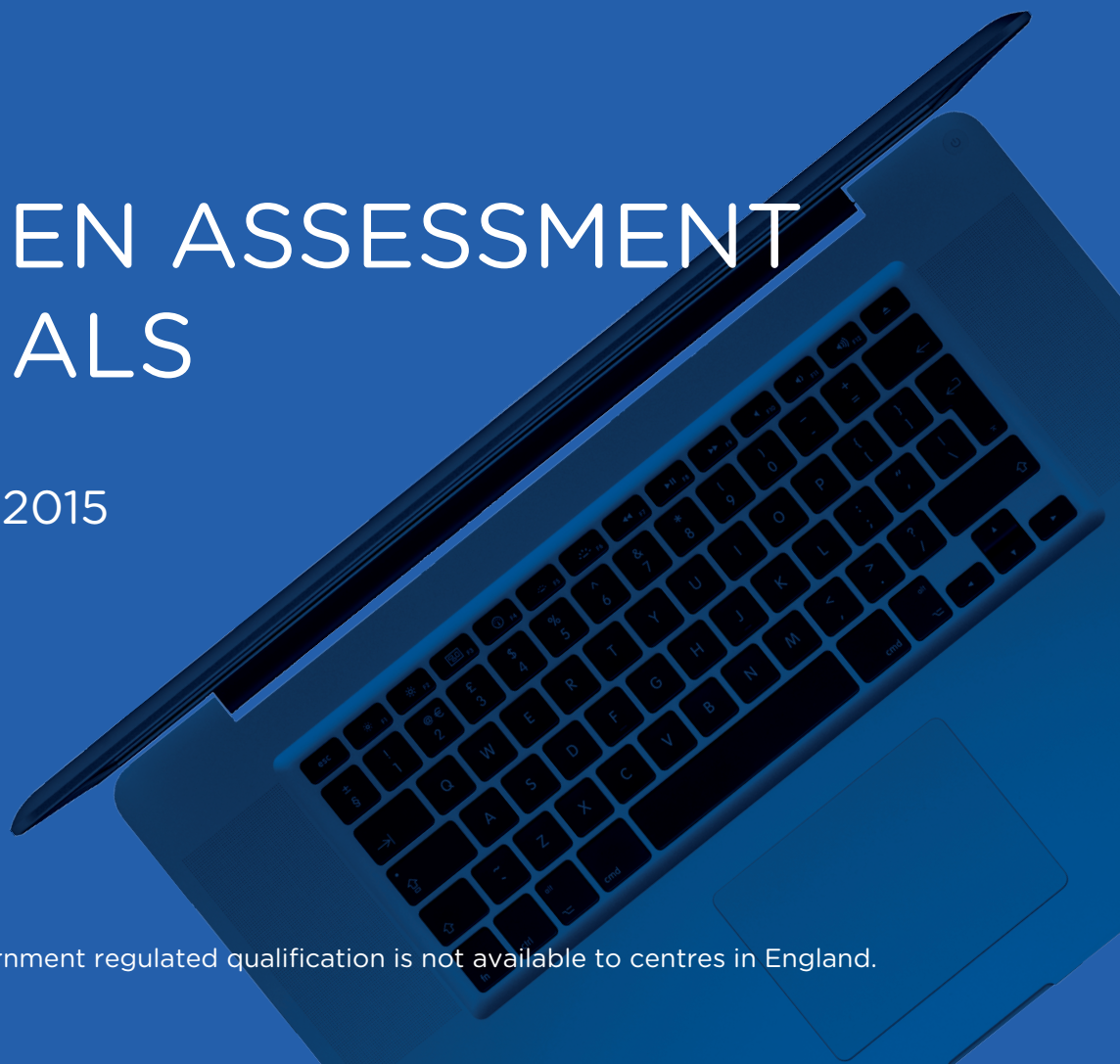
WJEC GCE AS/A LEVEL in

COMPUTER SCIENCE

ACCREDITED BY WELSH GOVERNMENT

SPECIMEN ASSESSMENT MATERIALS

Teaching from 2015



This Welsh Government regulated qualification is not available to centres in England.

For teaching from 2015

GCE AS and A LEVEL COMPUTER SCIENCE

SPECIMEN ASSESSMENT MATERIALS

Contents

Candidate Name	Centre Number					Candidate Number				
						0				

AS COMPUTER SCIENCE

UNIT 1

FUNDAMENTALS OF COMPUTER SCIENCE

SPECIMEN PAPER

2 hours

ADDITIONAL MATERIALS

The use of a calculator is permitted in this examination.

INSTRUCTIONS TO CANDIDATES

Answer **ALL** question(s).

Write your name, centre number and candidate number in the spaces at the top of this page.

Write your answers in the spaces provided in this booklet.

INFORMATION FOR CANDIDATES

The number of marks is given in brackets at the end of each question or part-question.

The total number of marks available is 100.

Assessment will take into account the quality of written communication used in your answers.

No certificate will be awarded to a candidate detected in any unfair practice during the examination.

Unit 1
*Answer **all** questions*

1. Discuss the differences between RAM and Cache memory. [6]

2. Describe the functional characteristics of two contemporary secondary storage devices. [6]

3. (a) Describe the use of the DHCP and HTTP protocols. [2]

……
……
……
……
……
……
……
……

(b) Explain why the FTP protocol is unsuitable for streaming a live video feed. Suggest a more suitable protocol for this purpose, justifying your choice. [6]

……
……
……
……
……
……
……
……
……
……
……
……

(c) Giving an example, explain the role of handshaking. [2]

……
……
……
……

4. The 8 bit binary number 00000001_2 is used in a masking process.

   (a) State what logical operation is used during masking and draw a truth table for this logical operation. [2]

   ……
   ……
   ……
   ……

   (b) State the effect that the 8 bit binary number given in (4.) would have when masked with any 8 bit binary number. [1]

   ……
   ……

5. Explain the terms serial transmission and parallel transmission in a computer system and give one advantage of each type of transmission. [4]

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

6. Related data stored on a computer system can become fragmented over time.

(a) Explain what is meant by the term fragmentation, give one possible effect and explain defragmentation. [4]

……
……
……
……
……
……
……
……

(b) Discuss the issues surrounding the defragmentation of a Solid State Drive (SSD). [3]

……
……
……
……
……
……

7. Explain file backup, generations of files and transaction logs. [6]

……
……
……
……
……
……
……
……
……
……
……

8. (a) Using binary addition, calculate the number that would result from adding 00110110_2 and 00101110_2.

Convert the result into hexadecimal. [2]

……
……
……
……
……
……

(b) Assuming that 1 is used to indicate a negative number, show how the negative number -12_{10} will be represented using sign/magnitude in an 8 bit register. [1]

……
……

(c) Integers can also be represented using two's complementation.

Using the binary number 00001011_2 as an example, explain how two's complement is derived. [3]

……

……

……

……

……

……

(d) (i) Give a disadvantage of using floating point form rather than integer form. [1]

……

……

(ii) Real numbers stored in floating point form can be stored using 16 bits as shown below:

Mantissa (12 bits in two's complement form. The binary point in the mantissa is immediately after the left bit.)	**Exponent** (4 bits in two's complement form.)

Convert the number 23.75_{10} into this floating point form. [2]

……

……

……

……

(iii) In a different computer system, the following is a floating point representation of a number, using an 8 bit mantissa and a 4 bit exponent:

$0 \bullet 1011000_2 \ 0101_2$

Calculate the mantissa, exponent and decimal equivalent of the number. [3]

……

……

……

……

……

9. (a) Complete the following Boolean expression to illustrate the distributive law. [1]

$$A . (B + C)$$

……

……

(b) Clearly showing each step, simplify the following Boolean expression. [5]

$$A.B + A.(B + C) + B.(B + C)$$

……

……

……

……

……

……

……

……

……

……

10. Below is an algorithm that determines whether a positive whole number greater than 2, input by a user, is a prime number or not.

```
1  Num is Integer
2  Divisor is Integer
3  Prime is Boolean
4  startmainprog
5     set Prime = TRUE
6     set Divisor = 2
7     output "type in a number"
8     input Num
9
10    repeat
11          if Num MOD Divisor = 0 then
12             set Prime = FALSE
13          endif
14          set Divisor = Divisor + 1
15    until (Prime = FALSE) OR (Divisor = Num)
16
17    if Prime = TRUE then
18          output Num, "is a prime number"
19    else
20          output Num, "is NOT a prime number"
21    Endif
22
23 Endmainprog
```

(a) Name the logical operator used in the algorithm: [1]

……
……

(b) Give an example of selection from the algorithm above and explain its purpose. [2]

……
……
……
……

(c) Give an example of repetition from the algorithm above and explain its purpose. [2]

……
……
……
……

(d) Give two examples of test data that would test that the algorithm on page 13 works as intended. [2]

……

……

……

……

11. Customers can obtain quotations for car insurance via email by completing an on-line application form. Verification and validation checks are carried out on the data input.

(a) The customer has to create an account so that the insurance quotation can be stored and retrieved at a later date. As part of this process, the customer has to input a password which is verified.

Describe one method of verification that could be applied to the password.

[1]

……

……

(b) The number of whole years since the driver made an insurance claim is validated. Excluding a presence check, describe a suitable validation check that would be carried out on the number of whole years giving an example of invalid data that would be detected by this check. [2]

……

……

……

……

(c) Describe a different suitable validation check that could be carried out on the customer's email address giving an example of invalid data that would be detected by this check. [2]

……

……

……

……

12. A weather station records monthly rainfall figures in millimetres (mm) for a year.

Write an algorithm, using pseudo-code or a high level programming language, which will use these twelve monthly rainfall figures as input. The program should output:

- the total rainfall for the year
- the mean monthly rainfall for the year
- the month numbers (1 for January, etc) where the rainfall was above the mean.

Your algorithm should contain meaningful identifiers. [6]

13. (a) Before new computer systems are introduced in an organisation, they are tested to ensure they meet the intended requirements. Describe the different types of system testing that will typically be carried out on the system. [3]

……

……

……

……

……

……

……

(b) Following the installation of the new system, it will need to be maintained. Giving suitable examples, discuss the different types of system maintenance that will need to be carried out on the system. [3]

……

……

……

……

……

……

14. Identify and describe the principal stages involved in the compilation process. [8]

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

15. Explain lossy and lossless data compression techniques. [8]

Continuation page

Continuation page

Candidate Name	Centre Number					Candidate Number				

AS COMPUTER SCIENCE

UNIT 2

Practical Programming to Solve Problems

SPECIMEN PAPER

2 hours

ADDITIONAL MATERIALS
In addition to this examination paper, you will need a 16 page answer book.

INSTRUCTIONS TO CANDIDATES

Answer **ALL** questions.
The question paper is divided into four main sections. Below are the recommended timings for this assessment.

Section A
You are advised to spend no more than **60 minutes** on this section.
You will be required to analyse and deconstruct the scenario so as to consider its component parts in terms that can be addressed through automated computation.

Section B
You are advised to spend no more than **60 minutes** on this section.
Questions will require you to develop programs that solve computing problems referring to the initial scenario.

INFORMATION FOR CANDIDATES

The number of marks is given in brackets at the end of each question or part-question.

The total number of marks available is 60.

Assessment will take into account the quality of written communication used in your answers.

You will need a computer with an installed functional copy of the Integrated Development Environment (IDE) appropriate to your chosen programming language and word processing software.

Remember to save your work regularly.

No certificate will be awarded to a candidate detected in any unfair practice during the examination.

Scenario

FITNESS
LEISURE CENTRE

Fitness Leisure Centre is an established organisation based in Trefforest, South Wales. The leisure centre has a large number of clients that take part in various activities on a regular basis.

The leisure centre wishes to offer its clients the facility of being able to track their activities using a computerised training log.

This new system will store the clients' details along with details of the different activities that are available, such as swimming, gym, aerobics, badminton, squash. When a client takes part in one of the activities they will be able to keep a record of this in their training log on the new system. The details stored in the training log include Activity ID, Client ID, Date, Time, Duration, Number of calories burnt, etc.

The main requirements of the new computer system for ***Fitness Leisure Centre*** are:

- The ability to store and search clients details
- The ability to store activity details and search for activities using various preferences, e.g. search for all "swimming classes"
- The ability to store a training log for each client

Fitness Leisure Centre has the following partially completed entity-relationship diagram as an overview of the system they would like to create:

Section A: Analysis and Design (44 marks)
Answer all questions.

You have been asked to analyse the scenario as a preliminary step towards creating a computer system for *Fitness Leisure Centre.*

Answer all questions. Present your answers as a single word-processed document named *Section A Analysis and Design*.

1. Draw two data structure tables that will allow ***Fitness Leisure Centre*** to store client and activity details. [10]

2. Copy and complete the entity relationship diagram provided in the scenario. [6]

3. Construct a flow chart for the process of searching for a specific activity, e.g. Swimming classes. [6]

4. Select and fully justify your proposed method of solution for the three main requirements listed in the scenario. [6]

5. Consider the different methods of changeover available to ***Fitness Leisure Centre.*** Your answer should be related to the impact of these methods on clients and staff. [8]

6. ***Fitness Leisure Centre*** is concerned that their clients will be shown duplicate results on screen when searching for activities.

   Using a recognised convention, design and implement an algorithm that performs a search for duplicate consecutive activities in an array. If a duplicate is found, the algorithm should output the location of the duplicate and the word “TRUE”. If a duplicate is not found, the algorithm should output the word “FALSE”. [8]

Section B: Develop programs (16 marks)

Select the programming language of your choice from section BI, BII or BIII and answer all 3 questions within your chosen section.

BI Visual Basic

***Fitness Leisure Centre* wants a computer system to be developed using Visual Basic that meets the requirements outlined in the scenario.**

1. Open the file Activities.frm
   - Read through the code and familiarise yourself with its contents
   - The file contains incomplete code that displays the activities available to clients.

   Complete this code.

   Remember to save the changes made to the file Activities.frm [4]

2. Create a new form that will allow ***Fitness Leisure Centre*** to:

   - Input customer details
   - Validate customer details
   - Store customer details on disc in a text file called customerdetails.txt
   - Retrieve specified customer details from disc.

   Save your new form as Customers.frm [8]

3. Using the internal facility of your chosen language, add annotated listings to your code from question 2 that would clearly explain the design of your program to another software developer.

   Remember to save the changes made to the file Customers.frm [4]

BII Java

***Fitness Leisure Centre* wants a computer system to be developed using Java that meets the requirements outlined in the scenario.**

1. Open the file Activities.java
   - Read through the code and familiarise yourself with its contents
   - The file contains incomplete code that displays the activities available to clients.

   Complete this code.

   Remember to save the changes made to the file Activities.java [4]

2. Create a new form that will allow ***Fitness Leisure Centre*** to:

   - Input customer details
   - Validate customer details
   - Store customer details on disc in a text file called customerdetails.txt
   - Retrieve specified customer details from disc.

   Save your new form as Customers.java [8]

3. Using the internal facility of your chosen language, add annotated listings to your code from question 2 that would clearly explain the design of your program to another software developer.

