AQA

SPECIMEN MATERIAL

A-level
COMPUTER SCIENCE
(7517/1A/1B/1C/1D/1E)

Paper 1

Specimen 2015 am/pm Time allowed: 2 hours 30 minutes

Materials
- For this paper you must have access to:
  - a computer
  - a printer
  - appropriate software.
- An electronic version of the **Skeleton Program** and **Data File**.
- A hard copy of the **Preliminary Material**.

Instructions
- Type the information required on the front of your **Electronic Answer Document**.
- Enter your answers into the **Electronic Answer Document**.
- Answer **all** questions.
- Before the start of the examination make sure your **centre number**, **candidate name** and **candidate number** are shown clearly in the footer of every page of your **Electronic Answer Document** (not the front cover).
- Tie together all your printed **Electronic Answer Document** pages and hand them to the invigilator.

Information
- The marks for questions are shown in brackets.
- The maximum mark for this paper is 100.
- No extra time is allowed for printing and collating.
- The question paper is divided into **four** sections.
- You are **not** allowed to use a calculator.

Advice
- You are advised to spend time on each section as follows:
  Section A – 55 minutes
  Section B – 20 minutes
  Section C – 15 minutes
  Section D – 60 minutes.
- Save your work at regular intervals.

Section A

You are advised to spend no more than **55 minutes** on this section.

Enter your answers to **Section A** in your Electronic Answer Document.

You **must save** this document at regular intervals.

---

0 1 The famous detective John Stout was called in to solve a perplexing murder mystery. He determined the following facts.

(a) Nathan, the murdered man, was killed by a blow on the head.
(b) Either Suzanne or Martin was in the dining room at the time of the murder.
(c) If Peter was in the kitchen at the time of the murder, then Ian killed Nathan using poison.
(d) If Suzanne was in the dining room at the time of the murder, then Steve killed Nathan.
(e) If Peter was not in the kitchen at the time of the murder, then Martin was not in the dining room when the murder was committed.
(f) If Martin was in the dining room at the time the murder was committed, then Paul killed Nathan.
(g) If Kevin was in the hall at the time of the murder, then Suzanne killed Nathan by a blow to the neck with a saucepan.

0 1 . 1 Who murdered Nathan?

A Paul
B Steve
C Suzanne
D Ian
E It is not possible for John Stout to solve the crime.

Write the letter corresponding to the correct answer in the box provided in your Electronic Answer Document.

[1 mark]

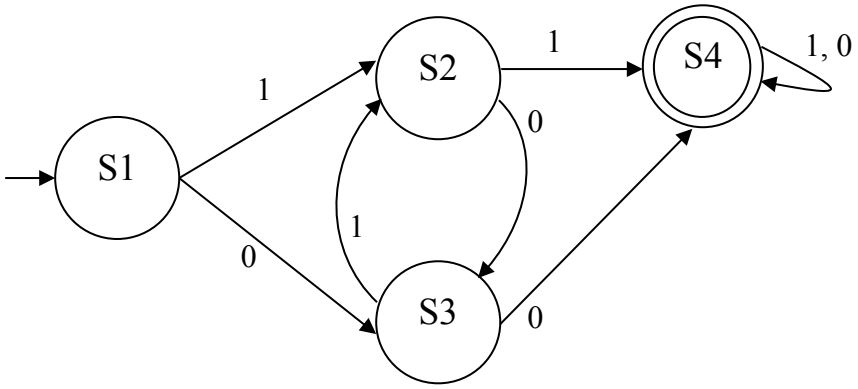
0 1 . 2 Explain how you know your answer to **0 1 . 1** is correct.

[2 marks]

Use the space below for rough working, then write your answer in your Electronic Answer Document.

0 2 A finite state machine (FSM) can be used to define a language: a string is allowed in a language if it is accepted by the FSM that represents the rules of the language. **Figure 1** shows the state transition diagram for an FSM.

Figure 1



An FSM can be represented as a state transition diagram or as a state transition table. **Table 1** is an incomplete state transition table for **Figure 1**.

0 2 . 1 Complete **Table 1** and copy the table into the Electronic Answer Document.

Table 1

Original state	Input	New state
S3		
S3		

[1 mark]

Any language that can be defined using an FSM can also be defined using a regular expression.

The FSM in **Figure 1** defines the language that allows all strings containing at least, either two consecutive 1s or two consecutive 0s.

