

AS COMPUTING

Unit 1 Problem Solving, Programming, Data Representation and
Practical Exercise

Monday 6 June 2016

Morning

Time allowed: 2 hours

You will need to:

- access the Electronic Answer Document
- refer to the Preliminary Material and Skeleton Program.

You must **not** use a calculator.

Instructions

- Type the information required on the front of your Electronic Answer Document.
- Enter your answers into the Electronic Answer Document.
- Answer **all** questions.
- You will need access to:
 - a computer
 - a printer
 - appropriate software
 - an electronic version of the Skeleton Program
 - a hard copy of the Preliminary Material.
- 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).

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 advised to spend time on each section as follows:
 - Section A – 35 minutes
 - Section B – 15 minutes
 - Section C – 15 minutes
 - Section D – 55 minutes.

At the end of the examination

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

Warning

- It may not be possible to issue a result for this unit if your details are not on every page of the Electronic Answer Document.

Section A

You are advised to spend no more than **35 minutes** on this section.

Enter your answers to **Section A** in your Electronic Answer Document.
You **must save** this document at regular intervals.

Question 1 **Figure 1** shows the contents of a memory location.

Figure 1

1	0	1	1	0	1	0	0
---	---	---	---	---	---	---	---

0 1

What is the denary equivalent of the contents of this memory location if it represents an **unsigned binary integer**?

Use the space below for rough working, then copy the answer to your Electronic Answer Document.

[1 mark]

0 2

What is the denary equivalent of the contents of this memory location if it represents an **unsigned binary fixed point number**, with 3 bits before and 5 bits after the binary point?

Use the space below for rough working, then copy the answer to your Electronic Answer Document.

[2 marks]

0 3

Why are bit patterns often displayed using hexadecimal instead of binary?

[1 mark]

0	4
---	---

What is the **hexadecimal** equivalent of the contents of the memory location shown in **Figure 1**?

Use the space below for rough working, then copy the answer to your Electronic Answer Document.

[2 marks]

0	5
---	---

What is the denary equivalent of the contents of this memory location if it represents a **two's complement binary integer**?

Use the space below for rough working, then copy the answer to your Electronic Answer Document.

[2 marks]

The ASCII system uses 7 bits to represent a character. The ASCII code in denary for the numeric character '0' is 48; other numeric characters follow on from this in sequence.

0	6
---	---

What character is being represented by the contents of the memory location shown in **Figure 1** if it represents a 7-bit ASCII code with the most significant bit being used as an even parity bit?

[1 mark]

Hamming code is an alternative to the use of a single parity bit.

0	7
---	---

State **two** advantages of using Hamming code instead of a single parity bit.

[2 marks]

0	8
---	---

If the memory location contains a bit pattern that is being transmitted using Hamming code, how many of the bits shown in **Figure 1** have been used as parity bits?

[1 mark]

Turn over ►

Question 2 Images are often represented in a computer's memory using bitmapped graphics.

0	9
---	---

What is meant by the **resolution** of a bitmapped graphic image?

[2 marks]

1	0
---	---

How many colours can a bitmapped graphic image with a colour depth of 4 bits contain?

[1 mark]

1	1
---	---

A bitmapped image has 16 x 16 pixels. It is stored in an image format that is limited to 65536 colours. Calculate the minimum image size in bytes.

Use the space below for rough working then copy the answer, and your working, to your Electronic Answer Document. You may get some marks for your working even if your answer is incorrect if you include the working in your Electronic Answer Document.

[3 marks]

Instead of using bitmapped graphics, images may be represented in a computer's main memory using vector graphics.

1	2
---	---

State **two** advantages of vector graphics compared to bitmapped graphics.

[2 marks]