   Remember to save the changes made to the file Customers.java [4]

BIII Python

***Fitness Leisure Centre* wants a computer system to be developed using Python that meets the requirements outlined in the scenario.**

1. Open the file Activities.py
   - Read through the code and familiarise yourself with its contents
   - The file contains incomplete code that displays the activities available to clients.

   Complete this code.

   Remember to save the changes made to the file Activities.py [4]

2. Create a new form that will allow ***Fitness Leisure Centre*** to:

   - Input customer details
   - Validate customer details
   - Store customer details on disc in a text file called customerdetails.txt
   - Retrieve specified customer details from disc.

   Save your new form as Customers.py [8]

3. Using the internal facility of your chosen language, add annotated listings to your code from question 2 that would clearly explain the design of your program to another software developer.

   Remember to save the changes made to the file Customers.py [4]

Candidate Name	Centre Number					Candidate Number				
						0				

A LEVEL COMPUTER SCIENCE

UNIT 3

PROGRAMMING AND SYSTEM DEVELOPMENT

SPECIMEN PAPER

2 hours

ADDITIONAL MATERIALS

In addition to this examination paper, you will need a 16 page answer book..

The use of a calculator is permitted in this examination.

INSTRUCTIONS TO CANDIDATES

Answer **ALL** question(s).
Write your answers in the separate answer book provided.

Write your name, centre number and candidate number in the spaces at the top of this page.

INFORMATION FOR CANDIDATES

The number of marks is given in brackets at the end of each question or part-question.

The total number of marks available is 100.

Assessment will take into account the quality of written communication used in your answers.

No certificate will be awarded to a candidate detected in any unfair practice during the examination.

Unit 3

*Answer **all** questions*

1. A binary tree can be constructed using the following rules:

Rule 1. The first item becomes the root node;
Rule 2. Items earlier or at the same position in the alphabet follow the left pointer;
Rule 3. Items later in the alphabet follow the right pointer.

(a) Draw a representation of a dynamic binary tree with pointers using the following data:

Narberth, Cardiff, Pontypridd, Wrexham, Rhyl, Bangor, Denbigh [2]

(b) Show how the above tree could be represented using a two dimensional array. [3]

(c) Make the following amendments to the original tree:

(i) Insert Pontardawe and Tonypandy into the tree; [2]
(ii) Delete Cardiff and Pontypridd from the tree. [3]

(d) Giving examples, compare a balanced and an un-balanced binary tree and evaluate their effectiveness to solve problems by comparing the maximum number of comparisons to locate an item in each of these trees. [4]

2. (a) Explain what is meant by the term linked list. Describe one benefit and one drawback of using a linked list compared with using an array. [4]

(b) In a certain implementation, a linked list of integers is actually stored in a table form as shown below. The integers are to be accessed in ascending numerical order. A variable points to the address 752, which contains the lowest integer, 2312.

Copy and complete the pointer column in the table below.

Address	Integer	Pointer
751	4811	
752	2312	
753	3599	
754	4166	
755	2567	
756	5218	
757	3100	

[3]

3. In the programming industry, professional Codes of Conduct are used extensively and they set rules that should be followed by programmers.

   State **six** reasons why programmers are required to follow rules under a professional Code of Conduct. Each reason must relate to a specific rule. [6]

4. (a) Using the laws of Boolean algebra, simplify the following Boolean expression. [3]

$$(A + B).(A + \overline{B})$$

   (b) Using the laws of Boolean algebra and De Morgan's theorem, simplify the following expression. [3]

$$C + \overline{BC}$$

   (c) Simplify the following Boolean expression. [3]

$$A + (A + \overline{B.C}) + \overline{C}$$

5. Two 8 bit numbers are:

X 10011011_2
Y 11010111_2

   Explain how the XOR operation can be used for encrypting data, using the two numbers above to illustrate your answer. You should also show how the original data can be recovered. [3]

6. Write an algorithm that performs a binary search for SearchValue in an array of size `n` called SearchArray. Your algorithm should output the position of the SearchValue if found or an error message if the SearchValue is not found. [7]

7. Name the the type of algorithm shown below. Give two features of this type of algorithm. Name a sort which uses this type of algorithm. [4]

```
function CalcF (Num: integer) :
integer
if Num = 1
then CalcF = 1
else CalcF = Num * CalcF(Num-1)
```

8. (a) Describe the purpose of Backus-Naur Form (BNF). [1]

(b) A warehouse stores a large number of mobile phone components. Each component has a component code consisting of 6 digits and a description. The description is made up of upper case letters only and may be of any length.

Examples of valid codes are:

621402DIGITISER
506742USBPORT

(i) Produce an appropriate BNF definition for a component code. [4]
(ii) Produce an appropriate syntax diagram for a component code. [3]

9. Below is an algorithm that sorts data stored in an array.

```
Algorithm sort

Declare A(1 to n)

n = length(A)

for i = 1 to n
      for j = 1 to  n-1 inclusive do

            if A[i-1] > A[i] then
                  swap( A[i-1], A[i] )
            end if

     next j
next i
```

(a) Determine how many comparisons are made when `n` is equal to:

10
100
1000 [1]

(b) Evaluate the efficiency of the algorithm and, using Big O notation, determine the growth rate for the time performance. Your answer should refer to the comparisons performed by the algorithm. [4]

(c) Using Big O notation, determine the growth rate of memory space used by this algorithm. [2]

10. Describe the Waterfall and Agile approaches to systems analysis including their advantages and disadvantages. [6]

11. A software development company specialises in developing computerised automatic car braking systems.

    (a) When the company develops such a system, a design validation is carried out. Describe how a design validation should be carried out. [3]

    (b) Explain why version control is important when developing computer programs. [1]

12. (a) Describe the purpose of compilers, interpreters and assemblers. [3]

    (b) Describe and give examples of three common translation and execution errors. [9]

13. Discuss the nature of different programming paradigms. [13]

Candidate Name	Centre Number					Candidate Number				
						0				

A LEVEL COMPUTER SCIENCE

UNIT 4

COMPUTER ARCHITECTURE, DATA, COMMUNICATION AND APPLICATIONS

SPECIMEN PAPER

2 hours

ADDITIONAL MATERIALS

In addition to this examination paper, you will need a 16 page answer book.

The use of a calculator is permitted in this examination.

INSTRUCTIONS TO CANDIDATES

Answer **ALL** question(s).

Write your answers in the separate answer book provided.

Write your name, centre number and candidate number in the spaces at the top of this page.

INFORMATION FOR CANDIDATES

The number of marks is given in brackets at the end of each question or part-question.

The total number of marks available is 100.

Assessment will take into account the quality of written communication used in your answers.

No certificate will be awarded to a candidate detected in any unfair practice during the examination.

Unit 4
Answer all questions

1. (a) Explain the sequence of operations which will occur during the fetch phase of the fetch execute cycle, making clear the role of the PC (program counter), the MAR (memory address register) and the MDR (memory data register). [3]

(b) In a certain computer, the following assembly language commands are available.

Assembly command	Description
STR R,X	Stores contents of register R in address X
DEC R	Decrement register R
CLR R	Clears the contents of register R
LSL R	Performs a logical shift left o register R
LOD R,X	Loads register R with the contents of address X
ADD R, S	Adds the contents of register R to the contents of register S

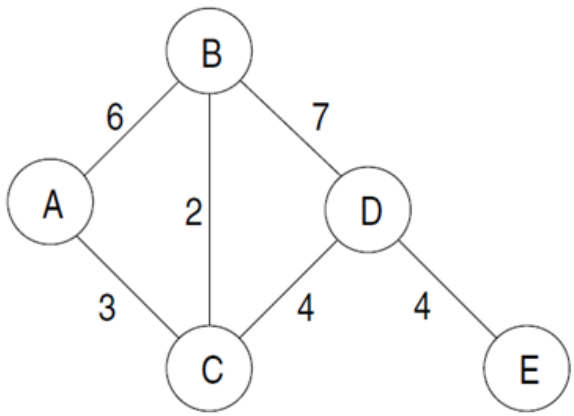
Write a simple program using the assembly language commands in the table above to demonstrate how two numbers can be added together and stored in a register. [3]

2. Explain the terms indexed sequential file and multilevel index. Draw a diagram to demonstrate the operation of a three-level index. [6]

3. Discuss random access files, making reference to:
   - the purpose of a hashing algorithm
   - the need for an overflow area
   - the need for the random access file to be re-organised on occasions. [6]

4. Produce a forwarding table for node A, calculating the lowest cost routes on the network below.

You may assume traversal of a node costs zero. [3]



5. Wales-Cycles owns a number of bicycle shops which sell bicycles, accessories, etc.

The table below shows some of items which it has for sale in its various shops, together with other information such as manufacturer details.

Item Code	Item Name	Shop Code	Shop Address	Shop Manager	Number In Stock	Manuf Code	Manuf Name	Manuf Contact
B2347	Smart Bike	S174	Manchester	Davies	4	M68	Dragon	Phillips
B2347	Smart Bike	S162	Bolton	Roggers	2	M68	Dragon	Phillips
A7219	Deluxe Hemlet	S162	Bolton	Roggers	12	M17	Safe Heads	Elhami
C2391	Peddles	S115	Stockport	Green	8	M68	Dragon	Phillips
C3844	Peddles	S162	Bolton	Roggers	5	M22	Fast Bikes	Kazembe
A1955	Bright Jacket	S201	Rochdale	Williams	22	M61	Nite Safe	Earley
A1955	Bright Jacket	S174	Manchester	Davies	16	M61	Nite Safe	Earley

The data in the table above has not been normalised.

(a) Using examples from the table, describe **two** problems which result when data has not been normalised. [4]

(b) Restructure the table shown above into third normal form.
There is no need to copy the data items. [6]

(c) Describe why distributed databases are often used and identify one difficulty associated with using distributed databases. Explain what is actually distributed in a distributed database. [3]

6. Two tables have been created in a database using SQL commands. They are:

PUPIL

PupNum	PupName	DateBirth	TeachNum
12675	Brown	02-Nov-96	307
13670	Jones	01-Jun-95	378
13777	Walker	23-Oct-95	307
14156	Vaas	29-Feb-96	307
14238	Thomas	18-Jan-95	378

TEACHER

TeachNum	TeachName	RoomNum
307	Kholi	106
345	Lester	113
378	Davies	113

(a) Write an SQL command to output the names and teacher numbers of all the pupils. [1]

(b) Write an SQL command to output the names of pupils who have the teacher with number 307. [1]

(c) Write an SQL command to change the teacher number of all those pupils whose teacher number is currently 378. Their new teacher number should be 345. [2]

(d) Write an SQL command to output the names of all pupils who have the same teacher as pupil 14238. [2]

(e) Write an SQL command to create a new table COMPUTER to contain the serial numbers of each computer at the school (assume there is only one computer per room). [2]

(f) Write an SQL command to enter the following data into the new table.

- Room 106 should have a computer with serial number 13457
- Room 113 should have a computer with serial number 66870 [2]

A certain computer uses two complementation and an 8 bit register.

7. (a) (i) Showing all your working, calculate the result of the binary addition of -7_{10} and 4_{10}. [2]

(ii) Showing all your working, calculate the result of the binary subtraction of 5_{10} from 12_{10}. [3]

(b) (i) Real numbers stored in floating point form can be stored using 16 bits as shown below:

Mantissa (12 bits in two's complement form. The binary point in the mantissa is immediately after the left bit.)	**Exponent** (4 bits in two's complement form.)

Convert the number 5.625_{10} into this floating point form. Your final answer should be normalised. [3]

(ii) In a different computer system, the following is a floating point representation of a number, using an 8 bit mantissa and a 4 bit exponent:

$0 \bullet 1111000_2 \ 0011_2$

Calculate the mantissa, exponent and decimal equivalent of the number. [3]

(iii) In the same computer system as Question 7(b)(ii), the closest possible representation of the denary number 6.9_{10} is shown below.

$0 \bullet 1101110_2 \ 0011_2$

There can be a loss of accuracy when a denary number is stored using floating point representation.

Calculate the absolute and relative error that has occurred above and explain how the floating point system used could be modified to allow a more accurate representation of 6.9_{10}. [6]

(c) (i) Calculate the effect of carrying out an arithmetic shift left by two places on the eight-bit positive integer 00001111_2 and state the effect of this operation on the number. [2]

(ii) If an arithmetic shift left by two places was carried out on the eight-bit positive integer 01001111_2, a problem would arise. Name and describe the problem. [2]

8. Explain what is meant by Big Data and data mining. Giving an example, explain what is meant by predictive analytics. [4]

9. (a) An operating system on a large computer uses multiprogramming. Describe the main features of a multiprogramming operating system. [8]

(b) (i) Explain what is meant by the term computer interrupt and describe three conditions or events which could generate interrupts. [4]

(ii) Describe high priority interrupt handling. [3]

10. A new suspended railway system uses driverless trains. A computer system controls the movement of trains between stations. Discuss the possible benefits of this system. [3]

11. Explain the need for and the purpose of cryptography. [3]

12. Explain the limiting factors to parallelisation in parallel processing. [10]

UNIT 1

MARK SCHEME

Guidance for examiners

Positive marking

It should be remembered that learners are writing under examination conditions and credit should be given for what the learner writes, rather than adopting the approach of penalising him/her for any omissions. It should be possible for a very good response to achieve full marks and a very poor one to achieve zero marks. Marks should not be deducted for a less than perfect answer if it satisfies the criteria of the mark scheme.

For questions that are objective or points-based the mark scheme should be applied precisely. Marks should be awarded as indicated and no further subdivision made.

For band marked questions mark schemes are in two parts.

Part 1 is advice on the indicative content that suggests the range of computer science concepts, theory, issues and arguments which may be included in the learner's answers. These can be used to assess the quality of the learner's response.

Part 2 is an assessment grid advising bands and associated marks that should be given to responses which demonstrate the qualities needed in AO1, AO2 and AO3. Where a response is not credit worthy or not attempted it is indicated on the grid as mark band zero.

Banded mark schemes

Banded mark schemes are divided so that each band has a relevant descriptor. The descriptor for the band provides a description of the performance level for that band. Each band contains marks.

Examiners should first read and annotate a learner's answer to pick out the evidence that is being assessed in that question. Once the annotation is complete, the mark scheme can be applied.

This is done as a two stage process.

Stage 1 – Deciding on the band

When deciding on a band, the answer should be viewed holistically. Beginning at the lowest band, examiners should look at the learner's answer and check whether it matches the descriptor for that band. Examiners should look at the descriptor for that band and see if it matches the qualities shown in the learner's answer. If the descriptor at the lowest band is satisfied, examiners should move up to the next band and repeat this process for each band until the descriptor matches the answer.

If an answer covers different aspects of different bands within the mark scheme, a 'best fit' approach should be adopted to decide on the band and then the learner's response should be used to decide on the mark within the band. For instance if a response is mainly in band 2 but with a limited amount of band 3 content, the answer would be placed in band 2, but the mark awarded would be close to the top of band 2 as a result of the band 3 content. Examiners should not seek to mark candidates down as a result of small omissions in minor areas of an answer.