The strings 0110, 00 and 01011 are all accepted by the FSM and so are valid strings in the language.

The strings 1010 and 01 are not accepted by the FSM and so are not valid strings in the language.

0 2 . 2 Write a regular expression that is equivalent to the FSM shown in **Figure 1**.

[3 marks]

Question 2 continues on the next page

Backus-Naur Form (BNF) can be used to define the rules of a language.

Figure 2 shows an attempt to write a set of BNF production rules to define a language of full names.

Figure 2

Note: underscores (_) have been used to denote spaces.
Note: rule numbers have been included but are not part of the BNF rules.

Rule number	
1	<fullname> ::= <title>_<name>_<endtitle> | <name> | <title>_<name> | <name>_<endtitle>
2	<title> ::= MRS | MS | MISS | MR | DR | SIR
3	<endtitle> ::= ESQUIRE | OBE | CBE
4	<name> ::= <word> | <name>_<word>
5	<word> ::= <char><word>
6	<char> ::= A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z

BNF can be used to define languages that are not possible to define using regular expressions. The language defined in **Figure 2** could not have been defined using regular expressions.

0 2 . 3 Complete **Table 2** below by writing either a '**Y**' for **Yes** or '**N**' for **No** in each row.

Table 2

Rule number (given in Figure 2)	Could be defined using a regular expression
1	
2	
3	
4	
5	
6	

Copy your answer in **Table 2** into the Electronic Answer Document.

[1 mark]

There is an error in rule 5 in **Figure 2** which means that no names are defined by the language.

0 2 . 4 Explain what is wrong with the production rule and rewrite the production rule so that the language does define some names – the names 'BEN D JONES', 'JO GOLOMBEK' and 'ALULIM' should all be defined.

[2 marks]

Turn over for the next question

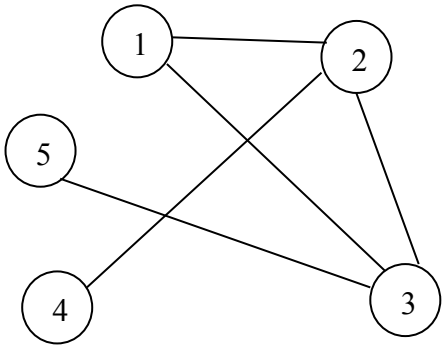
0 3 The Cat transportation company (CTC) is a business that specialises in preparing cats for cat shows.

They need to take five cats to the AQA cat show. They will transport the cats in their van. CTC owns only one van.

They cannot put all the cats in their van at the same time because some of the cats get stressed when in the company of some of the other cats. The cats would not therefore arrive in top condition for the cat show if they were all in the van at the same time.

The graph in **Figure 3** shows the relationships between the five cats (labelled 1 to 5). If there is an edge between two cats in the graph then they **cannot** travel in the van together at the same time.

Figure 3



0 3 . 1 Explain why the graph in **Figure 3** is **not** a tree.

[1 mark]

0 3 . 2 Represent the graph shown in **Figure 3** as an adjacency list by completing **Table 3**.

Complete **Table 3** and copy the table into the Electronic Answer Document.

[2 marks]

Table 3

Vertex (in Figure 3)	Adjacent vertices
1	
2	
3	
4	
5	

Table 4 shows how the graph in **Figure 3** can be represented as an adjacency matrix.

Table 4

Vertex (in Figure 3)	1	2	3	4	5
1	0	1	1	0	0
2	1	0	1	1	0
3	1	1	0	0	1
4	0	1	0	0	0
5	0	0	1	0	0

0 3 . 3 Explain the circumstances in which it is more appropriate to represent a graph using an adjacency list instead of an adjacency matrix.

[2 marks]

Question 3 continues on the next page

Figure 4 shows an algorithm, written in pseudo-code, that CTC use.

Figure 4

```
NoOfCats ← 5
Cat[1] ← 1
FOR A ← 2 TO NoOfCats
  B ← 1
  C ← 1
  WHILE B < A DO
    IF M[A, B] = 1
      THEN
        IF Cat[B] = C
          THEN
            B ← 1
            C ← C + 1
          ELSE B ← B + 1
        ENDIF
      ELSE B ← B + 1
    ENDIF
  ENDWHILE
  Cat[A] ← C
ENDFOR
```

The two-dimensional array, `M`, is used to store the adjacency matrix shown in **Table 4**.

