Centre Number						Candidate Number				
Surname										
Other Names										
Candidate Signature										

For Examiner's Use	
Examiner's Initials	
Question	Mark
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
TOTAL	

AQA

General Certificate of Education
Advanced Level Examination
June 2011

Computing COMP3

Unit 3 Problem Solving, Programming, Operating Systems, Databases and Networking

Thursday 23 June 2011 9.00 am to 11.30 am

You will need no other materials.
You may use a calculator.

Time allowed
- 2 hours 30 minutes

Instructions
- Use black ink or black ball-point pen.
- Fill in the boxes at the top of this page.
- Answer **all** questions.
- You must answer the questions in the spaces provided. Do not write outside the box around each page or on blank pages.
- All working must be shown.
- Do all rough work in this book. Cross through any work you do not want to be marked.

Information
- The marks for questions are shown in brackets.
- The maximum mark for this paper is 100.
- The use of brand names will **not** gain credit.
- Questions 8(e) and 10(c) should be answered in continuous prose. In these questions you will be marked on your ability to:
  - use good English
  - organise information clearly
  - use specialist vocabulary where appropriate.

Answer **all** questions.

1 The binary search method can be used to search for an item in an ordered list.

1 (a) Show how the binary search method works by writing numbers on **Figure 1** below to indicate which values would be examined to determine if the name "Richard" appears in the list.

Write the number "1" by the first value to be examined, "2" by the second value to be examined and so on.

Figure 1

Position	Value	Order Examined In
1	Adam	
2	Alex	
3	Anna	
4	Hon	
5	Mohammed	
6	Moonis	
7	Niraj	
8	Philip	
9	Punit	
10	Ravi	
11	Richard	
12	Timothy	
13	Tushara	
14	Uzair	
15	Zara	

(3 marks)

1 (b) A different list contains 137 names.

What is the maximum number of names that would need to be accessed to determine if the name "Rachel" appears in the list? Write your answer in the box below.

□

(1 mark)

1 (c) Tick **one** box to indicate the order of time complexity of the binary search method.

Order of time complexity	Tick one box
$O(\log_2 n)$	
$O(n)$	
$O(n^2)$	

(1 mark)

5

2 A computer programmer, developing a theatre seat booking system, has decided to use an event-driven object-oriented programming language.

2 (a) The operating system that is installed on the computer that the new booking system will be used on is an interactive network operating system. This has the characteristics of both an interactive operating system and a network operating system.

Explain what is meant by the terms *interactive operating system* and *network operating system*.

Interactive operating system: ....

....

....

Network operating system: ....

....

....

(2 marks)

2 (b) To allow it to work with the computer's operating system, the programs developed by the programmer should be event-driven.

Explain how an event-driven program works.

....

....

....

....

(2 marks)

4

3 A normalised floating point representation uses an 8-bit mantissa and a 4-bit exponent, both stored using **two's complement format**.

3 (a) In binary, write in the boxes below, the smallest positive number that can be represented using this normalised floating point system.

●							

Mantissa

Exponent

(2 marks)

3 (b) This is a floating point representation of a number:

1 ●	0	1	1	0	0	0	0

Mantissa

0	0	1	0

Exponent

Calculate the denary equivalent of the number. Show your working.

Working:...

....

....

(1 mark)

Answer:..

(1 mark)

3 (c) Write the normalised floating point representation of the denary value 12.75 in the boxes below. Space has been provided for you to do rough work, if required.

Rough Work:..

....

....

Answer:

●							

Mantissa

Exponent

(2 marks)

3 (d) Floating point numbers are usually stored in normalised form.

State **two** advantages of using a normalised representation.

Advantage 1: ....

....

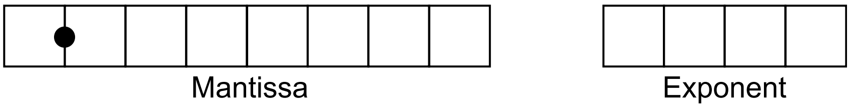
Advantage 2: ....

....

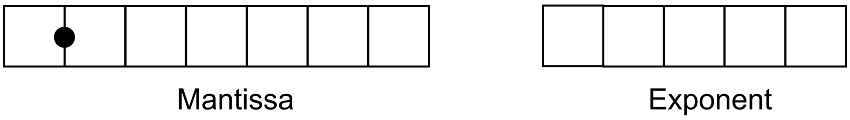
(2 marks)

3 (e) An alternative **two's complement format** representation is proposed. In the alternative representation **7 bits** will be used to store the mantissa and **5 bits** will be used to store the exponent.