Stage 2 – Deciding on the mark

Once the band has been decided, examiners can then assign a mark. During standardising (marking conference), detailed advice from the Principal Examiner on the qualities of each mark band will be given. Examiners will then receive examples of answers in each mark band that have been awarded a mark by the Principal Examiner. Examiners should mark the examples and compare their marks with those of the Principal Examiner.

When marking, examiners can use these examples to decide whether a learner's response is of a superior, inferior or comparable standard to the example. Examiners are reminded of the need to revisit the answer as they apply the mark scheme in order to confirm that the band and the mark allocated is appropriate to the response provided.

Indicative content is also provided for banded mark schemes. Indicative content is not exhaustive, and any other valid points must be credited. In order to reach the highest bands of the mark scheme a learner need not cover all of the points mentioned in the indicative content but must meet the requirements of the highest mark band. Where a response is not creditworthy, that is contains nothing of any significance to the mark scheme, or where no response has been provided, no marks should be awarded.

Q	Answer	Mark	AO1	AO2	AO3	Total
1	RAM (any three of) • it has fast read and write access and is volatile • used to store data and currently running programs • RAM is needed because most data on computers is stored in much slower "storage media" such as hard disks, solid state drives or flash memory • For the processor to be able to work on data or run programs at any reasonable speed, the programs or data need to be copied into RAM first. Cache memory (any three of) • is similar to RAM, except it resides on or close to the CPU • is faster than RAM and is also volatile • used to store frequently used data from main memory • used by the processor to avoid having to slow down to the speed of the RAM all the time • usually quite low-capacity (a few megabytes), so RAM is still needed in order to avoid constantly accessing things from slow storage media.	3 3	1.1b 1.1b			6
2	3 marks for functional characteristics of each device x 2 External hard disc drive: • Speed of access – Very fast data transfer, only flash memory is faster • Cost per unit of storage – external hard disc is quite cheap per byte of storage • Portability reason – external hard disc is physically quite small and can be easily stored securely and safely for example in a fire proof safe Third party storage provider: • Speed of access – very fast transfer achievable (depending on network speed) • Cost per unit of storage – could be cheaper or more expensive than external disk – accept either with justification • Data is stored securely and safely on protected servers Flash memory drive: • Speed of access – Very fast transfer which is important for daily updates • Cost per unit of storage – pen drive is quite cheap • Portability reason – pen drive is physically small and can be easily stored securely and safely for example in a fire proof safe Magnetic tape / tape streamer: • Speed of access reason – Access to tape is serial and can be slow but could only back up files amended that day • Cost per unit of storage reason – Tape is relatively	6	1.1b			6

<table>
<tr><th>Q</th><th>Answer</th><th>Mark</th><th>AO1</th><th>AO2</th><th>AO3</th><th>Total</th></tr>
<tr><td></td><td>(but drive can be expensive) cheap compared with other secondary storage mediums
• Portability reason – Tape is physically small and can be easily stored securely and safely for example in a fire proof safe</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>3a</td><td>DHCP - assigning dynamic IP addresses to devices on a network

HTTP - transferring multimedia web pages over the Internet.</td><td>1

1</td><td>1.1b

1.1b</td><td></td><td></td><td>2</td></tr>
<tr><td>3b</td><td>• The file transfer protocol, breaks data into packets and can re-send lost or damaged packets
• it allows packets that have arrived in a random order to be reassembled
• This is convenient for downloading files if network traffic is slows or some of your packets are dropped / arrive out of order
• However, the FTP protocol won't work as well for streaming media as it is more important to continue to receive new packets rather than retransmitting lost or dropped packets

• Voice and video traffic is can be transmitted using UDP
• Real-time video and audio streaming protocols are designed to handle occasional lost packets, so only slight degradation in quality occurs, rather than large delays if lost packets were retransmitted</td><td>1
1
1
1
1
1</td><td></td><td>2.1a
2.1a
2.1a
2.1a
2.1a
2.1a</td><td></td><td>6</td></tr>
<tr><td>3c</td><td>The exchange of signals between devices to establish their readiness to communicate.

Example: Establishing a printers readiness to print</td><td>1

1</td><td>1.1b</td><td>

2.1a</td><td></td><td>2</td></tr>
<tr><td>4a</td><td>AND

<table><tr><th>Input (A)</th><th>Input (B)</th><th>Output (A AND B)</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td>1

1</td><td>

1.1a</td><td>2.1a</td><td></td><td>2</td></tr>
<tr><td>4b</td><td>Any one of:
• Picks out / produces right bit / least significant bit (which is 1)
• Determines whether right bit / least significant bit is a 0 or 1</td><td>1</td><td></td><td>2.1a</td><td></td><td>1</td></tr>
<tr><td>5</td><td>Serial transmission: data is sent one bit at a time along the same data line

Advantage (any one of:)
• requires only two wires compared with 8 or 16 in parallel
• serial can travel longer distances than parallel
• simpler interface / circuit board / fewer lines required</td><td>1

1</td><td>1.1a

1.1b</td><td></td><td></td><td>4</td></tr>
</table>

Q	Answer	Mark	AO1	AO2	AO3	Total
	Parallel transmission: all bits in a byte are sent simultaneously along separate lines **Advantage** • transmission is faster than serial transmission	1 1	1.1a 1.1b			
6a	Fragmentation: related data is split and stored on different parts of the disc. If data is fragmented, it takes longer for the disc heads to move between parts of the file, which slows the process of loading it. Defragmentation is the process where files are physically re-arranged on disk so that they are no longer fragmented and the parts of each file are stored together. This improves the speed of accessing data from disk.	1 1 1 1	1.1b 1.1b 1.1b 1.1b			4
6b	Any three of: • SSD uses direct access to data (files) so there would be no improvement in read times as there's no physical read-head to move • Defragmentation may perform "trim" command which may slightly improve the speed of future write operations • SSD is currently made out NAND based flash memory • NAND based flash memory has a limited lifespan – defragmentation process may shorten its lifespan.	3	1.1b			3
7	**Backup** • Backup is a redundant copy of files, usually stored separately from the original system • It can be used to recover data in the event of catastrophic failure of the original storage media **Generations of files** • A generation file backup system involves storage of several of the most recent versions of a master file *Accept grandfather-father-son method* • Useful if one version is corrupted: the previous version(s) is still available **Transaction logs** • A transaction log is used with on-line updating - stores all the update data • It can be used in case of failure - could restore data by being combined with previous master/backup file, with minimal data loss.	 1 1 1 1 1 1	 1.1b 1.1b 1.1b 1.1b 1.1b 1.1b			6
8a	00110110_2 $\underline{00101110_2 +}$ 01100100 Hexadecimal value = 64_{16}	1 1		2.1a 2.1a		2
8b	10001100_2	1		2.1a		1
8c	11110101_2 One method is: • From RHS, rewrite it up to and including the first one	1 1	 1.1b	2.1a		3

Q	Answer	Mark	AO1	AO2	AO3	Total
	• Change other 1 digits to 0 and 0 digits to 1 Alternatively • Flip the bits • Add one • (Ignore carry (ninth bit)) (Other methods equally acceptable)	1	1.1b			
8di	Any one of: • are not normally stored accurately • require more complex processing • no exact representation of zero	1	1.1b			1
8dii	000101111100_2 0101_2 1 for correct mantissa, 1 for correct exponent	2		2.1a		2
8diii	• Mantissa = 0.6875_{10} or 11/16, Exponent = 5_{10} • Answer = Mantissa x $2^{exponent}$ • Answer = 22_{10}	1 1 1		2.1a 2.1a 2.1a		3
9a	A.(B + C) (A.B) + (A.C)	 1		 2.1a		1
9b	A.B + A.(B + C) + B.(B + C) A.B + A.B + A.C + B.B + B.C A.B + A.B + A.C + B + B.C A.B + A.C + B + B.C A.B + A.C + B B + A.C	 1 1 1 1 1		 2.1a 2.1a 2.1a 2.1a 2.1a		5
10a	OR	1		2.1a		1

10b	**Example**
	`24 if Num MOD Divisor = 0 then`
	`25  set Prime = FALSE`
	`26 endif`
	
	OR
	
	`17 if Prime = TRUE then`
	`18  output Num, "is a prime number"`
	`19  else`
	`20  output Num, "is NOT a prime number"`
	`21  Endif`
	
	The purpose of selection is to execute code if a certain condition is met.
		1
		
		1
			1.1b	2.1a		2

10c	**Example**
	`10 repeat`
	`11  if Num MOD Divisor = 0 then`
	`12  set Prime = FALSE`
	`13  endif`
	`14  set Divisor = Divisor + 1`
	`15 until (Prime = FALSE) OR (Divisor = Num)`
	
	The purpose of repetition is to repeatedly execute code until a certain condition is met.
		1
		
		1
			1.1b	2.1a		2

10d	• correct prime numbers and output " "x is a prime number" e.g. "3 is a prime number" • correct non-prime number and output "x is NOT a prime number" , e.g. " 4 is not a prime number"	1 1			3.1c 3.1c	2
11a	Ask customer to type password twice and compare both	1		2.1a		1

<table>
<tr><th>Q</th><th>Answer</th><th>Mark</th><th>AO1</th><th>AO2</th><th>AO3</th><th>Total</th></tr>
<tr><td></td><td>inputs to check that they are the same</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>11b</td><td>1 mark for check correctly described
1 mark for each example of invalid data that the check described would detect

<table>
<tr><th>Suitable checks</th><th>Example of invalid data</th></tr>
<tr><td>Range check – entries between sensible limits, e.g. 0-60</td><td>-1 or 74</td></tr>
<tr><td>Type check – all entries should be integer</td><td>B or #</td></tr>
</table>

NOTE - Do not accept length check, format check or look up check and examples of invalid data must follow check described</td><td>1
1</td><td></td><td>2.1a
2.1a</td><td></td><td>2</td></tr>
<tr><td>11c</td><td>1 mark for check correctly described
1 mark for each example of invalid data that the check described would detect

<table>
<tr><th>Suitable checks</th><th>Example of invalid data</th></tr>
<tr><td>Format check - email address has a string@string.string</td><td>abczyz$email.co.uk</td></tr>
</table>

NOTE - Do not accept length check, type check or look up check and examples of invalid data must follow check described</td><td>1
1</td><td></td><td>2.1a
2.1a</td><td></td><td>2</td></tr>
<tr><td>12</td><td>
<pre>
declare Rainfall array (1..12) of
integer
set Total = 0
for Count = 1 to 12
   input Rainfall(Count)
   set Total = Total + Rainfall(Count)
endfor
set Mean = Total / 12
output "Total = ", Total
ouput "Mean = ", Mean
output "Months above Mean = "
for Count = 1 to 12
   if Rainfall(Count) > Mean then
         output Count
Endfor
</pre>
Marking
• Declare array and initialise variables
• Input loop structure + increment
• Calculate mean
• Output Total and Mean
• Output loop structures
• Detect and output above mean months</td><td>
1
1
1
1
1
1</td><td></td><td></td><td>
3.1b
3.1b
3.1b
3.1b
3.1b
3.1b</td><td>6</td></tr>
<tr><td>13a</td><td>• Alpha testing – when software is issued to a restricted</td><td>1</td><td>1.1b</td><td></td><td></td><td>3</td></tr>
</table>

Q	Answer	Mark	AO1	AO2	AO3	Total
	audience of testers within the developer's own company • Beta testing - when a version is released to a number of people external to the company e.g. privileged customers in exchange for their constructive comments • Acceptance testing - when testing is carried out to prove to the customer / end user that the system works correctly.	 1 1	 1.1b 1.1b			
13b	• Perfective - is when the performance/functionality of the program has to be enhanced • Adaptive – is when the program has to be altered e.g. to run on a different operating system • Corrective – is while the program is being used and an error is discovered and corrected	1 1 1	1.1b 1.1b 1.1b			3
14	For each stage, 1 mark for each bullet point up to a maximum of 2 marks No marks for simply naming stages Lexical analysis • Comments and unneeded spaces are removed • Keywords, constants and identifiers are replaced by 'tokens' • A symbol table is created which holds the addresses of variables, labels and subroutines Syntax analysis • Tokens are checked to see if they match the spelling and grammar expected, using standard language definitions. This is done by parsing each token to determine if it uses the correct syntax for the programming language. • If syntax errors are found, error messages are produced Semantic analysis • Variables are checked to ensure that they have been properly declared and used • Variables are checked to ensure they are of the correct data type, e.g. real values are not being assigned to integers • Operations are checked to ensure that they are legal for the type of variable being used e.g. you would not try to store the result of a division operation as an integer Code generation • Machine code is generated • Code optimisation may be employed to make it more efficient / faster / less resource intense	8	1.1b			8
15	**Indicative content** • Data compression reduces the file size Lossy data compression • Compressed files can never be recovered exactly as	8	1.1b			8

Q	Answer	Mark	AO1	AO2	AO3	Total
	they were before they were compressed • When compressed files are decompressed they do not give back the original data, i.e. data is lost • Because lossy compression cannot be decompressed to yield the exact original data, it is not a good method of compression for critical data, such as textual data • It is most useful for digitally sampled analogue data, such as sound, video, graphics or images • Algorithms for lossy compression vary, but many use a threshold level truncation. This means that a level is chosen past which all data is truncated, e.g. in a sound file, the very high and low frequencies, which the human ear can not hear, may be truncated from the file • Some examples of lossy data compression algorithms are JPEG, MPEG, and MP3. Lossless data compression • The original message can be decompressed back to its original form (recovers all original data) • Lossless data compression works by finding repeated patterns in data and compressing those patterns in an efficient manner. For this reason, lossless data compression is also referred to as redundancy reduction. Becuase redundancy reduction is dependent on patterns in the message, it does not work well on random messages. Lossless data compression is ideal for text. Most of the algorithms for lossless compression are based on the LZ compression method developed by Lempel and Ziv. • One type of text encoding which is very effective for files with long strings of repeating bits is RLE. RLE stands for Run Length Encoding • RLE uses a sliding dictionary method of the LZ algorithm. The sliding dictionary method utilizes pointers within the compressed file that point to previously represented strings of bits within the file. • Here is an example of a message which could be effectively encoded with RLE: ○ The word the, is the most frequently used word in the English language. The string "the" could be represented only once and could be pointed to by all later calls to that string • Huffman coding works by analyzing the frequency of elements in data. The elements with the highest frequency get assigned the shortest encoding (with the fewest bits). Elements with lower frequencies get assigned longer encodings (with more bits) • Huffman coding could be used to compress sound files, particulary recordings containing frequecies of that heard in a human voice. *Other compression techniques accepted.*					

Band	AO1.1b Max 8 marks
3	**7 - 8 marks** The candidate has: • written an extended response that has a sustained line of reasoning which is coherent, relevant, and logically structured • shown clear understanding of the requirements of the question and a clear knowledge of the indicative content. Clear knowledge is defined as a response that provides seven to eight relevant detailed points on lossy and lossless data compression techniques, which relate to an extensive amount of the indicative content • addressed the question appropriately with minimal repetition and no irrelevant material • has presented a balanced discussion and justified their answer with examples • used appropriate technical terminology referring to the indicative content confidently and accurately.
2	**3 - 6 marks** The candidate has: • written a response that has an adequate line of reasoning with elements of coherence, relevance, and logical structure • shown adequate understanding of the requirements of the question and a satisfactory knowledge of the topic of changeover as specified in the indicative content. Satisfactory knowledge is defined as a response that provides three to six points on lossy and lossless data compression techniques as signalled in the indicative content. Up to five marks could be awarded to a response that provides detailed points on one data compression techniques (lossy or lossless) • has presented a discussion with limited examples • used appropriate technical terminology referring to the indicative content.
1	**1 - 2 marks** The candidate has: • written a response that that lacks sufficient reasoning and structure • produced a discussion which is not well developed • attempted to address the question but has demonstrated superficial knowledge of the topics specified in the indicative content. Superficial knowledge is defined as a response that provides one to two points on lossy and lossless data compression techniques as signalled in the indicative content • used limited technical terminology referring to the indicative content.
0	**0 marks** Response not credit worthy or not attempted.