0 3 . 4 Complete **Table 5** to show the result of tracing the algorithm in **Figure 4**.

[6 marks]

Copy your answer in **Table 5** into the Electronic Answer Document.

Table 5

				Cat				
NoOfCats	**A**	**B**	**C**	**1**	**2**	**3**	**4**	**5**

0 3 . 5 Explain the purpose of the algorithm in **Figure 4**.

[1 mark]

After a cat show, CTC needs to return the cats to their owners. They can have all the cats in the van at the same time because the show is now finished.

CTC likes to plan the return journey so that the shortest possible distance is travelled by the van. This is an example of an intractable problem.

0 3 . 6 What is meant by an intractable problem?

[2 marks]

0 3 . 7 What approach might a programmer take if asked to solve an intractable problem?

[2 marks]

0 4 **Figure 5** shows an incomplete algorithm for a binary search.

Figure 5

```
PROCEDURE BSearch(List, F, L,
ItemToFind)
  Found ← False
  Failed ← (1).....
  WHILE NOT Failed AND NOT Found
    M ← (F + L) DIV 2
    IF List[M] = ItemToFind
      THEN Found ← True
      ELSE
        IF F >= L
          (2).....
          ELSE
            IF List[M] > ItemToFind
              THEN (3).....
              ELSE F ← M + 1
            ENDIF
        ENDIF
    ENDIF
  ENDWHILE
  IF Found = True
    THEN OUTPUT "Item is in list"
    ELSE OUTPUT "Item is not in list"
  ENDIF
ENDPROCEDURE
```

The `DIV` operator calculates the whole number result of integer division. For example, `15 DIV 4 = 3`, `17 DIV 4 = 4`.

0 4 . 1 What code should be added at position **(1)** in **Figure 5**?

[1 mark]

0 4 . 2 What code should be added at position **(2)** in **Figure 5**?

[1 mark]

0 4 . 3 What code should be added at position **(3)** in **Figure 5**?

[2 marks]

Table 6 contains a list of orders of time complexity (in no particular order).

Table 6

Order of time complexity
O(1)
$O(n^2)$
O(log n)
$O(k^n)$
O(n)

Which of the orders of time complexity given in **Table 6:**

0 4 . 4 could be the time complexity of an intractable problem?

[1 mark]

0 4 . 5 is the time complexity for a binary search?

[1 mark]

0 4 . 6 is the time complexity for getting the first item in a list?

[1 mark]

0 4 . 7 is the time complexity for a linear-search algorithm?

[1 mark]

0 4 . 8 Explain why a linear-search has the order of time complexity given in your answer to question **0 4 . 7**.

[2 marks]

0 5 Convert the following Reverse Polish Notation expressions to their equivalent infix expressions.

0 5 . 1 `3 4 *`

[1 mark]

0 5 . 2 `12 8 + 4 *`

[1 mark]

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

0 5 . 3 State **one** advantage of Reverse Polish Notation over infix notation.

[1 mark]

END OF SECTION A

Section B begins on page 14

There are no questions printed on this page

Turn over for Section B

Section B

You are advised to spend no more than **20 minutes** on this section.

Enter your answers to **Section B** in your Electronic Answer Document.

You **must save** this document at regular intervals.

The question in this section asks you to write program code **starting from a new program/project/file**.

- Save your program/project/file in its own folder/directory.
- Save your program/project/file at regular intervals.

---

0 6 Create a folder/directory called **Question6** for your new program.

One method for converting a decimal number into binary is to repeatedly divide by 2 using integer division. After each division is completed, the remainder is output and the integer result of the division is used as the input to the next iteration of the division process. The process repeats until the result of the division is 0.

Outputting the remainders in the sequence that they are calculated produces the binary digits of the equivalent binary number, but in reverse order.

For example, the decimal number 210 could be converted into binary as shown in **Figure 7**.

Figure 7

```
210 ÷ 2 =  105   remainder 0
105 ÷ 2 =  52    remainder 1
 52 ÷ 2 =  26    remainder 0
 26 ÷ 2 =  13    remainder 0
 13 ÷ 2 =  6     remainder 1
  6 ÷ 2 =  3     remainder 0
  3 ÷ 2 =  1     remainder 1
  1 ÷ 2 =  0     remainder 1
```

The sequence 0, 1, 0, 0, 1, 0, 1, 1 which would be output by this process is the reverse of the binary equivalent of 210 which is 11010010.