Existing Representation (8-bit mantissa, 4-bit exponent):



Proposed Alternative Representation (7-bit mantissa, 5-bit exponent):



Explain the effects of using the proposed alternative representation instead of the existing representation.

....

....

....

....

....

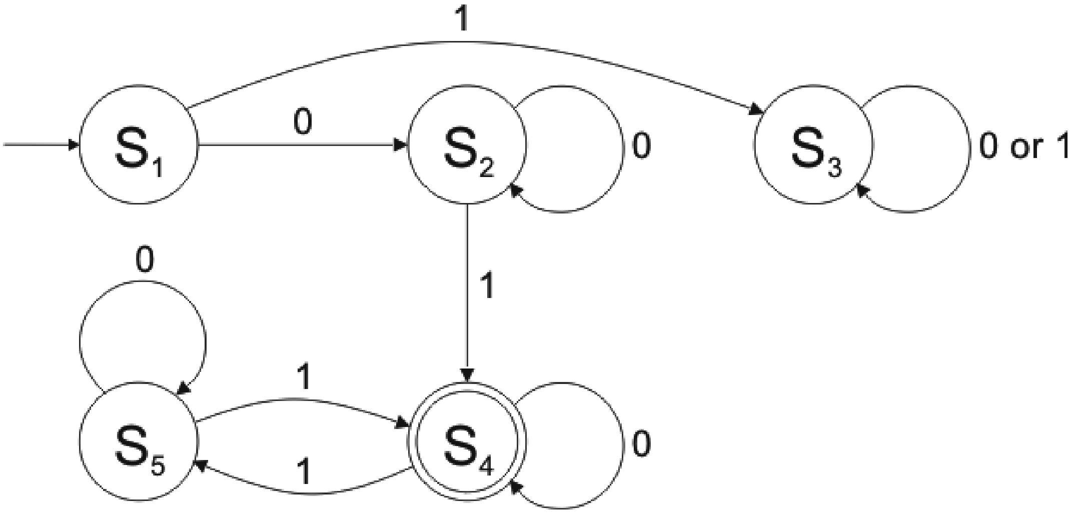
(2 marks)

10

Turn over ▸

4 **Figure 2** shows a Finite State Automaton (FSA). The FSA has input alphabet {0, 1} and five states, S_1, S_2, S_3, S_4 and S_5.

Figure 2



4 (a) Complete the transition table below for the FSA in **Figure 2**.

Current State	S_1	S_1	S_2	S_2	S_3	S_3	S_4	S_4	S_5	S_5
Input Symbol	0	1	0	1	0	1				
Next State	S_2	S_3	S_2	S_4	S_3	S_3				

(1 mark)

4 (b) The state S_4 is a special state. This is indicated by the double circle in the diagram. What does the double circle signify?

....

....

(1 mark)

4 (c) Write **Yes** or **No** in each row of the table below to indicate whether or not each of the four input strings would be accepted by the FSA in **Figure 2**.

Input String	**String Accepted? (Yes/No)**
101	
000	
010001101	
0100011011	

(2 marks)

4 (d) Describe the language (set of strings) that the FSA will accept.

....

....

....

(2 marks)

6

5 Reverse Polish Notation is an alternative to standard infix notation for writing arithmetic expressions.

5 (a) Convert the following Reverse Polish Notation expressions to their equivalent infix expressions.

Reverse Polish Notation	**Equivalent Infix Expression**
45 6 +	
12 19 + 8 *	

(2 marks)

5 (b) State **one** advantage of Reverse Polish Notation over infix notation.

....

....

(1 mark)

Question 5 continues on the next page

5 (c) The pseudo-code algorithm in **Figure 3** can be used to calculate the result of evaluating a Reverse Polish Notation expression that is stored in a string. The algorithm is designed to work only with the single digit denary numbers 0 to 9. It uses procedures and functions listed in **Table 1**, two of which operate on a stack data structure.

