General Certificate of Education Advanced Subsidiary Examination April 2015 (Based on 2015 Capture the Sarrum skeleton Code)

**Computing COMP1 AS Mock**

**For this paper you must have:**

! access to the Electronic Answer Document

! a copy of the *Preliminary Material*. You must **not** use a calculator.

**Time allowed**

! 1 Hour 30 Mins

**Instructions**

! Type your answers into the Electronic Answer Document.

! Enter the information required on the front of your Electronic Answer Document.

! Answer **all** questions.

! You will need access to:

– a computer

– a printer

– appropriate software

– the electronic version of the Skeleton Program.

! Before the start of the examination make sure your **Centre Number, Candidate Name** and **Number**

are shown clearly in the footer of the Electronic Answer Document (not the front over).

**Information**

! The marks for questions are shown in brackets.

! The maximum mark for this paper is

! No extra time is allowed for printing and collating.

! The question paper is divided into four sections.

You are advised to spend time on each section as follows:

Section A - 20 minutes

Section B - 20 minutes

Section C - 10 minutes

Section D - 40 minutes.

**At the end of the examination**

! Tie together all your printed Electronic Answer Document pages and hand them to the invigilator.

**Total Score 66**

M/Jun10/COMP1

**COMP1**

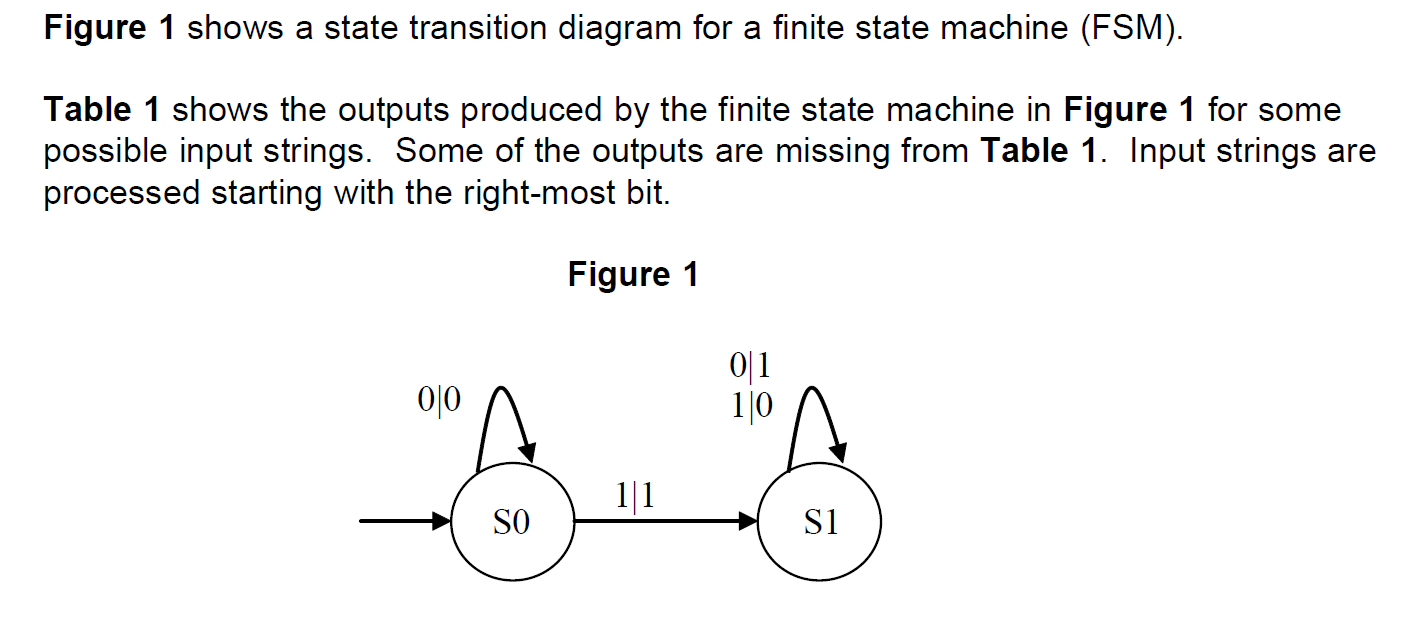
**Section A**

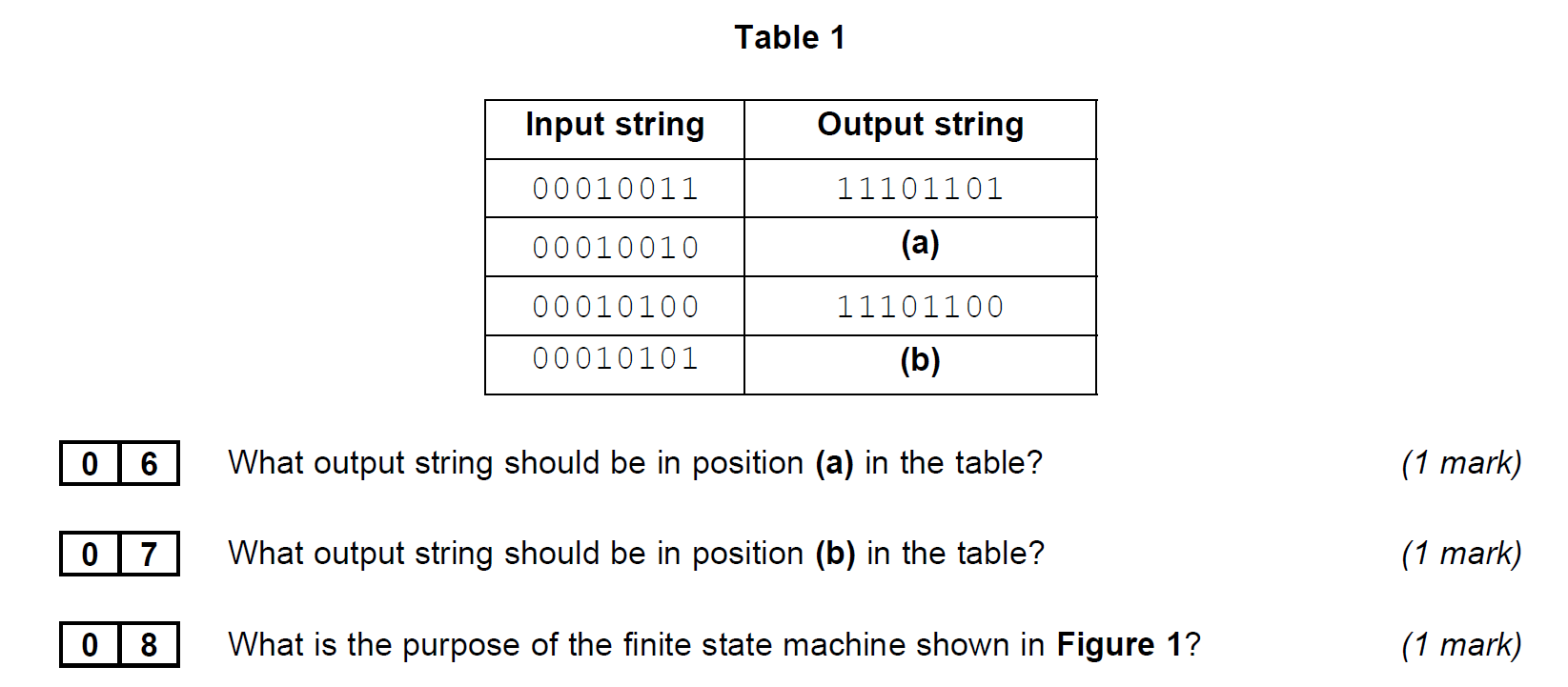
You are advised to spend no more than **20 minutes** on this section.