There are no questions printed on this page

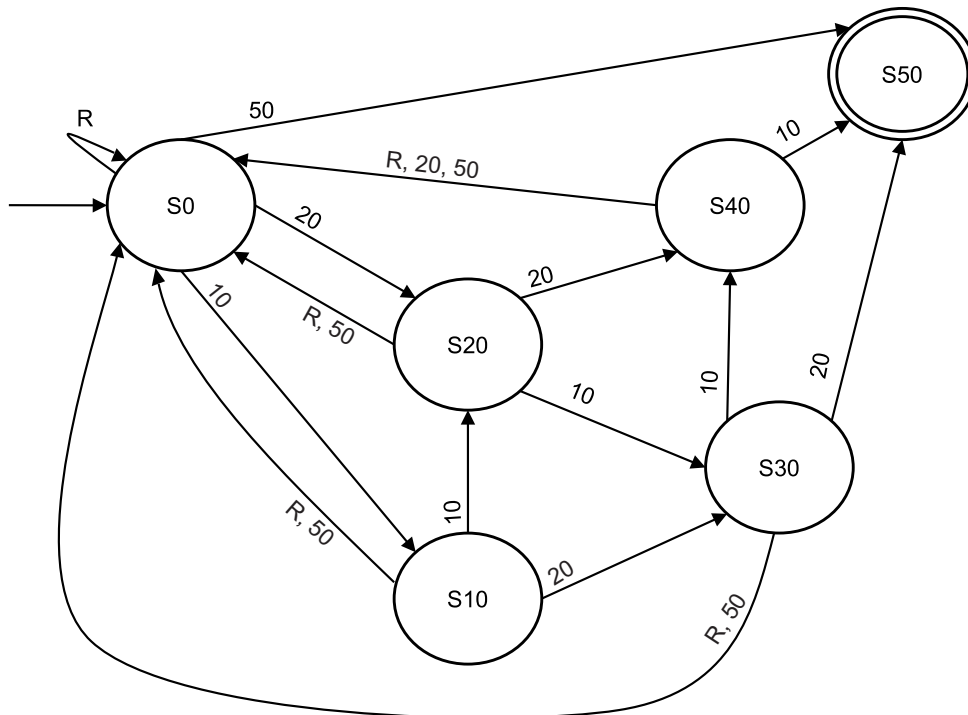
Question 3 starts on the next page

Turn over ►

Question 3 Figure 2 shows the state transition diagram of a finite state machine (FSM) used to control a vending machine.

The vending machine dispenses a drink when a customer has inserted exactly 50 pence. A transaction is cancelled and coins returned to the customer if more than 50 pence is inserted or the reject button (R) is pressed. The vending machine accepts 10, 20 and 50 pence coins. Only one type of drink is available.

Figure 2



1 3

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 part of **Figure 2**.

Complete **Table 1** by filling in the unshaded cells.

Copy the contents of all unshaded cells in **Table 1** into the Electronic Answer Document.

Table 1

Original State	Input	New State
S30	10	S40
S30		
S30		
S30		

[2 marks]

1 | 4

The vending machine is to be adapted so that it also accepts 5p coins. What is the minimum number of states that will need to be added to the FSM shown in **Figure 2** so that 5p coins are also accepted?

[1 mark]

Question 4 A speech track for a presentation is being recorded.

Figure 3 shows three samples stored in a computer's memory that have been taken from an analogue signal as part of the recording process. The highest frequency in the original speech is 800Hz (Hertz).

1Hz is one sample per second.

Figure 3

1000	1010	1110
1000	1100	1110
1000	1111	0011

1 | 5

What sampling resolution has been used?

[1 mark]

1 | 6

What is the minimum sampling rate (in Hz) that should be used to ensure that there is no noticeable loss in quality when the original analogue signal has been represented in digital form?

[1 mark]

1 | 7

If the original analogue signal lasts 20 seconds, how many bytes of storage will be required to store all the samples taken in the recording process with no noticeable loss in quality?

Use the space below for rough working then copy the answer, and your working, to your Electronic Answer Document. You may get some marks for your working even if your answer is incorrect if you include the working in your Electronic Answer Document.

[3 marks]

1 | 8

Define what an analogue signal is and explain how it differs from a digital signal.

[3 marks]

Turn over ►

Section B

You are advised to spend no more than **15 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/file in its own folder/directory.
- You are advised to save your program at regular intervals.

Question 5 Create a folder/directory **Question5** for your new program.

The algorithm, represented using pseudo-code in **Figure 4**, and the variable table, **Table 2**, describe a program that outputs an **estimate** for a particular calculation.

Figure 4

```

OUTPUT "Enter a number:"
INPUT N
F ← 16.0
IF N >= 1.0
  THEN
    X ← N
    WHILE X * X - N > 1.0 AND F - 1.0 > 1.0 DO
      L ← X
      X ← X ÷ F
      WHILE X * X <= N DO
        F ← F - 0.1
        X ← L ÷ F
      ENDWHILE
    ENDWHILE
  OUTPUT X
ELSE
  OUTPUT "Not a number greater than or equal to 1"
ENDIF

```

Table 2

Identifier	Data type
X	Real number
F	Real number
L	Real number
N	Real number

What you need to do

Write a program for the algorithm in **Figure 4**.

Test the program by conducting the tests **Test 1** and **Test 2**, below.

Save the program in your new **Question5** folder/directory.

Test 1

Test that your program works correctly by entering the number 0.1

Test 2

Test that your program works correctly by entering the number 4.1

Evidence that you need to provide

Include the following in your Electronic Answer Document.