UNIT 2

MARK SCHEME

Guidance for examiners

Positive marking

It should be remembered that learners are writing under examination conditions and credit should be given for what the learner writes, rather than adopting the approach of penalising him/her for any omissions. It should be possible for a very good response to achieve full marks and a very poor one to achieve zero marks. Marks should not be deducted for a less than perfect answer if it satisfies the criteria of the mark scheme.

For questions that are objective or points-based the mark scheme should be applied precisely. Marks should be awarded as indicated and no further subdivision made.

For band marked questions mark schemes are in two parts.

Part 1 is advice on the indicative content that suggests the range of computer science concepts, theory, issues and arguments which may be included in the learner's answers. These can be used to assess the quality of the learner's response.

Part 2 is an assessment grid advising bands and associated marks that should be given to responses which demonstrate the qualities needed in AO1, AO2 and AO3. Where a response is not credit worthy or not attempted it is indicated on the grid as mark band zero.

Banded mark schemes

Banded mark schemes are divided so that each band has a relevant descriptor. The descriptor for the band provides a description of the performance level for that band. Each band contains marks.

Examiners should first read and annotate a learner's answer to pick out the evidence that is being assessed in that question. Once the annotation is complete, the mark scheme can be applied.

This is done as a two stage process.

Stage 1 – Deciding on the band

When deciding on a band, the answer should be viewed holistically. Beginning at the lowest band, examiners should look at the learner's answer and check whether it matches the descriptor for that band. Examiners should look at the descriptor for that band and see if it matches the qualities shown in the learner's answer. If the descriptor at the lowest band is satisfied, examiners should move up to the next band and repeat this process for each band until the descriptor matches the answer.

If an answer covers different aspects of different bands within the mark scheme, a 'best fit' approach should be adopted to decide on the band and then the learner's response should be used to decide on the mark within the band. For instance if a response is mainly in band 2 but with a limited amount of band 3 content, the answer would be placed in band 2, but the mark awarded would be close to the top of band 2 as a result of the band 3 content. Examiners should not seek to mark candidates down as a result of small omissions in minor areas of an answer.

Stage 2 – Deciding on the mark

Once the band has been decided, examiners can then assign a mark. During standardising (marking conference), detailed advice from the Principal Examiner on the qualities of each mark band will be given. Examiners will then receive examples of answers in each mark band that have been awarded a mark by the Principal Examiner. Examiners should mark the examples and compare their marks with those of the Principal Examiner.

When marking, examiners can use these examples to decide whether a learner's response is of a superior, inferior or comparable standard to the example. Examiners are reminded of the need to revisit the answer as they apply the mark scheme in order to confirm that the band and the mark allocated is appropriate to the response provided.

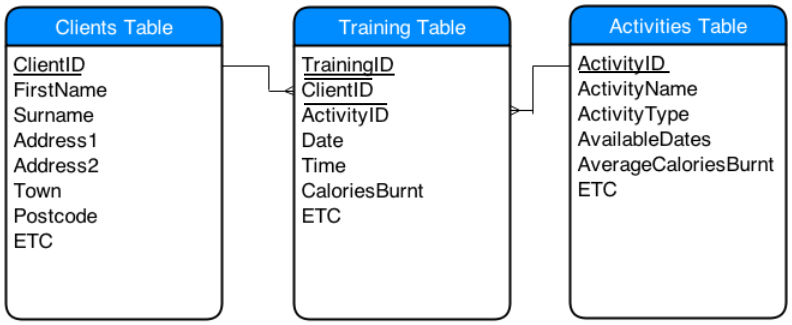
Indicative content is also provided for banded mark schemes. Indicative content is not exhaustive, and any other valid points must be credited. In order to reach the highest bands of the mark scheme a learner need not cover all of the points mentioned in the indicative content but must meet the requirements of the highest mark band. Where a response is not creditworthy, that is contains nothing of any significance to the mark scheme, or where no response has been provided, no marks should be awarded.

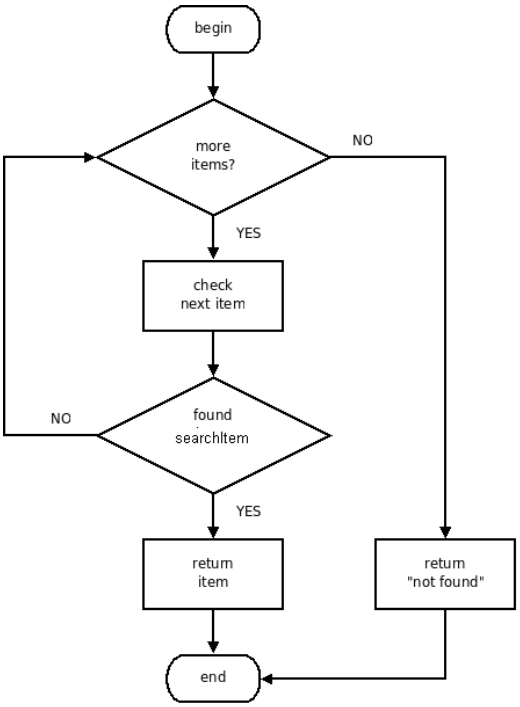
Section A

<table>
<tr><th>Q</th><th>Answer</th><th>Mark</th><th>AO1</th><th>AO2</th><th>AO3</th><th>Total</th></tr>
<tr><td>1</td><td>
Candidate has designed suitable:
• Fieldnames x 2
• Data types x 2
• Key Fields x 2
• Field lengths x 2
• Requirements for Validation (2 types) x 2
  ○ Range, Format, Presence, Length . . .

Indicative content

Non exhaustive example of Clients table:
<table>
<tr><th>Fieldname</th><th>Keyfield</th><th>Data Type</th><th>Field Length</th><th>Validation</th></tr>
<tr><td>ClientID</td><td>Yes-indexed</td><td>Integer</td><td>7</td><td>Presence</td></tr>
<tr><td>Title</td><td>-</td><td>String</td><td>10</td><td>Lookup Mr, Mrs, Miss …</td></tr>
<tr><td>FirstName</td><td>-</td><td>String</td><td>10</td><td></td></tr>
<tr><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td></tr>
<tr><td>Postcode</td><td>-</td><td>String</td><td>9</td><td>Format LL00 0LL</td></tr>
<tr><td>DOB</td><td>-</td><td>Date</td><td>2/2/4</td><td>Range 1-31, 1-12 …</td></tr>
</table>
Etc . . .
Non exhaustive example of Clients table:
<table>
<tr><th>Fieldname</th><th>Keyfield</th><th>Data Type</th><th>Field Length</th><th>Validation</th></tr>
<tr><td>ActivityID</td><td>Yes-indexed</td><td>Integer</td><td>7</td><td>Presence</td></tr>
<tr><td>ActivityName</td><td>-</td><td>String</td><td>10</td><td>-</td></tr>
<tr><td>ActivityType</td><td>-</td><td>String</td><td>10</td><td>Lookup Swimming, Weights ….</td></tr>
<tr><td>AvailableDates</td><td>-</td><td>Date</td><td>2/2/4</td><td>Range 1-31, 1-12 …</td></tr>
<tr><td>AverageCaloriesBurned</td><td>-</td><td>Real</td><td>3</td><td>Type Check – data must be Real</td></tr>
</table>
Etc. . .
</td><td>2
2
2
2
2</td><td></td><td>2.1b</td><td></td><td>10</td></tr>
</table>

Q	Answer	Mark	AO1	AO2	AO3	Total
2	 One mark for each: • Correct relationship links • Foreign key from Clients table (ClientID) • Foreign key from Activities Table (ActivityID) • Clients Table completed with example data • Training Table completed with example data • Activities Table completed with example data	1 1 1 1 1 1		2.1b 2.1b 2.1b 2.1b 2.1b 2.1b		6

Q	Answer	Mark	AO1	AO2	AO3	Total
3	One mark for each: • Correct symbols • Correct decision (search match) • Correct use of a loop (more items) • Correct use of terminating condition (more items) • Correct operation (check next item) • Correct output (return item) **Indicative content**  Notes: Flow chart will need to use standard conventions. As with programming code or pseudo-code there may be many ways to solve a given problem.	1 1 1 1 1 1		2.1b 2.1b 2.1b 2.1b 2.1b 2.1b		6

Q	Answer	Mark	AO1	AO2	AO3	Total
4	**Indicative content:** • Discussion of interface (CLI/GUI) • Data Structures (arrays/files) • File handling (serial/random) • Validation (range, format, presence, length) • Local or global variables used • Ability to handle data types (string/integer/boolean)	6		2.1b		6

Band	AO2.1b Max 6 marks
3	**5 - 6 marks** The candidate has: • written an extended response that has a sustained line of reasoning which is coherent, relevant, and logically structured • shown clear understanding of the requirements of the question and a clear knowledge of the indicative content. Clear knowledge is defined as a response that provides five to six relevant detailed points on the selection and justification of the proposed method of solution for the three main requirements listed in the scenario • addressed the question appropriately with minimal repetition and no irrelevant material • has presented a balanced discussion and justified their answer with examples • used appropriate technical terminology referring to the indicative content confidently and accurately.
2	**3 - 4 marks** The candidate has: • written a response that has an adequate line of reasoning with elements of coherence, relevance, and logical structure • shown adequate understanding of the requirements of the question and a satisfactory knowledge of the topic of changeover as specified in the indicative content. Satisfactory knowledge is defined as a response that provides three to four points on the selection and justification of the proposed method of solution for the three main requirements listed in the scenario • has presented a discussion with limited examples • used appropriate technical terminology referring to the indicative content.
1	**1 – 2 marks** The candidate has: • written a response that that lacks sufficient reasoning and structure • produced a discussion which is not well developed • attempted to address the question but has demonstrated superficial knowledge of the topics specified in the indicative content. Superficial knowledge is defined as a response that provides one to two points on the selection and justification of the proposed method of solution for the three main requirements listed in the scenario • used limited technical terminology referring to the indicative content.
0	**0 marks** Response not credit worthy or not attempted.

Q	Answer	Mark	AO1	AO2	AO3	Tot
5	**Indicative content** Answer must make reference to the impact on the clients and staff: Direct “big bang” approach can be adopted - sudden change to new system o Could be used where a failure would not be catastrophic o Can be cheaper to implement o New system is available immediately if required o Can be the least disruptive if implemented well o New system may not work as well until staff are fully used to using it o If new system fails organisation have no system which could be costly or dangerous Parallel running - both systems running together for a time o Safest option as if new system fails they still have existing system o New system is available immediately if required o The outputs from the old and new systems can be compared to check that the new system is running correctly o Expensive as require temporary staff or overtime for current staff to operate both systems o Could cause confusion for staff / customers having two systems Phased changeover - part-by-part (by functionality) o Allows users to gradually get used to the new system o Staff training can be done in stages o All staff can focus on one area to resolve any problems o Problems can be fixed quicker as more experts to resolve one functionality problem at a time o Difficulties identified in one area can be resolved and managed in next area o Might cause problems in the changeover period when they need to communicate with each other and have different systems o Slower to get new system up and running compared to some other methods o If a part of the new system fails, there is no back-up system, so data can be lost	8		2.1b		8

	Pilot changeover - part-by-part (by part of the organisation) ○ All features of the new system can be fully trialled ○ If something goes wrong with the new system, only a small part of the organisation is affected ○ The staff who were part of the pilot scheme can help train other staff. ○ All staff can focus on one area to resolve any problems ○ Difficulties identified in one area can be resolved and managed in next area ○ For the office / department doing the pilot, there is no back-up system if things go wrong ○ Might cause problems in the changeover period when they need to communicate with each other and have different systems ○ Slower to get new system up and running compared to some other methods					

Band	**AO2.1b** **Max 8 marks**
3	**7 - 8 marks** The candidate has: • written an extended response that has a sustained line of reasoning which is coherent, relevant, and logically structured • shown clear understanding of the requirements of the question and a clear knowledge of the indicative content. Clear knowledge is defined as a response that provides seven to eight relevant detailed points on changeover methods, which relate to an extensive amount of the indicative content • addressed the question appropriately with minimal repetition and no irrelevant material • has presented a balanced discussion and justified their answer with examples related to the clients and staff • used appropriate technical terminology referring to the indicative content confidently and accurately.
2	**3 - 6 marks** The candidate has: • written a response that has an adequate line of reasoning with elements of coherence, relevance, and logical structure • shown adequate understanding of the requirements of the question and a satisfactory knowledge of the topic of changeover as specified in the indicative content. Satisfactory knowledge is defined as a response that provides three to six points on changeover methods as signalled in the indicative content. • has presented a discussion with limited examples • used appropriate technical terminology referring to the indicative content.
1	**1 - 2 marks** The candidate has: • written a response that that lacks sufficient reasoning and structure • produced a discussion which is not well developed • attempted to address the question but has demonstrated superficial knowledge of the topics specified in the indicative content. Superficial knowledge is defined as a response that provides one to two points on changeover methods as signalled in the indicative content • used limited technical terminology referring to the indicative content.
0	**0 marks** Response not credit worthy or not attempted.

Q	Answer	Mark	AO1	AO2	AO3	Total

6	Any valid/functional comparison based algorithm that returns outputs as stated in question:
	
	Example
	
	`set i = 1`
	`set Position = -1`
	`set Found = false`
	`repeat`
	` if MyArray[i] = MyArray[i - 1] then`
	`  set Position = i`
	`  output “Position =”, i`
	`  output “TRUE”`
	`  set Found = true`
	` else`
	`  set i = i + 1`
	` endif`
	`until (Found = true OR i > LEN(MyArray))`
	`if Found = false then`
	`output “FALSE”`
	
	One mark for each:
						8

	• initialise variables	1			3.1b	
	• use of a loop	1			3.1b	
	• comparison	1			3.1b	
	• uses a flag to track “found”	1			3.1b	
	• use of terminating condition	1			3.1b	
	• use of logical operator for multiple terminating conditions	1			3.1b	
	• if “found” statement	1			3.1b	
	• correct outputs	1			3.1b	
	Marks awarded for concepts demonstrated above. Other solutions incorporating above concepts that provide exactly the same result would be awarded the mark. N.B. Above algorithm searches for adjacent duplicates in a pre populated array as stated in the question.					