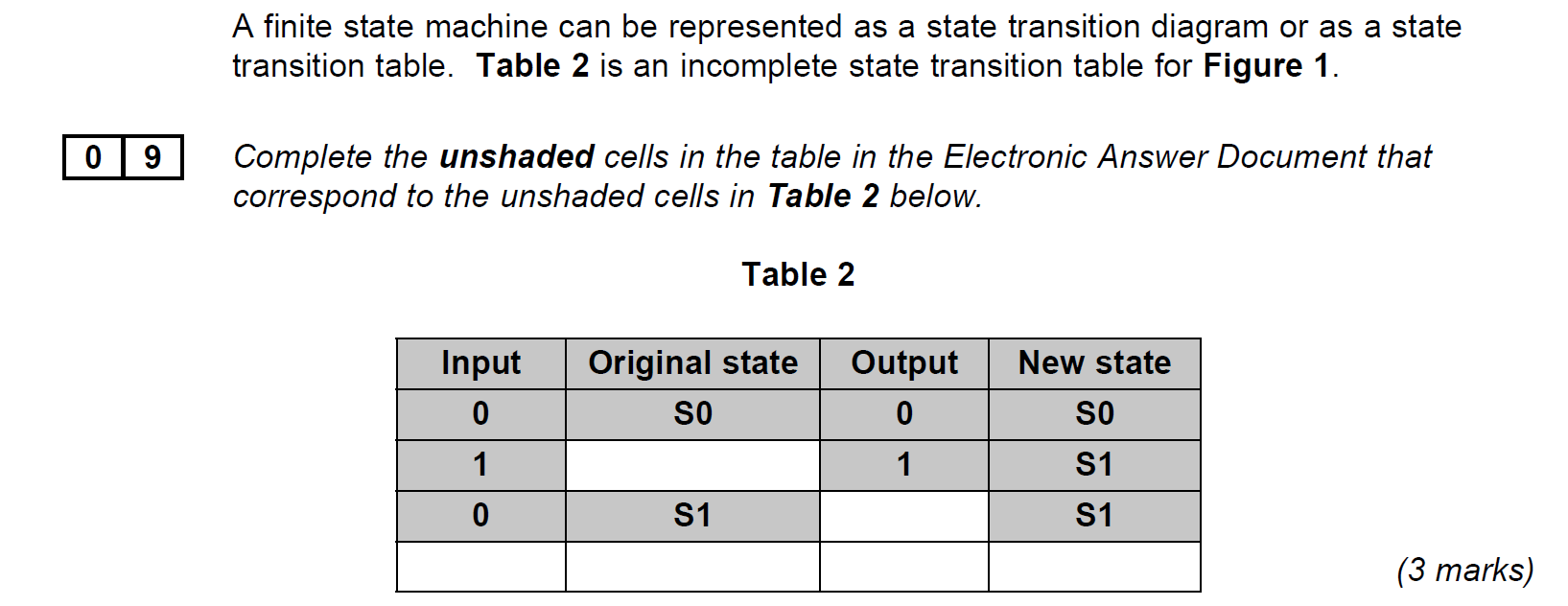
Type your answers to **Section A** in your Electronic Answer Document.