What you need to do

Task 1
Write a program that will perform the conversion process described above. The program should display a suitable prompt asking the user to input a decimal number to convert and then output the bits of the binary equivalent of the decimal number in reverse order.

Task 2
Improve the program so that the bits are output in the correct order, e.g. for 210 the output would be 11010010.

Task 3
Test the program works by entering the value 210.

Save the program in your new **Question6** folder/directory.

Evidence that you need to provide
Include the following in your Electronic answer document.

0 6 . 1 Your PROGRAM SOURCE CODE after you have completed both **Task 1** and **Task 2**.

If you complete **Task 1** but do not attempt **Task 2** then a maximum of 9 marks will be awarded.

[12 marks]

0 6 . 2 SCREEN CAPTURE(S) for the test showing the output of the program when 210 is entered.

The marks for this test will be awarded whether the binary digits are output in reverse order or in the correct order.

[2 marks]

END OF SECTION B

Turn over for Section C

Section C

You are advised to spend no more than **15 minutes** on this section.

Type your answers to **Section C** in your Electronic Answer Document.

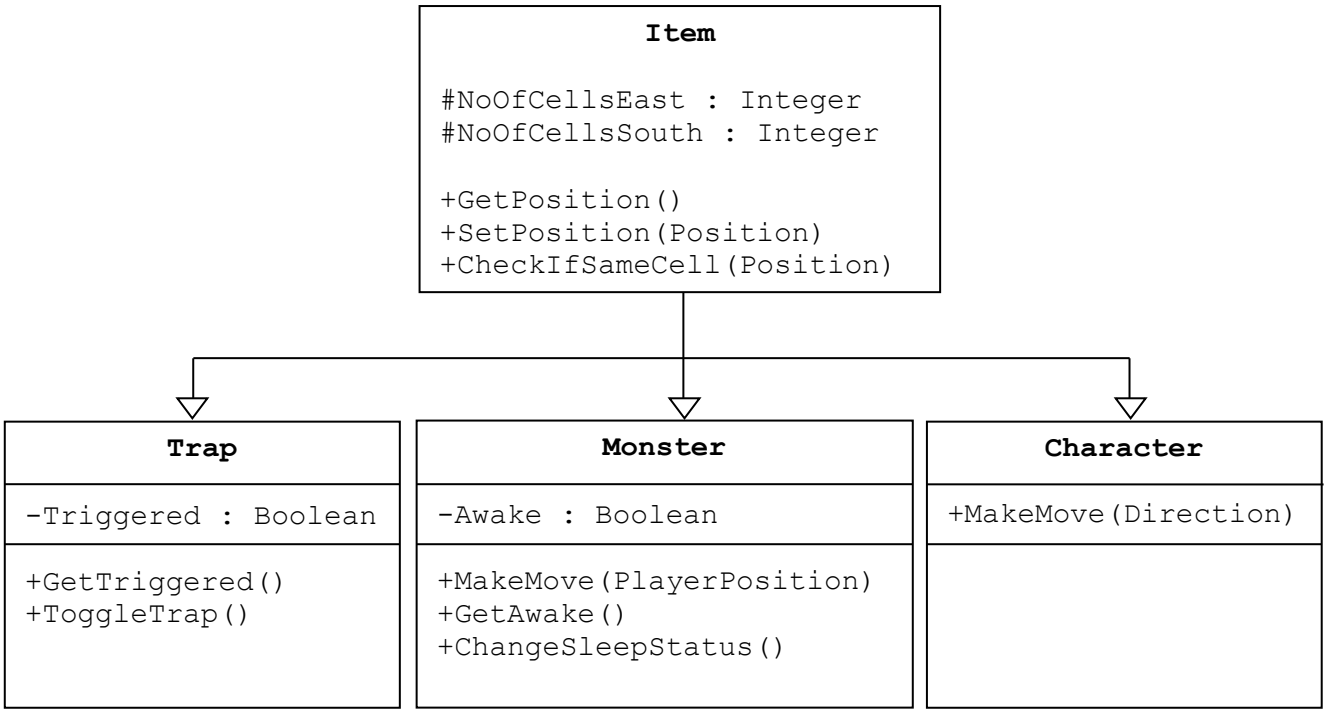
You **must save** this document at regular intervals.