Figure 3

```
StringPos ← 0
Repeat
  StringPos ← StringPos + 1
  Token ← GetCharFromString(InputString, StringPos)
  If Token = '+' Or Token = '-' Or Token = '/' Or Token = '*'
    Then
      Op2 ← Pop()
      Op1 ← Pop()
      Case Token Of
        '+': Result ← Op1 + Op2
        '-': Result ← Op1 - Op2
        '/': Result ← Op1 / Op2
        '*': Result ← Op1 * Op2
      EndCase
      Push(Result)
    Else
      IntegerVal ← ConvertToInteger(Token)
      Push(IntegerVal)
  EndIf
Until StringPos = Length(InputString)
Output Result
```

Table 1

Procedure/Function	**Purpose**	**Example(s)**
GetCharFromString (InputString: String, StringPos: Integer): Char	Returns the character at position StringPos within the string InputString. Note that the leftmost letter is position 1, not position 0.	GetCharFromString ("Computing", 1) would return the character 'C'. GetCharFromString ("Computing", 3) would return the character 'm'.
ConvertToInteger (ACharacter: Char): Integer	Returns the integer equivalent of the character in ACharacter.	ConvertToInteger('4') would return the integer value 4.
Length(AString: String): Integer	Returns a count of the number of characters in the string AString.	Length("AQA") would return the integer value 3.
Push(ANumber: Integer)	Puts the number in ANumber onto the stack.	Push(6) would put the number 6 on top of the stack.
Pop(): Integer	Removes the number from the top of the stack and returns it.	X ← Pop() would remove the value from the top of the stack and put it in X.

5 (c) Complete the table below to trace the execution of the algorithm when `InputString` is the string: `64+32+*`

In the `Stack` column, show the contents of the stack once for each iteration of the `Repeat..Until` loop, as it would be at the end of the iteration.

The first row and the leftmost column of the table have been completed for you.

StringPos	**Token**	**IntegerVal**	**Op1**	**Op2**	**Result**	**Stack**
0	-	-	-	-	-	
1						
2						
3						
4						
5						
6						
7						

(5 marks)

Final output of algorithm: ....

(1 mark)

Turn over ▶

5 (d) A programmer is going to implement the algorithm from **Figure 3** in a programming language that does not provide built-in support for a stack data structure.

The programmer intends to simulate a stack by using a fixed length array of 20 integers named `StackArray` with indices running from `1` to `20` and an integer variable `TopOfStackPointer` which will be initialised to `0`.

Write a pseudo-code algorithm for the `Push` operation to push a value stored in the variable `ANumber` onto the stack.

Your algorithm should cope appropriately with any potential errors that might occur.

....

....

....

....

....

....

....

....

(4 marks)

13

6 A computer simulation is to be used to imitate the flow of students through a school canteen. The simulation will be based on a model developed by the school's canteen manager and a Computing student.

6 (a) In the context of simulation, explain what a model is.

....

....

(1 mark)

6 (b) Students must queue at a particular serving point in the canteen if they wish to purchase hot food.

The Computing student intends to represent the queue of students waiting to be served as a dynamic data structure using a linked list.

6 (b) (i) Explain what pointers the student will need to create and what they will be used for.

....

....

....

....

(2 marks)

6 (b) (ii) Teachers are able to bypass the students in the queue by walking past them. However, a teacher may not always go directly to the very front of the queue as it may contain teachers already. In which case, the teacher joins the queue at the point just behind the other teachers.

What type of queue would the Computing student use to represent this situation?

....

(1 mark)

6 (c) The Computing student decides that she will need to use the random number generator in the programming language that she is using to develop the simulation.

Give **one** example of something that she might need to use random numbers for when producing this simulation.

....

....

(1 mark)

5

7 A company is building an e-commerce website. The website will display details of the products that the company sells and allow customers to place orders. Customers must register on the website before they can place an order and each order can be for one or more different products.

The product, customer and order details will be stored in a relational database. It was originally proposed that the following three relations were required:

Product(ProductNumber, ProductPrice, ProductDescription, QuantityInStock)

Order(OrderNumber, OrderDate, CustomerID, OrderingComputerIPAddress, ProductNumber, Quantity)

Customer(CustomerID, CustomerName, Address, Postcode, EmailAddress, PaymentCardNumber)

The computer programmer identified a problem with the Order relation and stated that it should be divided up into two separate relations:

Order(OrderNumber, OrderDate, CustomerID, OrderingComputerIPAddress)

OrderLine(OrderNumber, ProductNumber, Quantity)

7 (a) Describe the problem that the programmer identified with the original Order relation and explain what the cause of this problem was.