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

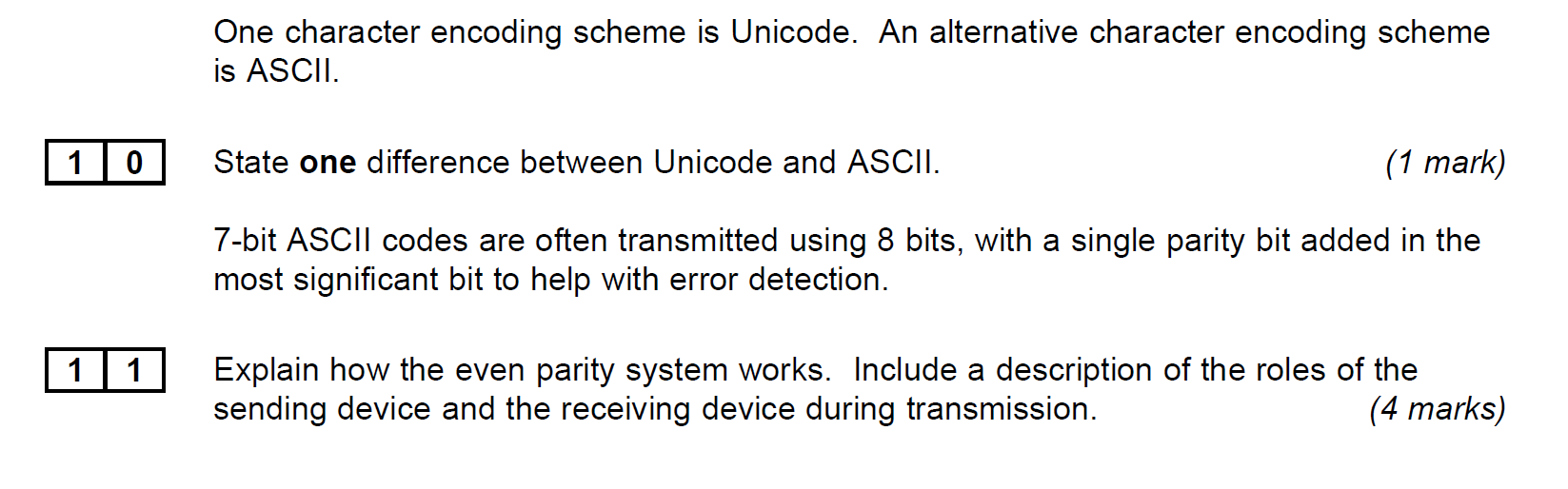
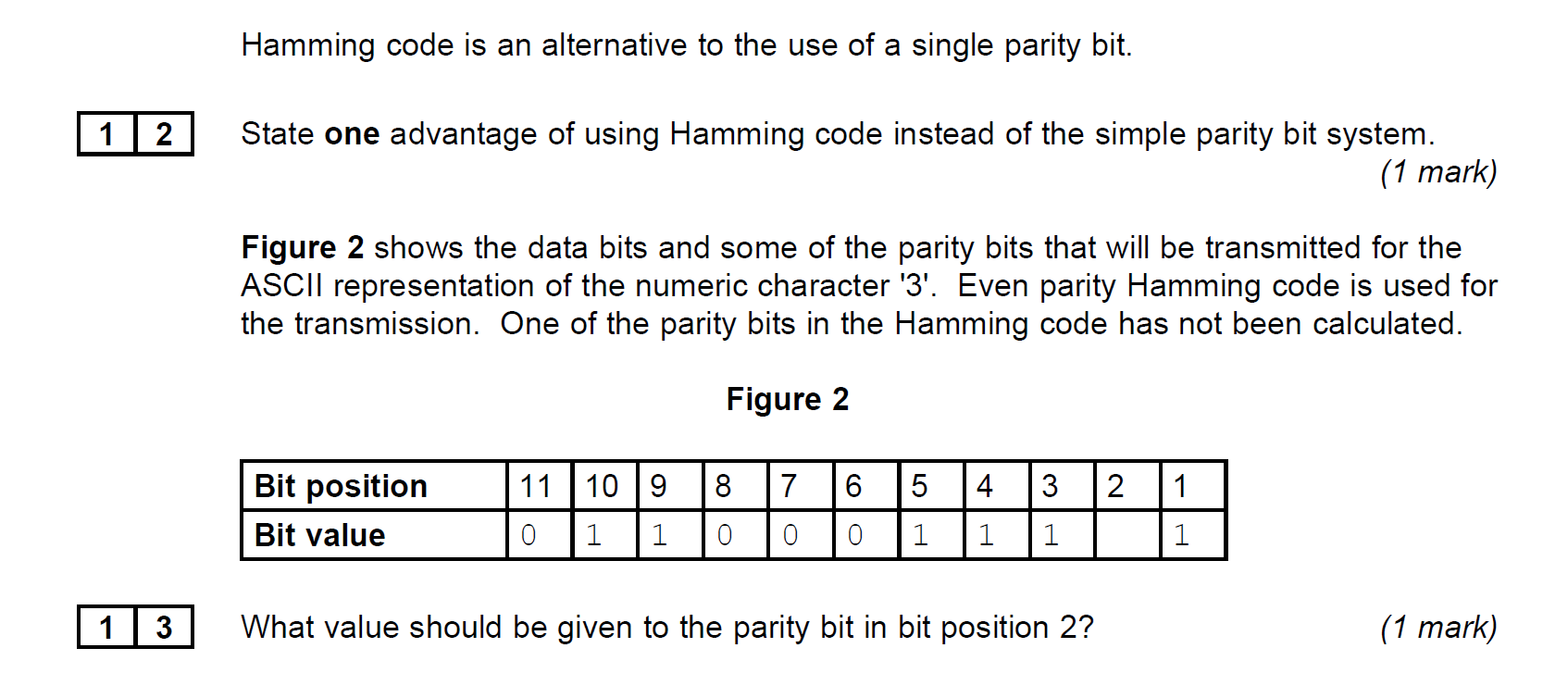
**Question 1**