1 9	Your PROGRAM SOURCE CODE.	[14 marks]
2 0	SCREEN CAPTURE(S) showing the result of Test 1 .	[1 mark]
2 1	SCREEN CAPTURE(S) showing the result of Test 2 .	[1 mark]
2 2	Describe the purpose of the algorithm represented using pseudo-code in Figure 4 .	[1 mark]
2 3	Explain what is meant by an algorithm.	[2 marks]

Turn over for Section C

Turn over ►

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.

Question 6 State the name of an identifier for:

- | | | |
|---------------------|--|-----------------|
| 2 4 | a variable that only stores one of two different values. | [1 mark] |
| 2 5 | a user-defined subroutine that has exactly two parameters. | [1 mark] |
| 2 6 | a variable that has a gatherer role. | [1 mark] |
| 2 7 | a variable that has a stepper role. | [1 mark] |
| 2 8 | a variable that has a transformation role. | [1 mark] |
| 2 9 | a variable that has a most recent holder role. | [1 mark] |
| 3 0 | a user-defined subroutine that has no selection structures. | [1 mark] |
| 3 1 | Give one example of an assignment statement in the subroutine <code>CheckIfMoveIsValid</code> . | [1 mark] |

The algorithm represented using pseudo-code in **Figure 5** shows a modified version of part of the `MakeMove` subroutine. The modifications, highlighted in bold, are supposed to change the game so that any opponent pieces between two of the player's pieces on a straight line formed along a diagonal are also flipped.

You should assume that all the statements after the `End If` in **Figure 5** (which are represented by '...') are correct.

Figure 5

```

Procedure MakeMove(Board, BoardSize, Move, HumanPlayersTurn)
  FlipOpponentPiecesInOneDirection(BoardSize, Row, Column, 1, 1)
  FlipOpponentPiecesInOneDirection(BoardSize, Row, Column, -1, -1)
  FlipOpponentPiecesInOneDirection(BoardSize, Row, Column, 0, 0)
  Row ← Move Mod 10
  Column ← Move Div 10
  If HumanPlayersTurn Then
    Board[Row, Column] ← "H"
  Else
    Board[Row, Column] ← "C"
  End If
  ...
End Procedure

```

3 | 2

Describe the errors that have been made in the pseudo-code shown in **Figure 5**.

[4 marks]

3 | 3

In the `ChangeBoardSize` subroutine there is a check to see if the board size is between the values of 4 and 9. If this check was altered so that the board size could be between the values of 4 and 10 then the program would not work correctly.

Explain why the `CheckIfMoveIsValid` subroutine would not always work correctly after this change, even if the `Board` array had been correctly modified so that a board size of 10x10 could be represented.

[2 marks]

Turn over ►

Section D

You are advised to spend no more than **55 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 make programming changes to it.

Question 7 This question refers to the subroutine `GetPlayersName`.

It should not be possible for a player to enter a blank name.

Adapt the subroutine `GetPlayersName` so that if a blank name is entered the message `That is not a valid name, using default name` instead is displayed. The player's name should then be set to a default value of `Human player`.

Test that the changes you have made work:

- run the **Skeleton Program**
- enter `e` at the menu
- press the enter key without entering a name for the player.

Evidence that you need to provide

Include the following in your Electronic Answer Document.

3 4	Your PROGRAM SOURCE CODE for the amended subroutine <code>GetPlayersName</code> .	[3 marks]
3 5	SCREEN CAPTURE(S) showing the result of the test.	[2 marks]

Question 8 This question refers to the subroutines `CheckIfMoveIsValid` and `PlayGame`.

The game is to be adapted so that a player is only allowed to make a move if the square they have chosen is within the boundaries of the board.

Adapt the **Skeleton Program** so that the new definition of a valid move is implemented. You should pass an additional parameter to the `CheckIfMoveIsValid` subroutine indicating the size of the board.

Test that the changes you have made work:

- run the **Skeleton Program**
- enter `p` at the menu
- enter a move of `74`
- enter a move of `47`
- enter a move of `10`
- enter a move of `01`
- enter a move of `66`.

Evidence that you need to provide

Include the following in your Electronic Answer Document.

3 | 6

Your PROGRAM SOURCE CODE for the amended subroutine `PlayGame`.

[1 mark]

3 | 7

Your PROGRAM SOURCE CODE for the amended subroutine `CheckIfMoveIsValid`.

[5 marks]

3 | 8

SCREEN CAPTURE(S) showing the requested test. You must make sure that evidence for all parts of the requested test is provided in the SCREEN CAPTURE(S).

[2 marks]

Turn over for the next question

Turn over ►

Question 9 This question will further extend the functionality of the **Skeleton Program** and refers to the `SetUpGameBoard` subroutine.