....

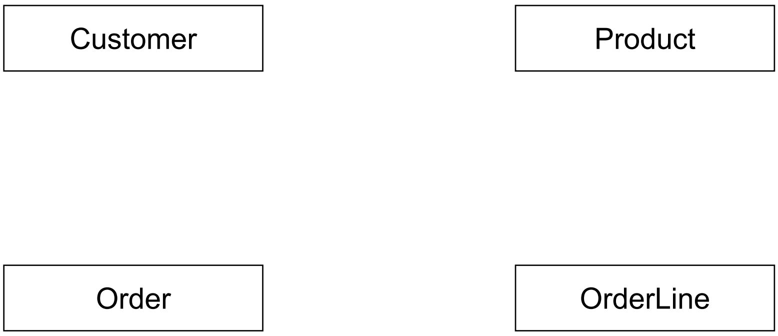
....

....

....

(2 marks)

7 (b) Complete the Entity-Relationship diagram below to show the degree of any **three** relationships that exist between the entities.



(3 marks)

7 (c) Complete the following Data Definition Language (DDL) statement to create the Product relation, including the key field.

CREATE TABLE Product (

....

....

....

....

....

....

....)

(3 marks)

7 (d) The individual web pages that describe each product will be generated dynamically using server-side scripting.

Explain what a *server-side script* is.

....

....

....

....

(2 marks)

Question 7 continues on the next page

Turn over ▸

7 (e) The definitions of the four relations in the database are repeated here so that you can answer the questions on these pages without having to turn back in the question booklet.

> Product(ProductNumber, ProductPrice, ProductDescription, QuantityInStock)
>
> Order(OrderNumber, OrderDate, CustomerID, OrderingComputerIPAddress)
>
> OrderLine(OrderNumber, ProductNumber, Quantity)
>
> Customer(CustomerID, CustomerName, Address, Postcode, EmailAddress, PaymentCardNumber)

A customer can add a product to an order by loading the product's web page, typing the quantity of the product required into a text box and then pressing the order button on the page. The web browser then sends the ProductNumber and Quantity to the web server.

After the user has pressed the order button, the CGI script in **Figure 4** is executed.

Figure 4

```
Line No
  1.   ProdNum = Request("ProductNumber")
  2.   SaleQuant = Request("Quantity")
  3.   ProdDetails = ExecuteSQL("SELECT ProductPrice
           FROM Product WHERE ProductNumber = " + ProdNum)
  4.   ItemPrice = ProdDetails.GetField("ProductPrice")
  5.   TotalPrice = ItemPrice * SaleQuant
  6.   Response.Write ("Total Price is " + TotalPrice)
```

7 (e) (i) Explain the purpose of lines *1.* and *2.* of the CGI script:

```
ProdNum = Request("ProductNumber")
SaleQuant = Request("Quantity")
```

....

....

(2 marks)

7 (e) (ii) Explain the purpose of line *3.* of the CGI script:

```
ProdDetails = ExecuteSQL("SELECT ProductPrice
        FROM Product WHERE ProductNumber = " + ProdNum)
```

....

....

....

(2 marks)

7 (e) (iii) Explain the purpose of line *6.* of the CGI script:

```
Response.Write ("Total Price is " + TotalPrice)
```

....

....

....

(1 mark)

7 (f) A web page is required that will display a summary of the products that are on a particular order.

The summary must include only the ProductNumber, ProductDescription, ProductPrice and the Quantity of the product that has been ordered. These must be displayed in ascending order of ProductNumber.

Write an SQL query that will find the data needed to produce the order summary web page for order number 4013.

....

....

....

....

....

....

....