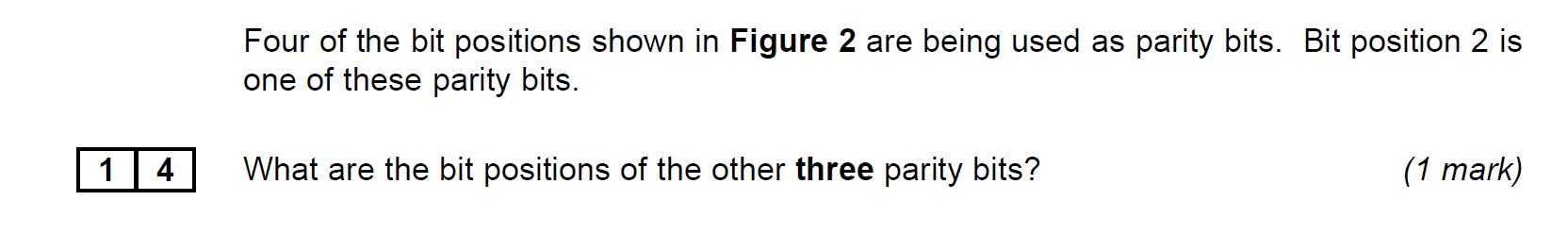




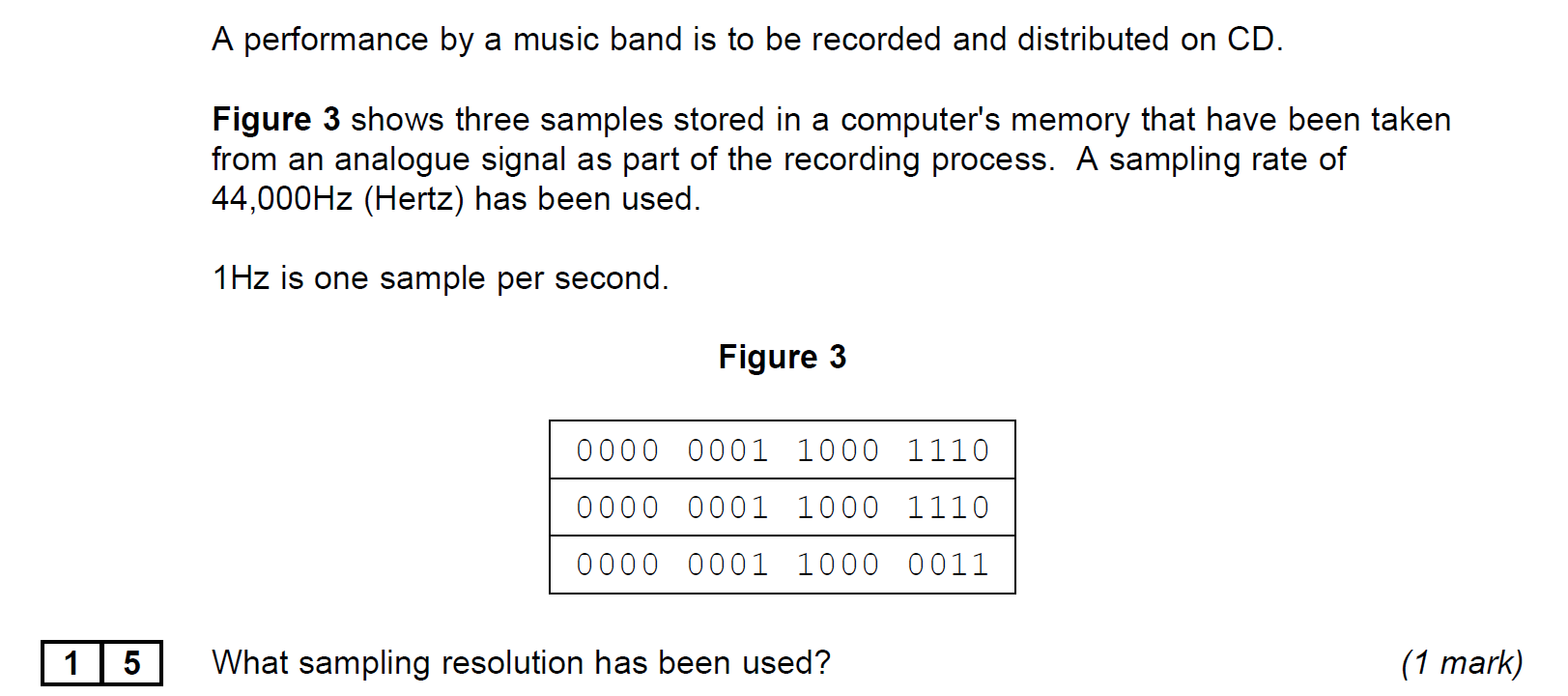