Section B

Q	Answer	Mark	AO1	AO2	AO3	Total
1	**Indicative content:** • Opening a data file • Reading contents • Comparing Activities to the list • Incrementing the contents of the activities array ○ Alternatively candidate may be unable to implement array, multiple variables accepted	4			3.1b	4

Band	AO3.1b Max 4 marks
3	**4 marks** The candidate has: • Implemented all the points required as stated in the indicative content • Used and fully exploited the programming facilities of the language • Demonstrated a sound understanding of the appropriate tools and techniques available to them
2	**2 - 3 marks** The candidate has: • Implemented the majority of the points required as stated in the indicative content. Majority is defined as a response that provides two or three items of the functionality signalled in the indicative content • Used and exploited the programming facilities of the language • Demonstrated an understanding of the tools and techniques available to them
1	**1 mark** The candidate has: • Implemented only one of the points required as stated in the indicative content • Used some of the programming facilities of the language • Demonstrated a limited understanding of the tools and techniques available to them
0	**0 marks** Response not credit worthy or not attempted.

Q	Answer	Mark	AO1	AO2	AO3	Total
2	**Indicative content:** • Input Validation methods of: o Range check o Format check o Length check o Presence check • Stores on disc in a text file called customerdetails.txt • Retrieves specified customer from disc • HCI fit for purpose (Textual or GUI)	8			3.1b	8

Band	AO3.1b Max 8 marks
3	**7-8 marks** The candidate has: • Created a new program including the majority of the functionality as required in the question and stated in the indicative content. The majority of the functionality is defined as a response that provides seven to eight items of the functionality signalled in the indicative content • Used and fully exploited the programming facilities of the language • Demonstrated a sound understanding of the appropriate tools and techniques available to them • Written code that is well structured • Provided evidence of a completed user interface which aids user interaction and is intuitive
2	**3-6 marks** The candidate has: • Created a new program including most of the functionality as required in the question and stated in the indicative content. Most of the functionality is defined as a response that provides three to six items of the functionality signalled in the indicative content • Made use of an appropriate range of the programming facilities of the language • Demonstrated an understanding of the tools and techniques available to them • Provided evidence of a completed user interface which aids user interaction
1	**1-2 marks** The candidate has: • Created a new program with a limited range of the functionality as stated in the indicative content or improved the prototype provided by adding a limited range of the new functionality as stated in the indicative content. A limited range of functionality is defined as a response that provides one to two items of the functionality signalled in the indicative content • Used a limited range of the programming facilities of the language • Demonstrated a limited understanding of the tools and techniques available to them • Provided evidence of a user interface
0	**0 marks** Response not credit worthy or not attempted.

Q	Answer	Mark	AO1	AO2	AO3	Total
3	**Indicative content:** ○ Clear annotation of steps within the following routines: ○ Validation ○ Storage of data to file ○ Retrieving specified data from file ○ Use of self-documenting identifiers / explanation of variables	4			3.1a	4

Band	AO3.1a Max 4 marks
3	**4 marks** The candidate has: • Produced listings that are appropriately laid out and included sufficient annotation to demonstrate an understanding of **all** programming routines listed in the indicative content • Written code using self-documenting identifiers / explained variables • Used appropriate technical terminology referring to the indicative content confidently and accurately.
2	**2-3 marks** **Three** marks can be awarded if the candidate has: • Produced listings that are appropriately laid out and included sufficient annotation to demonstrate an understanding of **all** programming routines listed in the indicative content • Not written code using self-documenting identifiers / not explained variables • Used appropriate technical terminology referring to the indicative content. OR • Produced listings that are appropriately laid out and included sufficient annotation to demonstrate an understanding of **two** of the programming routines listed in the indicative content • Written code using self-documenting identifiers / explained variables • Used appropriate technical terminology referring to the indicative content. **Two** marks can be awarded if the candidate has: • Produced listings that are appropriately laid out and included sufficient annotation to demonstrate an understanding of **two** of the programming routines listed in the indicative content • Not written code using self-documenting identifiers / not explained variables • Used appropriate technical terminology referring to the indicative content. OR • Produced listings that are appropriately laid out and included sufficient annotation to demonstrate an understanding of **one** of the programming routines listed in the indicative content • Written code using self-documenting identifiers / explained variables • Used appropriate technical terminology referring to the indicative content.
1	**1 mark** The candidate has: • Produced listings that are appropriately laid out and include sufficient annotation to demonstrate an understanding of **one** programming routine listed in the indicative content • Used limited technical terminology referring to the indicative content. OR • Written code using self-documenting identifiers • Used limited technical terminology referring to the indicative content.
0	**0 marks** Response not credit worthy or not attempted.

UNIT 3

MARK SCHEME

Guidance for examiners

Positive marking

It should be remembered that learners are writing under examination conditions and credit should be given for what the learner writes, rather than adopting the approach of penalising him/her for any omissions. It should be possible for a very good response to achieve full marks and a very poor one to achieve zero marks. Marks should not be deducted for a less than perfect answer if it satisfies the criteria of the mark scheme.

For questions that are objective or points-based the mark scheme should be applied precisely. Marks should be awarded as indicated and no further subdivision made.

For band marked questions mark schemes are in two parts.

Part 1 is advice on the indicative content that suggests the range of computer science concepts, theory, issues and arguments which may be included in the learner's answers. These can be used to assess the quality of the learner's response.

Part 2 is an assessment grid advising bands and associated marks that should be given to responses which demonstrate the qualities needed in AO1, AO2 and AO3. Where a response is not credit worthy or not attempted it is indicated on the grid as mark band zero.

Banded mark schemes

Banded mark schemes are divided so that each band has a relevant descriptor. The descriptor for the band provides a description of the performance level for that band. Each band contains marks.

Examiners should first read and annotate a learner's answer to pick out the evidence that is being assessed in that question. Once the annotation is complete, the mark scheme can be applied.

This is done as a two stage process.

Stage 1 – Deciding on the band

When deciding on a band, the answer should be viewed holistically. Beginning at the lowest band, examiners should look at the learner's answer and check whether it matches the descriptor for that band. Examiners should look at the descriptor for that band and see if it matches the qualities shown in the learner's answer. If the descriptor at the lowest band is satisfied, examiners should move up to the next band and repeat this process for each band until the descriptor matches the answer.

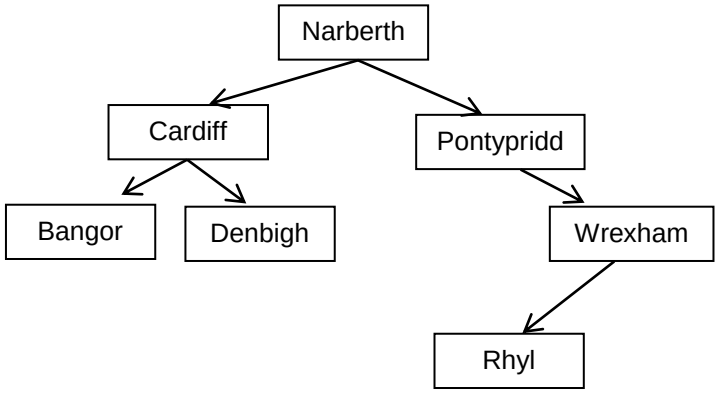
If an answer covers different aspects of different bands within the mark scheme, a 'best fit' approach should be adopted to decide on the band and then the learner's response should be used to decide on the mark within the band. For instance if a response is mainly in band 2 but with a limited amount of band 3 content, the answer would be placed in band 2, but the mark awarded would be close to the top of band 2 as a result of the band 3 content. Examiners should not seek to mark candidates down as a result of small omissions in minor areas of an answer.

Stage 2 – Deciding on the mark

Once the band has been decided, examiners can then assign a mark. During standardising (marking conference), detailed advice from the Principal Examiner on the qualities of each mark band will be given. Examiners will then receive examples of answers in each mark band that have been awarded a mark by the Principal Examiner. Examiners should mark the examples and compare their marks with those of the Principal Examiner.

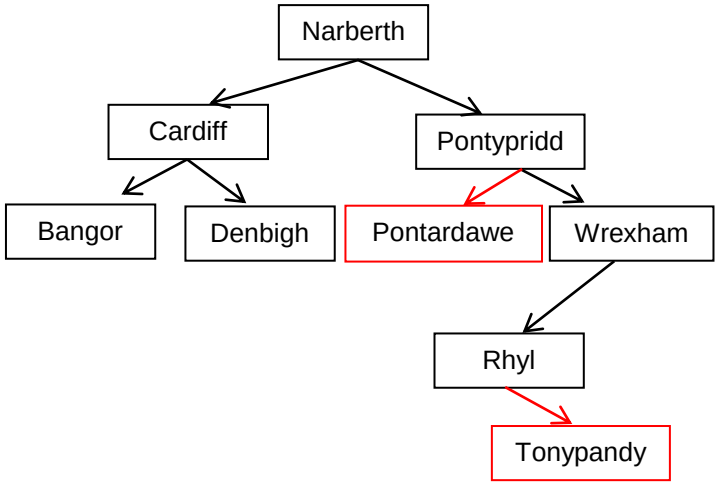
When marking, examiners can use these examples to decide whether a learner's response is of a superior, inferior or comparable standard to the example. Examiners are reminded of the need to revisit the answer as they apply the mark scheme in order to confirm that the band and the mark allocated is appropriate to the response provided.

Indicative content is also provided for banded mark schemes. Indicative content is not exhaustive, and any other valid points must be credited. In order to reach the highest bands of the mark scheme a learner need not cover all of the points mentioned in the indicative content but must meet the requirements of the highest mark band. Where a response is not creditworthy, that is contains nothing of any significance to the mark scheme, or where no response has been provided, no marks should be awarded.

<table>
<tr><th>Q</th><th>Answer</th><th>Mark</th><th>AO1</th><th>AO2</th><th>AO3</th><th>Total</th></tr>
<tr><td>1a</td><td> 

Marking
1 mark for correct root
1 mark for ALL left pointers correct and ALL right pointers correct</td><td>

1
1</td><td></td><td>

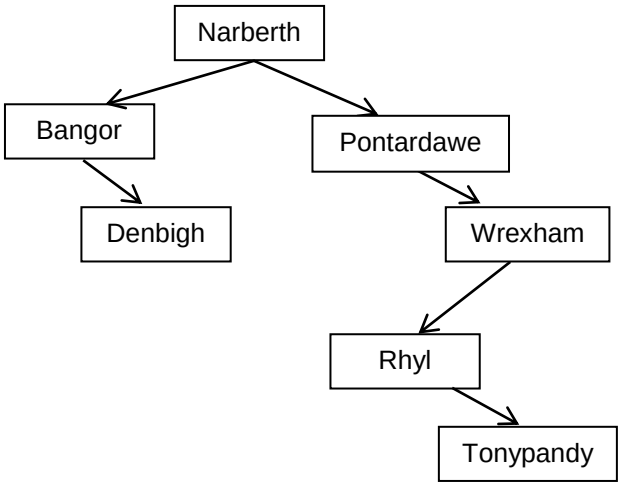
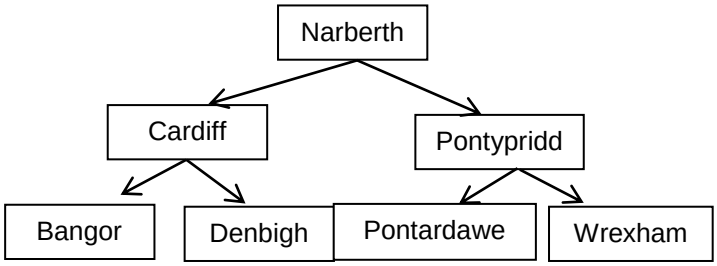
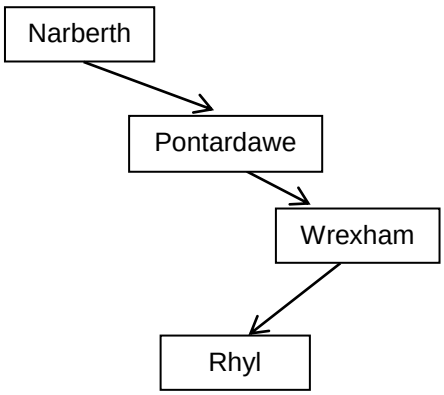
2.1b
2.1b</td><td></td><td>2</td></tr>
<tr><td>1b</td><td>
<table>
<tr><th></th><th>Left Pointer</th><th>Data</th><th>Right Pointer</th></tr>
<tr><td>Start 0</td><td>1</td><td>Narberth</td><td>2</td></tr>
<tr><td>1</td><td>5</td><td>Cardiff</td><td>6</td></tr>
<tr><td>2</td><td>-1</td><td>Pontypridd</td><td>3</td></tr>
<tr><td>3</td><td>4</td><td>Wrexham</td><td>-1</td></tr>
<tr><td>4</td><td>-1</td><td>Rhyl</td><td>-1</td></tr>
<tr><td>5</td><td>-1</td><td>Bangor</td><td>-1</td></tr>
<tr><td>6</td><td>-1</td><td>Denbigh</td><td>-1</td></tr>
<tr><td>7</td><td></td><td></td><td></td></tr>
<tr><td>8</td><td></td><td></td><td></td></tr>
<tr><td>9</td><td></td><td></td><td></td></tr>
</table>
Marking
1 mark for correct root with start pointer
1 mark for ALL left pointers correct
1 mark for ALL right pointers correct</td><td>3</td><td></td><td>2.1a</td><td></td><td>3</td></tr>
<tr><td>1ci</td><td> 
</td><td>