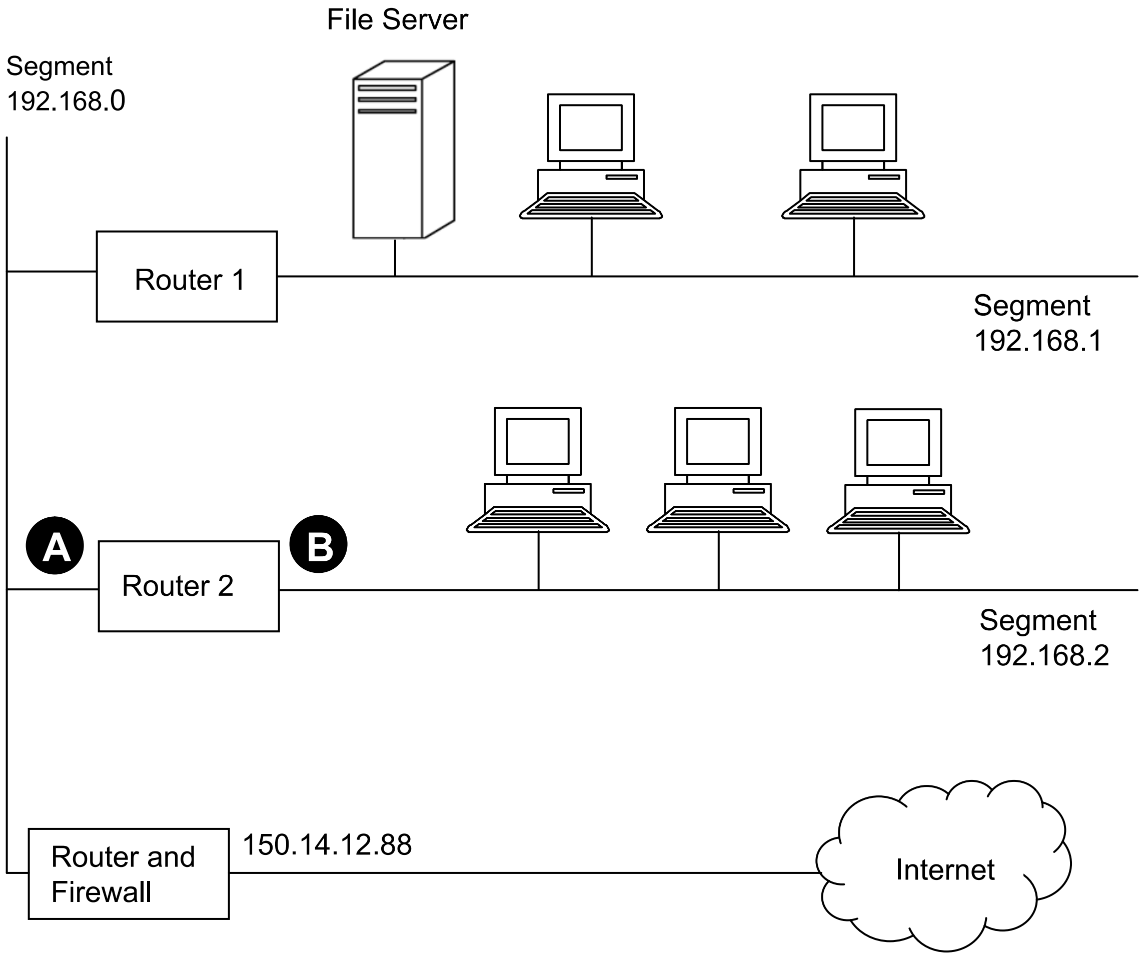
(5 marks)

20

Turn over ▸

8 **Figure 5** shows the topology of a particular computer Local Area Network (LAN) that is divided up into segments.

Figure 5



8 (a) Suggest suitable IP addresses for:

8 (a) (i) the Router 2 port labelled **A** :..

8 (a) (ii) the Router 2 port labelled **B** : ....

(2 marks)

8 (b) When the computers in segment 192.168.2 were configured on the LAN, they were programmed with a subnet mask.

What subnet mask should have been used?

....

(1 mark)

8 (c) The LAN has a bus topology and has been divided into segments.

Explain why the LAN has been segmented.

....

....

....

....

....

(2 marks)

8 (d) Alternatively, the LAN could have been constructed using a star topology.

8 (d) (i) State **one** advantage of using a bus topology and explain how the advantage is achieved.

....

....

....

(1 mark)

8 (d) (ii) State **one** advantage of using a star topology and explain how the advantage is achieved.

....

....

....

(1 mark)

Question 8 continues on the next page

Turn over ▶

8 (e) Discuss the security threats that the network manager will need to deal with because the LAN is connected to the Internet, together with how these may be dealt with.

In your answer you will also be assessed on your ability to use good English, and to organise your answer clearly and coherently in complete sentences, using specialist vocabulary where appropriate.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

(6 marks)

13

9 Regular expressions can be used to search for strings.

9 (a) For each of the following regular expressions, describe the set of strings that they would find.