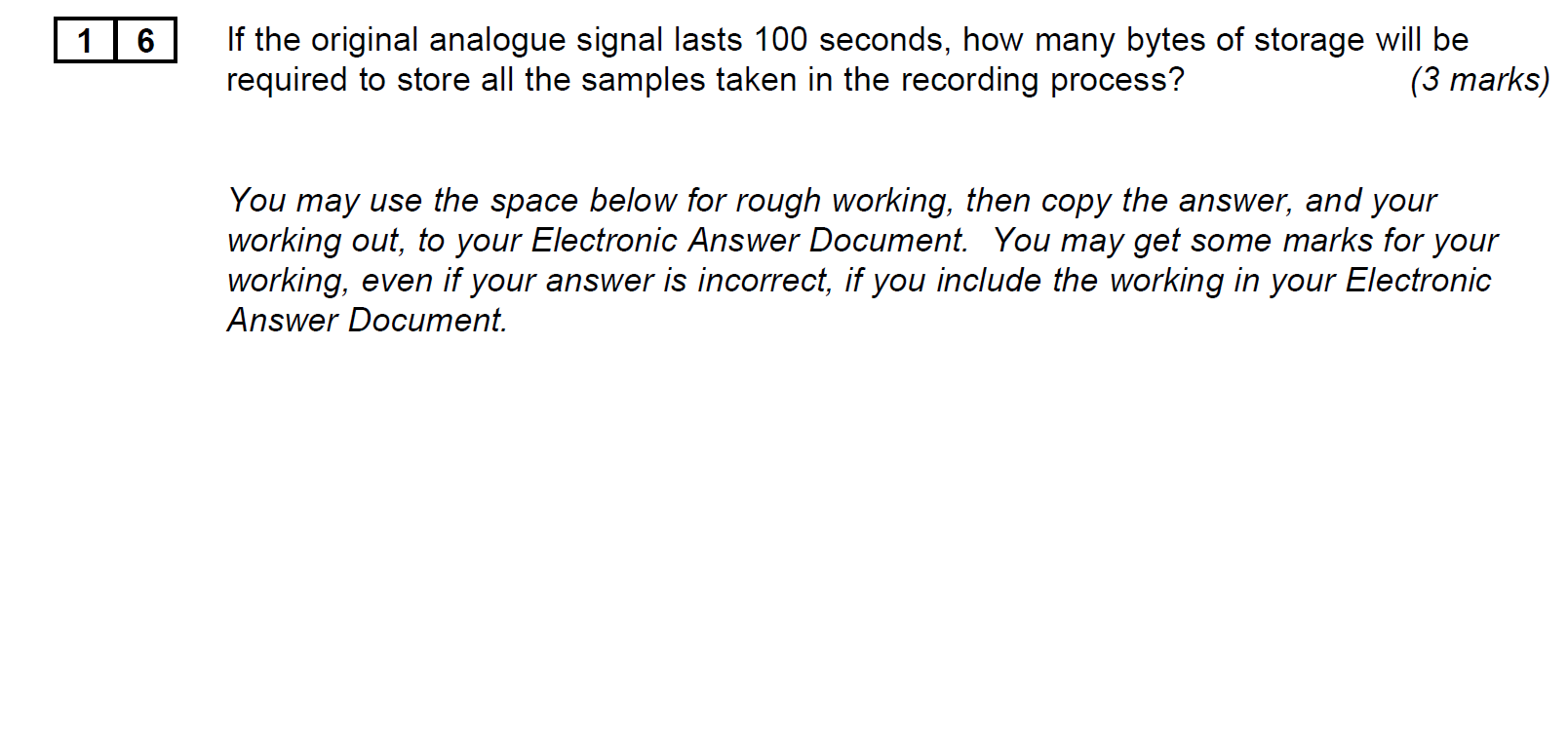
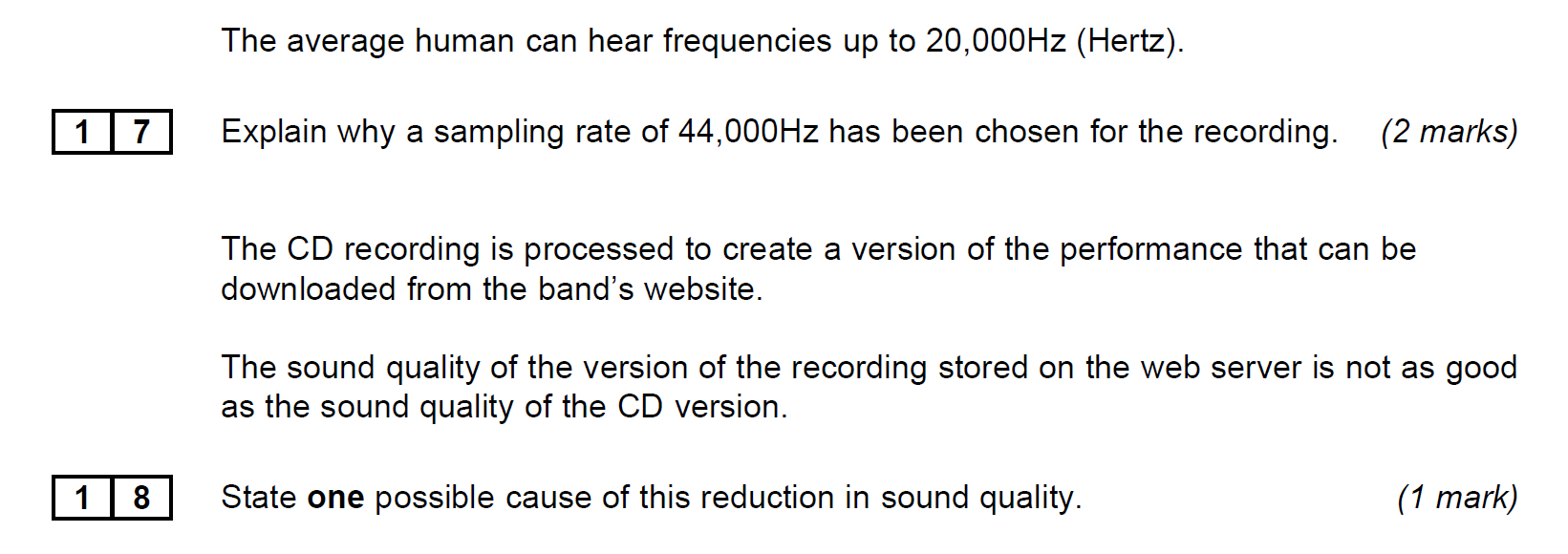
**Question 2**