1

1</td><td></td><td>

2.1b

2.1b</td><td></td><td>2</td></tr>
</table>

1cii	 **Marking** 1 mark for replacing Cardiff with Bangor 1 mark for replacing Pontypridd with Pontardawe 1 mark for removing original Bangor and Pontardawe nodes	1 1 1		2.1b 2.1b 2.1b		3
1d	 The above tree is balanced and the maximum number of comparisons to locate an item would be the same as the number of levels, 3 in this case  The above tree is unbalanced and the maximum number of comparisons to locate an item would be the same as the number of items, 4 in this case **Marking** 1 mark for example of each tree – balanced and unbalanced 1 mark for each description of number of comparisons	2 2		2.1a 2.1b		4
2a	A linked list is a set (accept list) of data elements, where					4

	each element contains: • the data itself • a pointer to the next element Benefit (any one of): • New items can be inserted into a linked list without rearranging all the other elements • If programmed dynamically uses memory more efficiently Drawback (any one of): • A linked list is more complex to program / manipulate than an array • Extra programming is required to access the data in the opposite direction (or the list needs to be doubly linked) • Can only be accessed in a linear manner.	1 1 1 1	1.1b 1.1b 1.1b 1.1b			
2b	<table><tr><th>Address</th><th>Integer</th><th>Pointer</th></tr><tr><td>751</td><td>4811</td><td>756</td></tr><tr><td>752</td><td>2312</td><td>755</td></tr><tr><td>753</td><td>3599</td><td>754</td></tr><tr><td>754</td><td>4166</td><td>751</td></tr><tr><td>755</td><td>2567</td><td>757</td></tr><tr><td>756</td><td>5218</td><td>Null / -1 / 0 / End</td></tr><tr><td>757</td><td>3100</td><td>753</td></tr><tr><td>...</td><td>...</td><td>…</td></tr></table> Marking: 7 correct - 3 marks 5 or 6 correct - 2 marks 3 or 4 correct - 1 mark	3		2.1b		3
3	One mark for each valid reason for following a specific rule, up to a maximum of six marks. No marks for simply stating rules, as question requires reasons for following rules. If rule is implicit within reasoning, award mark for response (e.g. ‘data must be kept safe **to prevent data from being stolen that could cause an individual embarrassment’**) – Security rule implied. Indicative content (bold indicates example valid reasoning within a summary, other valid reasoning accepted, if justified) • Programmers should have due regard for public health, privacy, security and wellbeing of others and the environment. **This will ensure that no harm physical, emotional or financial comes to an individual from not taking these factors into account (could also be reasoned with the use of an appropriate example; for instance in the case of privacy, data could be stolen that could cause an individual embarrassment)** • Programmers should have due regard for the legitimate rights of any person or organisation that might be affected by their activities. **This will ensure that the rights of others are respected and no harm comes to the public (could also be reasoned with the use of an**	6	1.1b			6

	appropriate example; for instance the right of an individual that data held is only used for an agreed purpose and not abused) • Programmers should conduct their professional activities without discrimination on the any grounds. **This will ensure that no individual is denied their rights (could also be reasoned with the use of an appropriate example; for instance that software is carefully designed to consider other groups' needs, for example accessibility for disabled people)** • Programmers should promote equal access to the benefits of IT and seek to promote the inclusion of all sectors in society wherever opportunities arise. **This will ensure that there is no technology gap between sectors in society (could also be reasoned with the use of an appropriate example; for instance that a deprived community has opportunity access to the same website as an affluent group)** • Programmers should not claim any level of competence that they do not possess. **This safeguards an employer placing a programmer on a task that could not be completed or would be completed with significant errors which would waste time or money. (could also be reasoned with the use of an appropriate example; for instance a programmer claiming that they could use a given language but could not and then could not write the necessary program)** • Programmers should develop their professional knowledge, skills and competence on a continuing basis, maintaining awareness of technological developments, procedures, and standards that are relevant to their field. **This ensures that the product produced by a programmer is up-to-date and will function in contemporary systems (could also be reasoned with the use of an appropriate example; for instance ensuring that a programmer writes software that will function on a new operating system)** • Programmers should ensure that they have the knowledge and understanding of legislation and that they comply with such legislation, in carrying out their professional responsibilities. **This ensures that the programmer does not unwittingly break the law when undertaking their day to day job which could cause embarrassment or losses (could also be reasoned with the use of an appropriate example; for instance developing insecure software that breaches data protection laws)** • Programmers should respect and value alternative viewpoints and, seek, accept and offer honest criticisms of work. **This ensures that all relevant approached and options are considered, and the best one chosen (could also be reasoned with the use of an appropriate example; for instance when developing a user interface all opinions should be considered and the best design used)** • Programmers should avoid injuring others, their property, reputation, or employment by false or malicious or negligent action or inaction. **This ensures that staff are aware that they should consider others before taking action and do not take risks that could injure others (could also be reasoned with the use of an**					

	appropriate example; for instance programmers should avoid altering a program that may lose work for others) • Programmers should reject and not make any offer of bribery or unethical inducement. **This ensures that staff are not open to corruption from others and take actions that could harm a company or client (could also be reasoned with the use of an appropriate example; for instance programmers should not disclose sensitive data if offered an incentive to do so)** • Programmers should carry out their professional responsibilities with due care and diligence in accordance with the employer or client's requirements whilst exercising professional judgement at all times. **This would ensure that programs are developed in line with a client's requirements and that time/money is not wasted in developing other, unrequired areas (could also be reasoned with the use of an appropriate example for instance a programmer should let an employer know if a certain methodology is not working and advise on methods of changing methodology)** • Programmers should seek to avoid any situation that may give rise to a conflict of interest between them and their employer or client. **This would ensure that a programmer does not have conflicting tasks that may result in one not being completed properly (could also be reasoned with the use of an appropriate example; for instance that a programmer should not embark on a personal programming project that competes with that commissioned by a client.)** • Programmers should accept professional responsibility for their work and for the work of colleagues who are defined in a given context as working under their supervision. **This gives ownership of work, and with this, less chance of neglecting the work as the programmer is directly responsible. (could also be reasoned with the use of an appropriate example; for instance if a programmer has a set role in a task, they are likely to feel that they own that task and are more likely to do that task to the best of their ability)** • Programmers should not disclose or authorise to be disclosed, or use for personal gain or to benefit a third party, confidential information except with the permission of their employer or client, or as required by legislation. **This would undermine a client, and possibly result in loss if a competitor were to develop a product based on information disclosed. (could also be reasoned with the use of an appropriate example; for instance this prevents a programmer from selling information on a product to a company developing a similar product)** • **Programmers should not** misrepresent or withhold information on the performance of products, systems or services (unless lawfully permitted to do so by a duty of confidentiality) or take advantage of the lack of relevant knowledge or inexperience of others. **This prevents making financial or other gain from overstating the work required for a given task. (could also be**					

	reasoned with the use of an appropriate example; for instance could mean that a programmer could charge more money by stating that a simple task took longer to complete than it actually did) • Programmers should accept their personal duty to uphold the reputation of the profession and not take any action which could bring the profession into disrepute. **This ensures that the profession is not seen negatively by the wider public and not undermined by a lack of trust. (could also be reasoned with the use of an appropriate example – many potential examples)** • Programmers should encourage and support fellow members in their professional development. **This ensures that fellow members are able to support their team in development and that individuals are not undermined or lose out as a result of a lack of knowledge. (could also be reasoned with the use of an appropriate example – many potential examples)** • Programmers seek to improve professional standards through participation in their development, use and enforcement. **This ensures that programmers have ownership of the standards and these standards are more likely to be relevant to programmers as a result. (could also be reasoned with the use of an appropriate example, for instance if there were a new programming certification, the fact that programmers were part of its development would give the certification more status)** • Programmers notify the employer if convicted of a criminal offence. **This ensures that an employer can judge if a programmer can continue in their role as there may be risks if the crime is relevant to their work. (could also be reasoned with the use of an appropriate example – for instance if convicted of fraud, a programmer would not be permitted to program financial systems)**					
4a	$(A + B).(A + \overline{B})$ $A.A + A.\overline{B} + B.A + B.\overline{B}$ $A + A.\overline{B} + B.A$ A	 1 1 1		 2.1a 2.1a 2.1a		3
4b	$C + \overline{B}\overline{C}$ $C + (\bar{B} + \bar{C})$ $(C + \bar{C}) + \overline{B}$ $1 + \overline{B}$ 1	 1 1 1		 2.1a 2.1a 2.1a		3
4c	$A + (A + \overline{B.C}) + \overline{C}$ $A + (A + \overline{B} + \overline{C}) + \overline{C}$ $A + A + \overline{B} + \overline{C} + \overline{C}$ $A + \overline{B} + \overline{C}$	 1 1 1		 2.1a 2.1a 2.1a		3
5	X 10011011_2 Y 11010111_2 **XOR 01001100_2** **Retrieving the original** 01001100_2 Y 11010111_2 **XOR 10011011_2**	1 1		2.1a 2.1a		3

	The encrypted data is produced by “XOR-ing” the actual data (X) with the key (Y). The resulting encrypted data can only be read by someone who knows the key. This decryption is achieved by XOR-ing the encrypted data again using the same key (Y)to obtain the original data Accept reversed solution (XOR of Y with X).	1		1.1b		

<table>
<tr><th>Q</th><th>Answer</th><th>Marks</th><th>AO1</th><th>AO2</th><th>AO3</th><th>Total</th></tr>
<tr><td rowspan="8">6</td><td><pre>
set SearchArray(0 to n-1)
set Start = 0
set End = n-1
set Found = False

input SearchValue

repeat
   set Mid = (Start + End) DIV 2
   if SearchValue = SearchArray(Mid) then
         set Found = True
         Output "SearchValue found at position", Mid
   endif

   if SearchValue > SearchArray(Mid) then
         set Start = Mid + 1
   endif

   if SearchValue < SearchArray(Mid) then
         set End = Mid - 1
   endif
until (Found = True) OR (End < Start)

   if Found = False
         Output "SearchValue not found"
   endif
</pre>Marking</td><td></td><td></td><td></td><td></td><td rowspan="8">7</td></tr>
<tr><td>• Declare array and initialise variables</td><td>1</td><td></td><td></td><td>3.1b</td></tr>
<tr><td>• Loop structure + increment</td><td>1</td><td></td><td></td><td>3.1b</td></tr>
<tr><td>• Calculate + output position if found</td><td>1</td><td></td><td></td><td>3.1b</td></tr>
<tr><td>• Correct terminating condition for loop</td><td>1</td><td></td><td></td><td>3.1b</td></tr>
<tr><td>• Correctly discard half of array if SearchArray(Mid) > SearchValue</td><td>1</td><td></td><td></td><td>3.1b</td></tr>
<tr><td>• Correctly discard half of array if SearchArray(Mid) < SearchValue</td><td>1</td><td></td><td></td><td>3.1b</td></tr>
<tr><td>• Output message if not found</td><td>1</td><td></td><td></td><td>3.1b</td></tr>
<tr><td rowspan="5">7</td><td>Name: Recursive algorithm</td><td>1</td><td></td><td>2.1a</td><td></td><td rowspan="5">4</td></tr>
<tr><td>Features:</td><td></td><td></td><td></td><td></td></tr>
<tr><td>• Must also have a terminating condition (base case / stopping condition)</td><td>1</td><td>1.1b</td><td></td><td></td></tr>
<tr><td>• A recursive algorithm is one which calls itself</td><td>1</td><td>1.1b</td><td></td><td></td></tr>
<tr><td>Example sort: Quicksort</td><td>1</td><td>1.1a</td><td></td><td></td></tr>
</table>

Q	Answers	Marks	AO1	AO2	AO3	Total
8a	BNF is used to describe (unambiguously) the syntax / grammar / rules of a programming / computer language	1	1.1b			1
8bi	<digit> ::= 0|1|2| ... |9 <letter> ::= A|B|C| ... |Z <numeric> ::= <digit><digit><digit><digit><digit><digit> <description> ::= <letter>|<letter><description> <component code> ::=<numeric><description> **Marking:** 1 mark for recursion Same item Left and Right are needed Cannot gain full marks unless completely correct Incorrect notation – deduct 1 mark	1 1 1 1		2.1b 2.1b 2.1b 2.1b		4
8bii	digit → digit → digit → digit → digit → digit → letter →  **Marking:** 1 mark for digit x 6 (lose if "sixdigits" single box) 1 mark for letter 1 mark for recursion	 1 1 1		 2.1b 2.1b 2.1b		3
9a	All three must be correct $10^2 = 100$ $100^2 = 10{,}000$ $1000^2 = 1{,}000{,}000$	 1		 2.1a		1
9b	Evaluation of algorithm Comparison The only comparison appears in the j loop. Since this loop will iterate a total of n^2 times, it will execute exactly n^3 comparisons Data swap There may be a swap operation carried out in the j loop. Swap(A[i-1], A[i]) Each of these will happen n^2 times. Therefore there are $2n^2$ operation carried out within the j loop The i loop has one addition operation incrementing i which happens n times Adding these up we the number of addition operations which is $2n^2 + n$ As n gets very big then n^2 will dominate therefore it is $O(n^2)$ NOTE: Calculations might include assignment operations but these will not affect overall time so ignore **Marking:** 1 mark for identifying i loop will execute n times. 1 mark for identifying j loop will execute $2n^2$ times. 1 mark for correct number of calculations $2n^2 + n$ 1 mark for determining that the order will be dominated by n^2 as n gets very big giving $O(n^2)$ for the algorithm	1 1 1 1			3.1c 3.1c 3.1c 3.1c	4

Q	Answer	Marks	AO1	AO2	AO3	Total
9c	Algorithm will need to store 1 array that will require n elements. The total storage will therefore be 1 x n As n increases the storage requirements will increase by n as constant (1) will be insignificant so storage requirements will be Order (n)	1 1			3.1c	2
10	**The Waterfall approach** • Sequential design process, in which various developers draft up all of the requirements for a system up front **Advantages (any one of)** • By having all the requirements beforehand, everyone knows exactly what's needed • Client knows what to expect, including time frame, size, and cost of the project, and they know exactly what their product will do • If employees leave or join the development team, the strong documentation allows bringing new people up to speed quickly **Disadvantages (any one of)** • Because the process is sequential, once a stage of development has been completed, you can't go back to a previous stage to make changes • If the initial requirements of the project are faulty in any way, the project is almost guaranteed to fail • The product is only tested once it is completed and if bugs were made early on, a large amount of code will be affected • If the client's needs change as the project goes on, the project will take longer than predicted **The Agile approach** • Incremental approach to development, in which developers start off with a simple project design instead of a huge document, and work on small modules at a time **Advantages (any one of)** • Changes can be made after the initial planning phase, and as the client makes changes the program can be re-written • Testing is done as the product is developed, ensuring that bugs are found earlier in the process • A smaller team can work on the product because you are removing the upper layers of project managers • There can be a closer relationship between the customer and the developer • When the end goal of the product is not clearly defined, Agile development is the most suitable approach • Sprints of work on the project are done and priorities of the project are discussed, evaluated, and tested Then, a simple product is released to the consumer and they are now able to use it and provide feedback **Disadvantages (any one of)** • It can be hard to employ new people into a team when you have less of a clearly defined structural process • It can be difficult to predict when the project will be completed, or how much it will ultimately cost.	1 1 1 1 1 1	1.1b 1.1b 1.1b 1.1b 1.1b			6

Q	Answer	Marks	AO1	AO2	AO3	Total
11a	• Check the correspondence between the actual design and its specification / user requirements / objectives / safety issues • Confirm that the most appropriate techniques have been used • Confirm the HCI is appropriate	1 1 1	1.1b 1.1b 1.1b			3
11b	If programmer A modifies current version, and programmer B modifies an earlier version, neither new version will contain both modifications	1	1.1b			1
12a	**Assemblers** • An assembler converts the low level assembly programming language into machine code. **Interpreters** • An interpreter converts high level language code one line at a time, into machine code, which is then executed by the CPU. **Compilers** • A compiler translates the entire high level programming language (source code) into machine code programs prior to execution.	1 1 1	1.1b 1.1b 1.1b			3

