0 1 The program uses different variables to store single characters or the empty string. The variable identifiers are:
`FirstSignal`, `MenuOption`, `PlainTextLetter`, `Signal` and `Symbol`.

Using the variable identifiers listed above, complete **Table 5** by filling in the unshaded cells with the correct variable identifier.

Table 5

Variable identifier	Description
	uncoded letter, part of `PlainText`
	single unit of `Transmission` (= or `SPACE` or `EOL`)
	first character in `Transmission`
	used to build `SymbolString` (. or -)

Copy the contents of all the unshaded cells in **Table 5** into your Electronic Answer Document.

[4 marks]

0 2 This question refers to the subroutine `ReceiveMorseCode`.

0 2 . 1 What is the purpose of the variable `i` in this subroutine?

[2 marks]

0 2 . 2 Explain what happens when the `GetTransmission` subroutine is called by the `ReceiveMorseCode` subroutine and the file to be read contains only spaces. You must include in your explanation how the subroutine `ReceiveMorseCode` deals with this situation.

[5 marks]

0 3 This question refers to the subroutine `GetNextSymbol`.

Give an example of a transmission string that would generate the error message `"Non-standard symbol received"` and explain how the code will detect the error caused by your string.

[3 marks]

0 4 Morse codes also exist for numerals, so that a text message including numbers can also be transmitted. For example, the Morse code for 3 is:

•••−−

The program is to be extended to include the Morse code for all numerals from 0 to 9.

Describe the changes that would need to be made to the **Skeleton Program** to achieve this.

In your answer you should discuss the changes that need to be made to the data structures and subroutines.

You are **not** expected to actually make the changes.

[6 marks]

0 5 The Skeleton Program uses several data structures.

State the identifier of the data structure that stores values of more than one data type.

[1 mark]

0 6 This question refers to the subroutines `Game` and `ListPossibleMoves`.

Describe what is contained in the parameter `PlayersPieces` when `ListPossibleMoves` is called for the first time in the subroutine `Game`.

[1 mark]

0 7 This question refers to the data structure `A` defined in the `Game` subroutine.

0 7 . 1 Explain the purpose of each of the first two values stored in row 0 in this data structure.

[2 marks]

0 7 . 2 Explain why there are 12 rows after row 0 in this data structure.

[1 mark]

0 7 . 3 Explain the purpose of the values stored in these 12 rows.

[2 marks]

0 8 This question refers to the subroutine `ListPossibleMoves`.

Explain the uses of the variable `NumberOfMoves`.

[2 marks]

Turn over ▶

0 9 This question will further change the rules of the game.

When a piece is promoted to a dame, the player who the new dame belongs to now chooses **one** of the opponent's pieces. This piece is removed from the board and the dame is placed in the square the removed piece was in.

0 9 . 1 State the identifier of the data structure that now needs to be passed as a parameter into the subroutine `MoveDame`.

[1 mark]

What you need to do:

Task 1
Amend the subroutine `MoveDame`.

This subroutine is to:

- ask the player the piece ID of the opponent's piece they want to remove
- check that the piece is an opponent's piece and is on the board
- remove the opponent's piece
- return the coordinates for the new dame.

Task 2
Amend the calls to `MoveDame` from within the subroutine `MovePiece`.

You will need to amend the parameter list of the subroutine heading of `MovePiece` and the call to `MovePiece` from within the subroutine `MakeMove`.

Task 3
Test that the changes you have made work by conducting the following test:

- run the program
- enter `Y`
- load `game3.txt`
- move `a2` to row 7, column 0
- take piece `b1`

Task 4

- move `b5` to row 0, column 3
- take piece `a6`

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

0 9 . 2 Your PROGRAM SOURCE CODE for the entire subroutine `MoveDame` and the entire subroutine `MakeMove`.

[9 marks]

0 9 . 3 SCREEN CAPTURE(S) showing the requested test including the board display after piece `b1` has been taken.

[1 mark]

0 9 . 4 SCREEN CAPTURE(S) showing the requested test including the board display after piece `a6` has been taken.

[1 mark]

1 0 The **Skeleton Program** uses a number of data structures.

1 0 . 1 State the identifier of the data structure that stores values of **more than one data type**.

[1 mark]

1 0 . 2 State the identifier of a data structure that stores values of **only one data type**.

[1 mark]

1 1 State the identifier **and** data type of a variable in the **Skeleton Program** that can only represent two different values.

[2 marks]

1 2 The **Skeleton Program** uses the data structure `Answer`.

1 2 . 1 State what the second element of this data structure, `Answer[1]`, is used for.

[1 mark]

1 2 . 2 Explain the purpose of the third element of this data structure, `Answer[2]`, **and** how it is used.

[3 marks]

1 3 The grid size of the puzzle is currently 9 and sub-grids are 3 x 3.

If the grid size were to be increased to 16, the sub-grids would be 4 x 4 and the hexadecimal digits 0 1 2 3 4 5 6 7 8 9 A B C D E F could be used to solve the puzzle.

Describe **two** changes that would need to be made to the subroutine `DisplayGrid` to enable this.

You should not make any changes to the **Skeleton Program** to answer this question.

[2 marks]

1 4 The **Skeleton Program** uses a number of data structures.

1 4 . 1 State the identifier of a data structure that stores values of **more than one data type**.

[1 mark]

1 4 . 2 State the identifier of a data structure that stores values of **only one data type**.

[1 mark]

15 The **Skeleton Program** uses a number of data structures.

15.1 State the identifier of a data structure that stores values of a **user-defined** data type.

[1 mark]

15.2 State the identifier of a data structure that stores values of only one **built-in** data type.

[1 mark]

16 One of the statistics calculated is average queue length.

This is calculated using the data structure `Stats`.

16.1 Describe how some of the values in `Stats` are used to calculate the average queue length in the subroutine `OutputStats`.

[1 mark]

16.2 Describe how the relevant values in `Stats` are updated in the subroutine `Serving` to enable the calculation in the subroutine `OutputStats` to produce the correct result.

[2 marks]

17 In the **Skeleton Program** buyers join a single queue and are then served at one of several tills.

Outline the design changes needed for buyers to form a separate queue for each till.

You are not expected to actually make the changes.

[2 marks]