When a new game is started the human player is to be offered a choice of two different start configurations. In the first, the four pieces on the board will be placed in the middle as is currently the case. In the second, the pieces will be placed in the four corner squares of the board.

When the four corner squares option is chosen the human player will start with pieces in the top-left and bottom-right corner squares, and the computer will start with pieces in the top-right and bottom-left corner squares.

Figure 6 shows what the board looks like at the start of a new game if the human player chooses the four corner squares option on a 9x9 size board.

Figure 6

H								C
C								H

Figure 7 shows what the board looks like at the start of a new game if the human player chooses the four corner squares option on a 4x4 size board.

Figure 7

H			C
C			H

Adapt the `SetUpGameBoard` subroutine so that the human player is asked which of the two starting configurations they wish to choose (pieces in middle or pieces in corner).

If the four centre squares initial setup is chosen then the game should start with the four pieces in the middle of the board, as is currently the case.

If the four corner squares initial setup is chosen then the game should start with the board displayed as shown in **Figure 6** and **Figure 7**. To get full marks for this question your program code will need to work correctly for any allowed board size.

After writing your program code you will need to carry out the following tests.

Test 1

Test that the changes you have made work:

- run the **Skeleton Program**
- enter `p` at the menu
- choose the **four centre squares** initial setup.

Test 2

Test that the changes you have made work:

- run the **Skeleton Program**
- enter `p` at the menu
- choose the **four corner squares** initial setup.

Test 3

Test that the changes you have made work:

- run the **Skeleton Program**
- enter `c` at the menu
- enter a board size of 8
- enter `p` at the menu
- choose the **four corner squares** initial setup.

Evidence that you need to provide

Include the following in your Electronic Answer Document.

3 9	Your PROGRAM SOURCE CODE for the amended subroutine <code>SetUpGameBoard</code> .	[7 marks]
4 0	SCREEN CAPTURE(S) showing the results of Test 1 . You must make sure that evidence for all parts of the requested test is provided in the SCREEN CAPTURE(S).	[1 mark]
4 1	SCREEN CAPTURE(S) showing the results of Test 2 . You must make sure that evidence for all parts of the requested test is provided in the SCREEN CAPTURE(S).	[1 mark]
4 2	SCREEN CAPTURE(S) showing the results of Test 3 . You must make sure that evidence for all parts of the requested test is provided in the SCREEN CAPTURE(S).	[1 mark]

Turn over ►

Question 10 This question will further extend the functionality of the **Skeleton Program**.

The game is to be adapted so that before the human player is asked to enter their move, the message `Choosing one of the following squares will mean you flip some pieces:` is displayed. After this message, the program should display a list of all of the squares that the player could choose that would result in some of the computer's pieces being flipped.

Task 1

Create a subroutine called `GetFlipSquares` which takes two parameters (the board and the board size) and displays a list of all the squares where the human player could place a piece that would result in some of the computer's pieces being flipped.

You might find it easier to answer this question if you make subroutine calls to the `CheckIfThereArePiecesToFlip` subroutine from within your new subroutine `GetFlipSquares`.

You are likely to get some marks for this task even if your subroutine is only partially working.

Task 2

Adapt the `PlayGame` subroutine so that there is a call to the subroutine `GetFlipSquares`. This call should be after the board is displayed and before the player is asked to enter their move. This call should happen every time it is the human player's move but not when it is the computer's move.

Task 3

Test that your program works by conducting the following test:

- run the **Skeleton Program**
- enter `p` at the menu
- if you have answered **Question 9** then you should choose the **four centre squares** setup. You do not need to have answered **Question 9** to answer **Question 10**.

Evidence that you need to provide

Include the following in your Electronic Answer Document.

4	3	Your PROGRAM SOURCE CODE for the subroutine <code>GetFlipSquares</code> .	[10 marks]
4	4	Your PROGRAM SOURCE CODE for the amended <code>PlayGame</code> subroutine.	[2 marks]
4	5	SCREEN CAPTURE(S) showing the requested test.	[1 mark]

END OF QUESTIONS

Copyright information

For confidentiality purposes, from the November 2015 examination series, acknowledgements of third party copyright material will be published in a separate booklet rather than including them on the examination paper or support materials. This booklet is published after each examination series and is available for free download from www.aqa.org.uk after the live examination series.

Permission to reproduce all copyright material has been applied for. In some cases, efforts to contact copyright-holders may have been unsuccessful and AQA will be happy to rectify any omissions of acknowledgements. If you have any queries please contact the Copyright Team, AQA, Stag Hill House, Guildford, GU2 7XJ.

Copyright © 2016 AQA and its licensors. All rights reserved.