Q	Answer	Marks	AO1	AO2	AO3	Total
12b	1 mark for naming an error x 3 1 mark for describing that error x 3 1 mark for each example x 3 **Syntax** A syntax error occurs when a command does not follow the expected syntax of the language. For instance, when a keyword is incorrectly spelt. Error • Incorrect: IF A ADN B Then • Correct: IF A AND B Then **Runtime / Execution** A runtime error is an error that only occurs when the program is running and is difficult to foresee before a program is compiled and run. Error: • Program requests more memory when none is available so the program crashes. **Logical** A logical error is an error that causes a program to output an incorrect answer (that does not necessarily crash the program). Error: • An algorithm that calculates a person's age from their date of birth but ends up giving negative numbers. **Linking** A linking error occurs when a programmer calls a function within a program and the correct library has not been linked to that program. Error: • When the Square-Root function is used and the library that calculates the Square-Root has not been linked to the program. **Rounding** Rounding is when a number is approximated to nearest whole number / tenth / hundredth, etc. Error: • 34.5 rounded to nearest whole number is 35, an error of +0.5. **Truncation** Truncating is when a number is approximated to a whole number / tenth / hundredth, etc. nearer zero. Error: • 34.9 truncated to whole number is 34, an error of-0.9.	3 3 3	1.1b 1.1b 1.1b			9

Q	Answer	Marks	AO1	AO2	AO3	Total
13	**Indicative content** **<u>Procedural languages</u>** • Procedural languages are used in traditional programming based on algorithms or a logical step-by-step process for solving a problem • They obey (ordered) instructions • They carry out actions / calculations etc. • A procedural programming language provides the programmer a way to define precisely each step when performing a task • Allows tight control over the underlying operation of the hardware • Used in (large complicated) programs where similar operations may be carried out at varying stages of the program execution **<u>Scripting Language</u>** • Set of commands understood by the applications software • Usually embedded in another language and is used to control aspects of the software • Usually a High-level programming language • Can be interpreted not compiled • Scripting languages provides the programmer a way to define precisely each step when performing a task • Allows tight integration with existing programs or data • Script embedded in (the HTML in) a web site to control graphics, etc. • Script embedded in a web site to load / execute a file when clicked, etc. **<u>Non-Procedural languages</u>** • Non-procedural programming languages allow programmers to specify the results they want without specifying how to solve the problem • Non-procedural languages are to do with rules / making queries • Used in database interrogation where retrieving answers are more important than the exact steps required to calculate the result • Artificial intelligence and modelling applications are often written in a non-procedural language **<u>Object Orientated Language</u>** • Uses objects and classes - include both data and associated processing • Applies the principles of encapsulation, inheritance and polymorphism to aid programming • Enables production of buttons / icons etc. - useful in a visual environment • A class defines the methods and properties (data) for a group of similar objects	13	1.1b			13

Q	Answer	Marks	AO1	AO2	AO3	Total
	• Once an object is created, knowledge of its implementation is not necessary for its use. • Objects control how other objects interacts with themselves, preventing other kinds of errors, e.g. a programmer cannot set the width of a window to -500 • In Visual Basic, the programmer places objects on forms. It is an event-driven language • An event, e.g. click a command button, initiates a sequence of code to be executed • Objects created using object oriented languages can easily be reused in other programs **Special Purpose Language** • Languages that were designed with a specific purpose in mind as opposed to a more general use language • Might have essential / helpful features relevant to the application • Are available for simulation, control etc. • Very specialised with built in functions/abilities that lend themselves directly to solving the problem that the language was design to work on • Used in: o Computer aided design o Artificial intelligence o Expert systems o Scientific applications o Games programming (DirectX etc) **4th Generation Language** • First generation programming languages created construct above the machine-code program • Each subsequent generation represented a further distancing from the binary code that the computer hardware actually reads • Some packages, e.g. Microsoft Access, have in-built programming capabilities. This allows the programmer to customise general purpose packages to exactly meet the needs of the business. • Generally a very high level programming language (English syntax and grammar) • Many features such as query, manipulation of data • May have report generators and possibly application generators • May attempt to produce natural language interface • Requires less programming skill • Would be useful in a database query / manipulation situation • Often used in conjunction with end user applications to customise their operation without requiring highly developed and specialised programming skills					

Q	Answer	Marks	AO1	AO2	AO3	Total
	Natural Languages • The user would not need to structure voice (or typed) input in any way - could communicate with the computer as if with another person • A natural language interface would need very high processing power / very complex software • Natural language used by most people is very ambiguous / imprecise / doesn't tend to conform to set grammar / slang is often used / English language is changing • Symbolic languages are capable of interpreting and processing queries by sentences, e.g. calculating mathematical equations • Allows the user to speak in their normal everyday language in order to interact with the computer • Speak everyday commands, such as “Open the last document I used” **Visual Programming Languages** • High level programming language • Particularly suitable for production of objects / buttons / icons, etc. • Particularly suitable for developing in a GUI / graphics content / event driven environment (e.g. double-click > execute) • May be easier to learn / more intuitive because visual / very good help / tools available **Application packages that have programming capabilities** • Additional functionality can be added without a programmer / buying another package / program • Can customise the package / tailor to specific needs etc • Requires less / no programming skill > more help is available in the package • Is probably cheaper / quicker since most facilities are provided by the package • Can import / export from / to other packages • Is less likely to contain errors “bugs” / package has already been well tested • Users are probably familiar with interface • Programming might be restricted and have certain functionality unavailable in the package					

Band	AO1.1b Max 13 marks
3	**10 - 13 marks** The candidate has: • written an extended response that has a sustained line of reasoning which is coherent, relevant, and logically structured • shown clear understanding of the requirements of the question and a clear knowledge of the indicative content. Clear knowledge is defined as a response that provides ten to thirteen relevant detailed points on the nature of different programming paradigms, which relate to an extensive amount of the indicative content • addressed the question appropriately with minimal repetition and no irrelevant material • has presented a balanced discussion and justified their answer with examples • used appropriate technical terminology referring to the indicative content confidently and accurately.
2	**5 - 9 marks** The candidate has: • written a response that has an adequate line of reasoning with elements of coherence, relevance, and logical structure • shown adequate understanding of the requirements of the question and a satisfactory knowledge of the topic of changeover as specified in the indicative content. Satisfactory knowledge is defined as a response that provides five to nine points on the nature of different programming paradigms as signalled in the indicative content. • has presented a discussion with limited examples • used appropriate technical terminology referring to the indicative content.
1	**1 - 4 marks** The candidate has: • written a response that that lacks sufficient reasoning and structure • produced a discussion which is not well developed • attempted to address the question but has demonstrated superficial knowledge of the topics specified in the indicative content. Superficial knowledge is defined as a response that provides one to four points on the nature of different programming paradigms(or a single paradigm) as signalled in the indicative content • used limited technical terminology referring to the indicative content.
0	**0 marks** Response not credit worthy or not attempted.

UNIT 4

MARK SCHEME

Guidance for examiners

Positive marking

It should be remembered that learners are writing under examination conditions and credit should be given for what the learner writes, rather than adopting the approach of penalising him/her for any omissions. It should be possible for a very good response to achieve full marks and a very poor one to achieve zero marks. Marks should not be deducted for a less than perfect answer if it satisfies the criteria of the mark scheme.

For questions that are objective or points-based the mark scheme should be applied precisely. Marks should be awarded as indicated and no further subdivision made.

For band marked questions mark schemes are in two parts.

Part 1 is advice on the indicative content that suggests the range of computer science concepts, theory, issues and arguments which may be included in the learner's answers. These can be used to assess the quality of the learner's response.

Part 2 is an assessment grid advising bands and associated marks that should be given to responses which demonstrate the qualities needed in AO1, AO2 and AO3. Where a response is not credit worthy or not attempted it is indicated on the grid as mark band zero.

Banded mark schemes

Banded mark schemes are divided so that each band has a relevant descriptor. The descriptor for the band provides a description of the performance level for that band. Each band contains marks.

Examiners should first read and annotate a learner's answer to pick out the evidence that is being assessed in that question. Once the annotation is complete, the mark scheme can be applied.

This is done as a two stage process.

Stage 1 – Deciding on the band

When deciding on a band, the answer should be viewed holistically. Beginning at the lowest band, examiners should look at the learner's answer and check whether it matches the descriptor for that band. Examiners should look at the descriptor for that band and see if it matches the qualities shown in the learner's answer. If the descriptor at the lowest band is satisfied, examiners should move up to the next band and repeat this process for each band until the descriptor matches the answer.

If an answer covers different aspects of different bands within the mark scheme, a 'best fit' approach should be adopted to decide on the band and then the learner's response should be used to decide on the mark within the band. For instance if a response is mainly in band 2 but with a limited amount of band 3 content, the answer would be placed in band 2, but the mark awarded would be close to the top of band 2 as a result of the band 3 content. Examiners should not seek to mark candidates down as a result of small omissions in minor areas of an answer.

Stage 2 – Deciding on the mark

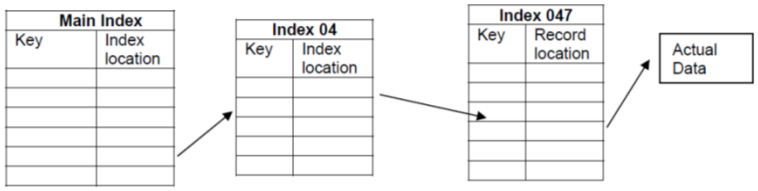
Once the band has been decided, examiners can then assign a mark. During standardising (marking conference), detailed advice from the Principal Examiner on the qualities of each mark band will be given. Examiners will then receive examples of answers in each mark band that have been awarded a mark by the Principal Examiner. Examiners should mark the examples and compare their marks with those of the Principal Examiner.

When marking, examiners can use these examples to decide whether a learner's response is of a superior, inferior or comparable standard to the example. Examiners are reminded of the need to revisit the answer as they apply the mark scheme in order to confirm that the band and the mark allocated is appropriate to the response provided.

Indicative content is also provided for banded mark schemes. Indicative content is not exhaustive, and any other valid points must be credited. In order to reach the highest bands of the mark scheme a learner need not cover all of the points mentioned in the indicative content but must meet the requirements of the highest mark band. Where a response is not creditworthy, that is contains nothing of any significance to the mark scheme, or where no response has been provided, no marks should be awarded.

Q	Answer	Mark	AO1	AO2	AO3	Total
1a	• The address of the next instruction is copied from the PC to the MAR • The instruction is copied to the MDR • The PC is incremented so that it holds the address of the next instruction	1 1 1	1.1b 1.1b 1.1b			3

1b	Using Register R as the working register and Register S as secondary register
	Memory locations containing numbers:
	0001 = location of first number
	0002 = location of second number
	0003 = location to store result
	(accept any consistent use)
	
	`CLR R`
	`LOD R, 0001`
	`LOD S, 0002`
	`ADD R,S`
	`STR R 0003`
	`CLR R`
	
	1 mark for clearing register ready for use and at end.
	1 mark for loading values into registers.
	1 mark for adding and storing result.
		
		1
		1
		1
			
					3.1b
					3.1b
					3.1b
						3

2	**Indexed sequential file** • Records in key order • Index allows data to be accessed directly / faster / contains key and record address **Multilevel index** • There is a main index which contains the location of the next index • This process may extend to several levels. The last index contains the physical address of the record Diagram:  **Marking of diagram** 1 mark for three suitably labelled tables plus “actual data” 1 mark for three suitable arrows	 1 1 1 1 1 1	 1.1b 1.1b 1.1b 1.1b 	 2.1a 2.1a	 	6
3	**The purpose of a hashing algorithm:** • file where physical location of the record is calculated from the data in the key field • this calculation is carried out by a hashing algorithm **The need for an overflow area:** • A data collision occurs when two data items are hashed to the same location • In this case there needs to be an overflow areas	 1 1 1 1	 1.1b 1.1b 1.1b 1.1b	 	 	6

<table>
<tr><th>Q</th><th>Answer</th><th>Mark</th><th>AO1</th><th>AO2</th><th>AO3</th><th>Total</th></tr>
<tr><td></td><td>where the latest data is stored
The need for the random access file to be re-organised on occasions:
• When there are many items in the overflow area, access may become slow,
• In which case a new hashing algorithm is required and a larger file may be needed</td><td>

1

1</td><td>

1.1b

1.1b</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td><table><tr><th>Destination</th><th>Cost</th><th>GoTo</th></tr><tr><td>A</td><td>0</td><td>A</td></tr><tr><td>B</td><td>5</td><td>C</td></tr><tr><td>C</td><td>3</td><td>C</td></tr><tr><td>D</td><td>7</td><td>C</td></tr><tr><td>E</td><td>11</td><td>C</td></tr></table>
1 mark for correct first, second and third row
1 mark for correct fourth row
1 mark for correct fifth row</td><td>

1
1
1</td><td></td><td>

2.1a
2.1a
2.1a</td><td></td><td>3</td></tr>
<tr><td>5a</td><td>1 mark for description x 2
1 mark for example (must match description) x 2

Any one of:
(Example may not include actual data)

• Data duplication/redundancy (storing of the same data more than once) is likely to occur if a database is not normalised.

Example: the manufacturer M68 has name (Dragon) and contact (Phillips) and is stored three (or more) times in the database, whereas the name and contact need only to be stored once.

• Update anomaly (term accepted not expected): if some data is updated, there may be a need to update other related data a number of times causing a risk of inconsistent data.

Example: if the contact at the manufacturer Dragon changes from Phillips to (say) Smith, then the name Phillips needs replacing with Smith three (or more) times in the database. This increases the chance of incorrect data being included.

• Delete anomaly (term accepted not expected): if some data is deleted, the unwanted side effect may be that other data is lost.