**Question 3**



**Turn over for the next section**

**Section B**

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

Type 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/Þ le in its own folder/directory.

• You are advised to save your program at regular intervals.

**Question 4**

Create a folder/directory **Question4** for your new program.

The variable table, the table below, and the Structured English algorithm below, describe a simplified version of a game of guessing Heads or Tails by flipping a coin.

|  |  |  |
| --- | --- | --- |
| **Identifier** | **Data type** | **Purpose** |
| CurrentGuess | Character | Stores the Guess of either H or T entered by the user |
| CurrentFlip | Character | Stores either H or T To represent Heads or Tails as the result of a simulated coin toss. |
| NumberOfCorrectGuesses | Integer | Stores an integer that holds the number of correct guesses the user has made. Should be initialized as zero |
| FlipSequence | String | Stores a string of H or T characters that represent a series of results of Tossing a coin. It should be hard coded to hold “XHTHTT”  (the X is there so that you can treat this as a string array starting at element1) |
| NumberOfFlips | Integer | A constant that holds the maximum number of flips to be tossed during the game. It should be set to 5 |

LOOP

OUTPUT "Please Guess Heads (H) or Tails (T)"

INPUT CurrentGuess

OUTPUT "Coin Fliping...."

CurrentFlip = FlipSequence(NumberOfCorrectGuesses + 1)

OUTPUT "The coin landed "

OUTPUT CurrentFlip

IF CurrentGuess = CurrentFlip

THEN

OUTPUT "well done. Let’s flip again."

NumberOfCorrectGuesses = NumberOfCorrectGuesses + 1

IF NumberOfCorrectGuesses = NumberOfFlips

OUTPUT "Congratulations You completed the sequence. Game Over"

ENDIF

ELSE OUTPUT "sorry you guessed incorrectly. GAME OVER")

ENDIF

UNTIL CurrentFlip <> CurrentGuess Or NumberOfCorrectGuesses = NumberOfFlips

OUTPUT "you guessed " NumberOfCorrectGuesses " flips correctly"

**What you need to do**

Write a program for the above algorithm.

Test the program by playing the game twice.   
(i) Once where the user correctly guesses the first flip and then incorrectly guesses the second flip.   
(ii) Secondly where the user correctly guesses all 5 flips.   
Save the program in your new folder/directory.

**Evidence that you need to provide**

|  |  |
| --- | --- |
| **1** | **9** |

(a)     Your PROGRAM SOURCE CODE.