These questions refer to the **Preliminary Material** and require you to load the **Skeleton Program**, but do not require any additional programming.

Refer **either** to the **Preliminary Material** issued with this question paper **or** your electronic copy.

0 7 The class diagram in **Figure 8** is an attempt to represent the relationships between some of the classes in the MONSTER! Game.

Figure 8



0 7 . 1 Explain what errors have been made in **Figure 8**.

[2 marks]

0 7 . 2 Give an example of instantiation from the **Skeleton Program**.

[1 mark]

0 7 . 3 State the name of an identifier for an array variable.

[1 mark]

0 7 . 4 State the name of an identifier for a subclass.

[1 mark]

0 7 . 5 State the name of an identifier for a variable that is used to store a whole number.

[1 mark]

0 7 . 6 State the name of an identifier for a class that uses composition.

[1 mark]

Look at the `GetNewRandomPosition` subroutine in the `Game` class in the **Skeleton Program**.

0 7 . 7 Explain why the generation of a random position needs to be inside a repetition structure.

[1 mark]

0 7 . 8 Look at the `Game` class in the **Skeleton Program**.

Why has a named constant been used instead of a numeric value?

[2 marks]

0 7 . 9 Describe the changes that would need to be made to the `Game` class to add a third trap to the cavern. The third trap should have exactly the same functionality as the other two traps. You do **not** need to describe the changes that would need to be made to the `SetUpGame` subroutine.

[2 marks]

END OF SECTION C

Turn over for Section D

Section D

You are advised to spend no more than **60 minutes** on this section.

Type your answers to **Section D** in your Electronic Answer Document.

You **must save** this document at regular intervals.

These questions require you to load the **Skeleton Program** and to make programming changes to it.

0 8 This question refers to the subroutines `CheckValidMove` and `Play` in the `Game` class.

The **Skeleton Program** currently does not make all the checks needed to ensure that the move entered by a player is an allowed move. It should not be possible to make a move that takes a player outside the 7 × 5 cavern grid.

The **Skeleton Program** needs to be adapted so that it prevents a player from moving west if they are at the western end of the cavern.

The subroutine `CheckValidMove` needs to be adapted so that it returns a value of FALSE if a player attempts to move west when they are at the western end of the cavern.

The subroutine `Play` needs to be adapted so that it displays an error message to the user if an illegal move is entered. The message should state "`That is not a valid move, please try again`".

Evidence that you need to provide
Include the following in your Electronic Answer Document.

0 8 . 1 Your amended PROGRAM SOURCE CODE for the subroutine `CheckValidMove`.

[3 marks]

0 8 . 2 Your amended PROGRAM SOURCE CODE for the subroutine `Play`.

[2 marks]

0 8 . 3 SCREEN CAPTURE(S) for a test run showing a player trying to move west when they are at the western end of the cave.

[1 mark]

0 9 This question will extend the functionality of the game.

The game is to be altered so that there is a new type of enemy: a sleepy enemy. A sleepy enemy is exactly the same as a normal enemy, except that after making four moves it falls asleep again.

Task 1
Create a new class called `SleepyEnemy` that inherits from the `Enemy` class.

Task 2
Create a new integer attribute in the `SleepyEnemy` class called `MovesTillSleep`.

Task 3
Create a new public subroutine in the `SleepyEnemy` class called `ChangeSleepStatus`. This subroutine should override the `ChangeSleepStatus` subroutine from the `Enemy` class. The value of `MovesTillSleep` should be set to 4 in this subroutine.

Task 4
Create a new public subroutine in the `SleepyEnemy` class called `MakeMove`. This subroutine should override the `MakeMove` subroutine from the `Enemy` class. When called this subroutine should reduce the value of `MovesTillSleep` by 1 and then send the monster to sleep if `MovesTillSleep` has become equal to 0.

Task 5
Modify the `Game` class so that the Monster object is of type `SleepyEnemy` (instead of `Enemy`).

Task 6
Check that the changes you have made work by conducting the following test:

- play the training game
- move east
- move east
- move south.

Evidence that you need to provide
Include the following in your Electronic Answer Document.

0 9 . 1 Your PROGRAM SOURCE CODE for the new `SleepyEnemy` class.

[8 marks]

0 9 . 2 SCREEN CAPTURE(S) showing the requested test.