Example: if the data relating to Deluxe Helmet (A7129) is deleted, we also lose all data about manufacturer Safe Heads.</td><td>2
2</td><td>1.1b</td><td>
2.1a</td><td></td><td>4</td></tr>
</table>

Q	Answer	Marks	AO1	AO2	AO3	Total
5b	ITEM (ItemCode, ItemName, ManufCode) SHOP (ShopCode, ShopAddress, ShopManager) MANUFACTURER (ManufCode, ManufName, ManufContact) ITEMS-SHOPS (ItemCode, ShopCode, NoInStock) **Marking** • Four suitably named tables • Each table with a suitable primary key shown (4 correct → 2 marks 2 or 3 correct → 1 mark) • 1 mark for each correct foreign key x 3	 1 2 3		 2.1b 2.1b 2.1b		6
5c	• (Databases often contain huge amounts of data.) It is often more efficient to store data on a number of different computers (probably in different locations) to maximise performance. **Not:** improves security **Not:** will still work if one computer fails etc • It is difficult to ensure that all the data in all the computers is always up-to-date / maintain integrity. • Both processing and data are distributed across the different computers that the data is stored upon.	1 1 1	1.1b 1.1b 1.1b			3
6a	`SELECT PupName, TeachNum FROM PUPIL`	1			3.1b	1
6b	`SELECT PupName FROM PUPIL WHERE TeachNum = '307'`	1			3.1b	1
6c	`UPDATE PUPIL SET TeachNum = '345' WHERE TeachNum = '378'` 1 mark for update; 1 mark for changing TeachNum	2			3.1b	2
6d	`SELECT PupName FROM PUPIL WHERE TeachNum = (SELECT TeachNum FROM PUPIL WHERE PupNum = '14238')` 1 mark for each select (1 mark for main select and one from sub-select)	2			3.1b	2
6e	`CREATE TABLE COMPUTER ( RoomNum Char(5) SerialNum Char(5) )` 1 mark for table; 1 mark for fields (any suitable field size acceptable)	2			3.1b	2

6f	`INSERT INTO COMPUTER VALUES ('106' , '13457')`
	`INSERT INTO COMPUTER VALUES ('113' , '66870')`
	
	1 mark for insert; 1 mark for values inserted
		2			3.1b	2

Q	Answers	Marks	AO1	AO2	AO3	Total
7ai	11111001_2 00000100_2 + **11111101_2**	1 1		2.1a 2.1a		2
7aii	00001100_2 (mark for negating the 5_{10}) 11111011_2 + (mark for addition operation) **00000111_2**	1 1 1		2.1a 2.1a 2.1a		3
7bi	Mantissa = $00000101 . 1010_2$ Exponent = 0011_2 Normalised: 0.10110100000_2 x 2^{0011} or 0.703125_{10} x 2^3	1 1 1		2.1a 2.1a 2.1a		3
7bii	Mantissa = 0.9375_{10} Exponent = 3_{10} Decimal Equivalent: 0.9375_{10} x $2^3 = 7.5_{10}$	1 1 1		2.1a 2.1a 2.1a		3
7biii	Mantissa = 0.859375_{10} Exponent = 3_{10} Decimal Equivalent: 0.8595_{10} x $2^3 = 6.875_{10}$ **Absolute error** 0.025 / -0.025 / 6.9-6.875 / 1/40 **Relative error** 0.00362319 / 0.025/6.9 / 1/276 = 0.362319% **Modifications (any one of)** • Adjust the mantissa to use more bits / Accept similar wording, e.g. longer mantissa • Reallocate (one) bit; from the exponent to the mantissa • Infer one of the two bits on either side of the binary point and use the freed up bit to store one more significant digit in the mantissa	1 1 1 1 1 1		2.1a 2.1a 2.1a 2.1b 2.1b 2.1b		6
7ci	00111100_2 The effect is multiplying by 4 (i.e. by 100_2 - needs subscript) **Not** gets bigger **Not** "Moves two places to left"	1 1		2.1a 2.1b		2
7cii	(Numeric) **overflow** (number becomes (~~1~~) 00111100) The resulting number is too large to be contained in the eight bits available	1 1		2.1a 2.1a		2
8	• **Big Data** refers to data sets so large and complex that it becomes difficult to process using standard relational database techniques • **Data mining** the analysis of (a large amount of) data (in a data warehouse) • **Predictive analytics** consist of a variety of statistical techniques including modelling, machine learning, and data mining. Example: In business, predictive models analyses patterns found in historical and transactional data to identify patterns that may present risks or opportunities.	1 1 1 1	1.1b 1.1b 1.1b 1.1b			4

Q	Answer	Marks	AO1	AO2	AO3	Total
9a	• More than one job in memory at same time • More than one job processed (apparently) at same time • Time-slice is amount of time allocated to each job by operating system • Scheduling decides which job is to be processed next (may be prioritised) • Partitioning is division of computer memory for different jobs • Paging jobs in and out to make better use of memory • Promotes efficient use of CPU • Achieved by use of interrupts **Marking** • Any point can be extended for an additional mark • Max 6 for brief points only • Max 6 if not covered all four of multiprogramming, time-slicing, scheduling and memory partitioning	1 1 1 1 1 1 1 1	1.1b 1.1b 1.1b 1.1b 1.1b 1.1b 1.1b 1.1b			8
9bi	An interrupt is a signal generated by a device or software, which may cause a break in the execution of the current routine. NOTE – do not accept an interrupt is generated when there is a fault (this is an example not a description) Examples (any three of) • Hardware or Software fault • Input/output device requesting attention e.g. printer out of paper / requesting more data / key press / mouse click • User interrupt e.g. <ctrl> <break> • Operating system generated interrupt e.g. end of time slice • Run time error e.g. division by zero	1 3	1.1b 1.1b			4
9bii	Answers **MUST** be in this order: 1. O/S suspends current interrupt routine 2. Runs the new higher priority interrupt routine 3. Finally, the O/S returns to original interrupt routine and continues	 1 1 1	 1.1b 1.1b 1.1b			3
10	Any three of: • No possibility of human error (for instance passing a signal at red) • Train's control system could apply the brakes at/before a red signal • Obstruction / train ahead detection could be included • Could govern the maximum speed • Could prevent starting with any doors open • No driver so save salaries, plus no sickness, lateness etc. • Computer may apply optimisation algorithms to ensure trains reach their stations as efficiently as possible (e.g. on time and regular)	3		2.1a		3

Q	Answer	Marks	AO1	AO2	AO3	Total
11	Need for • Cryptography is needed in computer systems, both online and locally to protect the information stored on individual devices and whilst being transmitted over computer networks The purpose of • Cryptography is to encode messages or information in such a way that only parties with the appropriate decryption key can read it • The message or information, referred to as plaintext, is encrypted using an encryption algorithm and a key, generating cipher text that can only be read if decrypted using the original key	1 1 1	1.1b 1.1b 1.1b			3
12	**Indicative content** • Parallel computing is a form of computation in which many calculations are carried out simultaneously • It operates on the principle that large problems can often be divided into smaller ones, which are then solved concurrently • Parallel computer programs are more complex to design and to write than sequential ones • Concurrency introduces several new classes of potential software bugs • Race conditions are the most common class of potential software bug • Communication and synchronisation between the different subtasks are typically some of the greatest obstacles to getting efficient parallel program performance • The maximum possible speed-up of a single program as a result of parallelisation is known as Amdahl's law: ○ $T(n) = T(1)(B + \frac{1}{n}(1 - B))$ ○ $Where$: ▪ $T(n) = time\ taken\ on\ n\ threads$ ▪ $n = number\ of\ threads$ ▪ $B = fraction\ of\ algorithm\ that\ is\ sequential$ • Example: If a program needs 10 hours using a single processor core, and a particular portion of the program which takes one hour to execute cannot be parallelised, while the remaining 9 hours (90%) of execution time can be parallelised, then regardless of how many processors are devoted to a parallelised execution of this program, the minimum execution time cannot be less than that critical one hour. • Using the formula above with one thread (n=1) we get: ○ $T(n) = T(1)(B + \frac{1}{n}(1 - B)$ ○ $T(1) = T(1)(B + \frac{1}{1}(1 - B)$ ○ $T(1) = T(1)(0.9 + (1 - 0.9) = 1\ hour\ minimum$ • The speedup of a program using multiple processors in parallel computing is limited by the time needed for the sequential fraction of the program	10	1.1b			10

Band	AO1.1b Max 10 marks
3	**8 - 10 marks** The candidate has: • written an extended response that has a sustained line of reasoning which is coherent, relevant, and logically structured • shown clear understanding of the requirements of the question and a clear knowledge of the indicative content. Clear knowledge is defined as a response that provides eight to ten relevant detailed points on the limiting factors to parallelisation in parallel processing, which relate to an extensive amount of the indicative content • addressed the question appropriately with minimal repetition and no irrelevant material • has presented a balanced discussion and justified their answer with examples • used appropriate technical terminology referring to the indicative content confidently and accurately.
2	**4 - 7 marks** The candidate has: • written a response that has an adequate line of reasoning with elements of coherence, relevance, and logical structure • shown adequate understanding of the requirements of the question and a satisfactory knowledge of the indicative content. Satisfactory knowledge is defined as a response that provides four to seven points on the limiting factors to parallelisation in parallel processing as signalled in the indicative content. • has presented a discussion with limited examples • used appropriate technical terminology referring to the indicative content.
1	**1 - 3 marks** The candidate has: • written a response that that lacks sufficient reasoning and structure • produced a discussion which is not well developed • attempted to address the question but has demonstrated superficial knowledge of the topics specified in the indicative content. Superficial knowledge is defined as a response that provides one to three points on the limiting factors to parallelisation in parallel processing as signalled in the indicative content • used limited technical terminology referring to the indicative content.
0	**0 marks** Response not credit worthy or not attempted.

UNIT 1

Coverage of Assessment Objectives

Assessment Objective		Elements	Question															Total
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
AO1	**Demonstrate knowledge and understanding of the principles and concepts of computer science, including abstraction, logic, algorithms and data representation**	a – Demonstrate knowledge of the principles and concepts of abstraction, logic, algorithms, data representation or others as appropriate				1	2											**3**
		b – Demonstrate understanding of the principles and concepts of abstraction, logic, algorithms, data representation or others as appropriate	6	6	3		2	7	6	3		2			6	8	8	**57**
		TOTAL AO1	**6**	**6**	**3**	**1**	**4**	**7**	**6**	**3**	**0**	**2**	**0**	**0**	**6**	**8**	**8**	**60**
AO2	**Apply knowledge and understanding of the principles and concepts of computer science, including to analyse problems in computational terms**	a – Apply knowledge and understanding of the principles and concepts of computer science			7	2				9	6	3	5					**32**
		b – Analyse problems in computational terms																**0**
		TOTAL AO2	**0**	**0**	**7**	**2**	**0**	**0**	**0**	**9**	**6**	**3**	**5**	**0**	**0**	**0**	**0**	**32**
AO3	**Design, program and evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions**	a – Design computer systems that solve problems																**0**
		b – Program computer systems that solve problems												6				**6**
		c – Evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions										2						**2**
		TOTAL AO3	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**0**	**6**	**0**	**0**	**0**	**8**
		TOTAL AO1 + AO2 + AO3	**6**	**6**	**10**	**3**	**4**	**7**	**6**	**12**	**6**	**7**	**5**	**6**	**6**	**8**	**8**	**100**

UNIT 2

Coverage of Assessment Objectives

Assessment Objective		Elements	Question									Total
			Section: A						B			
			1	2	3	4	5	6	1	2	3	
AO1	**Demonstrate knowledge and understanding of the principles and concepts of computer science, including abstraction, logic, algorithms and data representation**	a – Demonstrate knowledge of the principles and concepts of abstraction, logic, algorithms, data representation or others as appropriate										**0**
		b – Demonstrate understanding of the principles and concepts of abstraction, logic, algorithms, data representation or others as appropriate										**0**
		TOTAL AO1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
AO2	**Apply knowledge and understanding of the principles and concepts of computer science, including to analyse problems in computational terms**	a – Apply knowledge and understanding of the principles and concepts of computer science										**0**
		b – Analyse problems in computational terms	10	6	6	6	8					**36**
		TOTAL AO2	**10**	**6**	**6**	**6**	**8**	**0**	**0**	**0**	**0**	**36**
AO3	**Design, program and evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions**	a – Design computer systems that solve problems									4	**4**
		b – Program computer systems that solve problems						8	4	8		**20**
		c – Evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions										**0**
		TOTAL AO3	**0**	**0**	**0**	**0**	**0**	**8**	**4**	**8**	**4**	**24**
		TOTAL AO1 + AO2 + AO3	**10**	**6**	**6**	**6**	**8**	**8**	**4**	**8**	**4**	**60**

UNIT 3

Coverage of Assessment Objectives

Assessment Objective		Elements	Question															Total
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
AO1	**Demonstrate knowledge and understanding of the principles and concepts of computer science, including abstraction, logic, algorithms and data representation**	a– Demonstrate knowledge of the principles and concepts of abstraction, logic, algorithms, data representation or others as appropriate							1									**1**
		b – Demonstrate understanding of the principles and concepts of abstraction, logic, algorithms, data representation or others as appropriate		4	6		1		2	1		6	4	12	13			**49**
		TOTAL AO1	**0**	**4**	**6**	**0**	**1**	**0**	**3**	**1**	**0**	**6**	**4**	**12**	**13**	**0**	**0**	**50**
AO2	**Apply knowledge and understanding of the principles and concepts of computer science, including to analyse problems in computational terms**	a – Apply knowledge and understanding of the principles and concepts of computer science	5			9	2		1		1							**18**
		b – Analyse problems in computational terms	9	3						7								**19**
		TOTAL AO2	**14**	**3**	**0**	**9**	**2**	**0**	**1**	**7**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**37**
AO3	**Design, program and evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions**	a – Design computer systems that solve problems																**0**
		b – Program computer systems that solve problems						7										**7**
		c – Evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions									6							**6**
		TOTAL AO3	**0**	**0**	**0**	**0**	**0**	**7**	**0**	**0**	**6**	**0**	**0**	**0**	**0**	**0**	**0**	**13**
		TOTAL AO1 + AO2 + AO3	**14**	**7**	**6**	**9**	**3**	**7**	**4**	**8**	**7**	**6**	**4**	**12**	**13**	**0**	**0**	**100**

UNIT 4

Coverage of Assessment Objectives

Assessment Objective		Elements	Question															Total
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
AO1	**Demonstrate knowledge and understanding of the principles and concepts of computer science, including abstraction, logic, algorithms and data representation**	a – Demonstrate knowledge of the principles and concepts of abstraction, logic, algorithms, data representation or others as appropriate																**0**
		b – Demonstrate understanding of the principles and concepts of abstraction, logic, algorithms, data representation or others as appropriate	3	4	6		5			4	15		3	10				**50**
		TOTAL AO1	**3**	**4**	**6**	**0**	**5**	**0**	**0**	**4**	**15**	**0**	**3**	**10**	**0**	**0**	**0**	**50**
AO2	**Apply knowledge and understanding of the principles and concepts of computer science, including to analyse problems in computational terms**	a – Apply knowledge and understanding of the principles and concepts of computer science		2		3	2		17			3						**27**
		b – Analyse problems in computational terms					6		4									**10**
		TOTAL AO2	**0**	**2**	**0**	**3**	**8**	**0**	**21**	**0**	**0**	**3**	**0**	**0**	**0**	**0**	**0**	**37**
AO3	**Design, program and evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions**	a – Design computer systems that solve problems																**0**
		b – Program computer systems that solve problems	3					10										**13**
		c – Evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions																**0**
		TOTAL AO3	**3**	**0**	**0**	**0**	**0**	**10**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**13**
		TOTAL AO1 + AO2 + AO3	**6**	**6**	**6**	**3**	**13**	**10**	**21**	**4**	**15**	**3**	**3**	**10**	**0**	**0**	**0**	**100**