**(9)**

|  |  |
| --- | --- |
| **2** | **0** |

SCREEN CAPTURE(S)The first test described above, where the user correctly guesses the first flip and then incorrectly guesses the second flip **(2)**

|  |  |
| --- | --- |
| **2** | **1** |

SCREEN CAPTURE(S) where the user correctly guesses all 5 flips **(1)**

**Turn over for the next section**

**Section C**

You are advised to spend no more than **10 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.

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

**Question 5**

 Give the name of an identifier that has a fixed value role. *(1 mark)*

 Give an example of a variable that has a transformation role. *(1 mark)*

 State the name of a user-defined function that has exactly 5 parameters *(1 mark)*

 What are the possible values that can be returned by:

Board(RankNo, FileNo)(0) *(2 marks)*

 Assuming that the board has been set up in the default mode at the start of a game, what will be returned by:

Board(1, 5)(1)

*(1 mark)*

 This time, the sample game has been chosen and “White” is about to make his first move.

If white enters a start square and a finish square of 81 and 83 respectively, what will be returned by the function CheckMoveIsLegal?

*(1 mark)*

The following decision table shows the logic used in the CheckMoveIsLegal function to determine whether the correct colour piece is about to be moved.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Conditions | WhoseTurn=”W” | True | True | False | False |
| PieceColour <>”W” | False | True | True | False |
| PieceColour <>”B” | True | False | False | True |
| Actions | Return True | X |  | (a) | (c) |
| Return False |  | X | (b) | (d) |

 Which cell(s) out of (a), (b), (c) and (d) should also have an ‘X’ in them? *(1 mark)*

 There is a variable called RankNo in the DisplayBoard subroutine.

There is also a variable called RankNo in the InitialiseBoard subroutine.

Explain why these two different variables can have the same identifier. *(2 marks)*

**Turn over**

**Section D**

You are advised to spend no more than **40 minutes** on this section.

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

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

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Question 6** | | | This question refers to the procedure GetMove  The user is prompted to *"Enter coordinates of square to move piece from (file first):"*  Add Validation to the procedure so that the user is shown the prompt repeatedly until a valid entry is entered: The user should see the message “*You have entered and invalid coordinate”* before being prompted to input the coordinate again.  A valid coordinate is defined as a two digit integer where the first digit is between 1 and 8 and the second digit is between 1 and 8. You should ensure the program cannot not throw a type exception.  You should apply a similar validation to the input of the finish square with the same conditions and message. Each input should be validated separately the finish square coordinate only being allowed after a valid start square coordinate has been inputted.  All amended code should be written within the GetMove Procedure  **Evidence that you need to provide**  *Include the following in your Electronic Answer Document.* |
| **3** | **4** | Your amended PROGRAM SOURCE CODE for the subroutine GetMove. (6 marks) | |
|  |  |
| **3** | **5** | Perform a series of tests using the inputs below there are 6 tests in total, the first 4 shouldn’t require a FinishSquare input the last test shouldn’t require the input of Start square again.  StartSquare FinishSquare  8 -  05 -  59 -  four one -  17 09  - 27  Screenshots of the test you have performed ensuring you can see the input and resultant outputs (3 Marks) | |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Question 7** | | | This question will require you to create a new Function called **CheckKashshaptuMoveIsLegal**  as well as amending existing code.  You are to change the way the games plays to take into account the ancient rule that when a Reddum reaches the far side of the board it is changed into a Kashshaptu rather than a Marzaz pani. The Kashaptu is able to move in the following ways:   * Any number of squares at a time Vertically or horizontally OR * One square at a time along any diagonal * A Kashshaptu cannot jump over pieces * A Kashshaptu can capture the oppositions piece by landing on that square   The Kashshaptu is represented on the board by the letter “K” WK for white BK for black  In testing the game modifications you will play a sample game and make the following moves:  **12🡪11**  86🡪87  **11🡪22**  87🡪88  **22🡪28** (shouldn’t be allowed) **22🡪23**  88🡪18  23🡪83  **18🡪13**  **Evidence that you need to provide:**  *Include the following in your Electronic Answer Document.* |
| **3** | **6** | Your amended PROGRAM SOURCE CODE for the subroutines C**heckKashshaptuMoveIsLegal**  , **CheckMoveIsLegal** and **MakeMove**  (9 marks) | |
|  |  |
| **3** | **7** | Provide a screenshot showing the **input for the (attempted) moves** highlighted above and the **resulting game output (move or error msg)**.  DO NOT SCREENSHOT ANY OTHER MOVES. You may need to make the console bigger! (5 Marks) | |
|  |  |

**END OF QUESTION**