[2 marks]

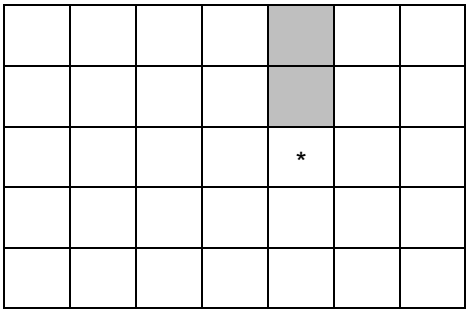
1 0 This question refers to the `Game` and `Character` classes and will extend the functionality of the game.

The game should be altered so that once per game the player can shoot an arrow instead of making a move in the cavern. The arrow travels in a straight line, in a direction of the player's choice, from the cell the player is in to the edge of the cavern. If the arrow hits the monster then the player wins the game and a message saying that they have shot the monster should be displayed.

For this question you are **only** required to extend the program so that it checks if the monster is hit by the arrow when the user chooses to shoot an arrow northwards. However, the user should be able to select any of the four possible directions.

In **Figure 9**, the two shaded cells show the cells which, if the monster is in one of them, would result in the player winning the game, as long as the player is in the cell five to the east and three to the south and chooses to shoot an arrow northwards.

Figure 9



Task 1
Modify the `DisplayMoveOptions` subroutine in the `Game` class so that the option to enter A to shoot an arrow is added to the menu.

Task 2
Create a new Boolean attribute called `HasArrow` in the `Character` class.

The value of `HasArrow` should be set to `True` when a new object of class `Character` is instantiated.

Task 3
Create a new public subroutine called `GetHasArrow` in the `Character` class that returns the value of the `HasArrow` attribute to the calling routine.

Task 4
Modify the `CheckValidMove` subroutine in the `Game` class so that:

- it is a valid move if A is selected and the player does have an arrow
- it is not a valid move if A is selected and the player does not have an arrow.

Task 5
Create a new public subroutine called `GetArrowDirection` in the `Character` class.

This subroutine should return a character to the calling routine.

The user should be asked in which direction they would like to shoot an arrow (N, S, E or W) and the value entered by the user should be returned to the calling routine.

If an invalid direction is entered then the user should be repeatedly asked to enter a new direction, until a valid direction is entered.

The value of `HasArrow` should then be changed to FALSE.

Task 6
Modify the `Play` subroutine in the `Game` class so that if the move chosen by the user is not M it then checks if the move chosen is A.

If the move chosen was A, then there should be a call to the player's `GetArrowDirection` subroutine. If the user chooses a direction of N then the program should check to see if the monster is in one of the squares directly north of the player's current position. If it is then a message saying "`You have shot the monster and it cannot stop you finding the flask`" should be displayed. The value of `FlaskFound` should then be set to TRUE.

After the arrow has been shot, if the monster is still alive and awake, it is now the monster's turn to move, the player should remain in the same cell as they were in before the arrow was shot.

There is **no** need to write any code that checks if the monster has been shot when the player chooses to shoot either to the east, to the west or to the south.

Task 7: test 1
Test that the changes you have made work by conducting the following test:

- play the training game
- shoot an arrow
- choose a direction of N for the arrow.

Task 8: test 2
Test that the changes you have made work by conducting the following test:

- play the training game
- move east
- shoot an arrow
- choose a direction of N for the arrow
- shoot an arrow.

Question 10 continues on the next page

Evidence that you need to provide
Include the following in your Electronic Answer Document.

1 0 . 1 Your amended PROGRAM SOURCE CODE for the subroutine `DisplayMoveOptions`.

[1 mark]

1 0 . 2 Your amended PROGRAM SOURCE CODE for the subroutine `CheckValidMove`.

[2 marks]

1 0 . 3 Your amended PROGRAM SOURCE CODE for the class `Character`.

[8 marks]

1 0 . 4 Your amended PROGRAM SOURCE CODE for the subroutine `Play`.

[6 marks]

1 0 . 5 SCREEN CAPTURE(S) showing the results of **Test 1**.

[1 mark]

1 0 . 6 SCREEN CAPTURE(S) showing the results of **Test 2**.

[1 mark]

END OF QUESTIONS

There are no questions printed on this page

DO NOT WRITE ON THIS PAGE
ANSWER IN THE SPACES PROVIDED

There are no questions printed on this page

DO NOT WRITE ON THIS PAGE
ANSWER IN THE SPACES PROVIDED