9 (a) (i) `a+b`

....

....

(1 mark)

9 (a) (ii) `a?b`

....

....

(1 mark)

9 (a) (iii) `(ab)*`

....

....

(1 mark)

9 (b) Write regular expressions that match:

9 (b) (i) either `Clare` or `Claire`.

....

(1 mark)

9 (b) (ii) any non-empty string that:

- starts with `10`
- has zero or more occurrences of any combination of `0` or `1` in the middle
- ends with `01`

Example strings that the expression should match are `1001`, `100010101`, `101111010101001`.

....

(2 marks)

6

10 A home desktop computer is connected to a number of peripherals including a printer and a keyboard. It is also connected to the Internet and to a wired Local Area Network (LAN).

10 (a) The keyboard is connected to the computer using a serial connection at a speed of 9,600 bits per second with a baud rate of 9,600 baud.

Explain what is meant by *baud rate*.

....

....

(1 mark)

10 (b) A printer is connected to the same computer using a faster serial connection at a speed of 128,000 bits per second and a baud rate of 64,000 baud.

10 (b) (i) Explain how it is possible for the number of bits transmitted per second to be higher than the baud rate.

....

....

(1 mark)

10 (b) (ii) When the computer has a document to print, the computer and printer must perform a handshake. **Table 2** shows the steps involved in a handshake to send a single character along the serial link to the printer.

Write labels for the missing steps in the **Data/Request Sent** column of **Table 2**, assuming that the printer is able to accept the character.

Table 2

Step	**Direction**	**Data/Request Sent**
1	Computer → Printer	Is printer ready to receive data?
2	Computer ← Printer	
3	Computer → Printer	
4	Computer ← Printer	Printer receiving data
5	Computer → Printer	Sending has ended
6	Computer ← Printer	

(3 marks)

10 (c) The computer is connected to a small LAN using a wired baseband connection and to the Internet using a broadband connection.

Explain the difference between baseband and broadband connections and justify why the LAN connection is baseband whereas the Internet connection is broadband.

In your answer you will also be assessed on your ability to use good English, and to organise your answer clearly and coherently in complete sentences, using specialist vocabulary where appropriate.

....

....

....

....

....

....

....

....

....

....

....

....

(4 marks)

9

Turn over for the next question

11 A particular Turing machine has states S_1, S_2, S_3 and S_4. S_1 is the start state and S_4 is the stop state. The machine uses one tape which is infinitely long in one direction to store data. The machine's alphabet is 1, □. The symbol □ is used to indicate a blank cell on the tape.

The transition rules for this Turing machine can be expressed as a transition function δ. Rules are written in the form:

$$\delta(\text{Current State, Input Symbol}) = (\text{Next State, Output Symbol, Movement})$$

So, for example, the rule:

$$\delta\,(S_1, 1) = (S_1\,, 1, \rightarrow)$$

means:

IF the machine is currently in state S_1 AND the input symbol read from the tape is 1
THEN the machine should remain in state S_1, write a 1 to the tape and move the read/write head one cell to the right

The machine's transition function, δ, is defined by:

$$\begin{aligned}
\delta\,(S_1, 1) &= (S_1, 1, \rightarrow) \\
\delta\,(S_1, \square) &= (S_2, \square, \leftarrow) \\
\delta\,(S_2, 1) &= (S_3, \square, \leftarrow) \\
\delta\,(S_3, 1) &= (S_4, \square, \leftarrow)
\end{aligned}$$

11 (a) The Turing machine is carrying out a computation. The machine starts in state S_1 with the string `1111` on the tape. All other cells contain the blank symbol, □. The read/write head is positioned at the leftmost `1`, as indicated by the arrow.

			1	1	1	1			
			↑						

………. Current State: S_1

Trace the computation of the Turing machine, using the transition function δ. Show the contents of the tape, the current position of the read/write head and the current state as the input symbols are processed.

………. Current State: □

………. Current State: □

………. Current State: □

………. Current State: □

………. Current State: □

………. Current State: □

………. Current State: □

(6 marks)

11 (b) Explain what this Turing machine does.

....

....

(1 mark)

11 (c) Explain what a *Universal Turing machine* is.

....

....

....

....

....

(2 marks)

9

END OF QUESTIONS