General Certificate of Education
Advanced Subsidiary Examination
June 2014

Computing COMP1

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

Monday 2 June 2014 9.00 am to 11.00 am

You will need to:
- access the Electronic Answer Document
- refer to the Preliminary Material, Data File and Skeleton Program

You must **not** use a calculator

Time allowed
- 2 hours

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 and Data File
  - 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 – 20 minutes
  Section C – 10 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 an 8-bit bit pattern.

Figure 1

1	0	1	1	0	1	1	0

0 1 If the bit pattern in **Figure 1** is an **unsigned binary integer**, what is the denary equivalent of this bit pattern?

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

[1 mark]

0 2 If the bit pattern in **Figure 1** is a **two's complement binary integer**, what is the denary equivalent of this bit pattern?

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

[2 marks]

0 3 What is the range of **denary** numbers that can be represented using **8-bit two's complement binary integers**?

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

[2 marks]

0 4 If the bit pattern in **Figure 1** is an **unsigned binary fixed point** number with 3 bits before and 5 bits after the binary point, what is the denary equivalent of this bit pattern?

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

[2 marks]

0 5 What is the **hexadecimal** equivalent of the bit pattern in **Figure 1**?

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

[2 marks]

0 6 Why are bit patterns often displayed using hexadecimal instead of binary?

[1 mark]

0 7 Describe a method that can, without the use of binary addition, multiply any **unsigned binary integer** by the binary number 10 (the denary number 2).

[2 marks]

Turn over for the next question

Question 2 A pseudo-code representation of an algorithm is given in **Figure 2**.

Figure 2

```
FOR x ← 0 TO 7 DO
  IF (x MOD 16 >= 4) AND (x MOD 16 <= 11)
    THEN c ← 1
    ELSE c ← 0
  ENDIF
  IF (x MOD 8 >= 2) AND (x MOD 8 <= 5)
    THEN b ← 1
    ELSE b ← 0
  ENDIF
  IF (x MOD 4 = 0) OR (x MOD 4 = 3)
    THEN a ← 0
    ELSE a ← 1
  ENDIF
  OUTPUT (c, b, a)
ENDFOR
```

The `MOD` operator calculates the remainder resulting from an integer division. For example, `7 MOD 5 = 2, 14 MOD 5 = 4`.

The statement `OUTPUT (c, b, a)` will display the contents of the variable `c`, followed by the contents of the variable `b` and then the contents of the variable `a`.

0 8 Explain what is meant by an **algorithm**.

[2 marks]

0 9 The decision table shown in **Table 1** represents the logic of the second selection structure in the algorithm shown in **Figure 2**. The decision table is only partially complete; the shaded cells that should show the actions to be taken have not been filled in.

Table 1

Conditions	`x MOD 8 >= 2`	`False`	`True`	`True`
	`x MOD 8 <= 5`	`True`	`False`	`True`
Action	`b ← 1`			
	`b ← 0`			

Complete **Table 1** so that it shows the actions to be taken when the conditions have particular values: an 'X' symbol should be placed in the relevant shaded cells to indicate the action that will be executed for the given conditions. Some of the shaded cells will need to be left empty.

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

[2 marks]

1 0 Dry run the algorithm in **Figure 2** by completing **Table 2**. The first row has been completed for you.

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

Table 2

x	c	b	a	Printed output
0	0	0	0	000

[5 marks]

1 1 Explain, precisely, the purpose of the algorithm in **Figure 2**.

[1 mark]

Question 3 **Figure 3** shows ten numbers stored in an array L.

Figure 3

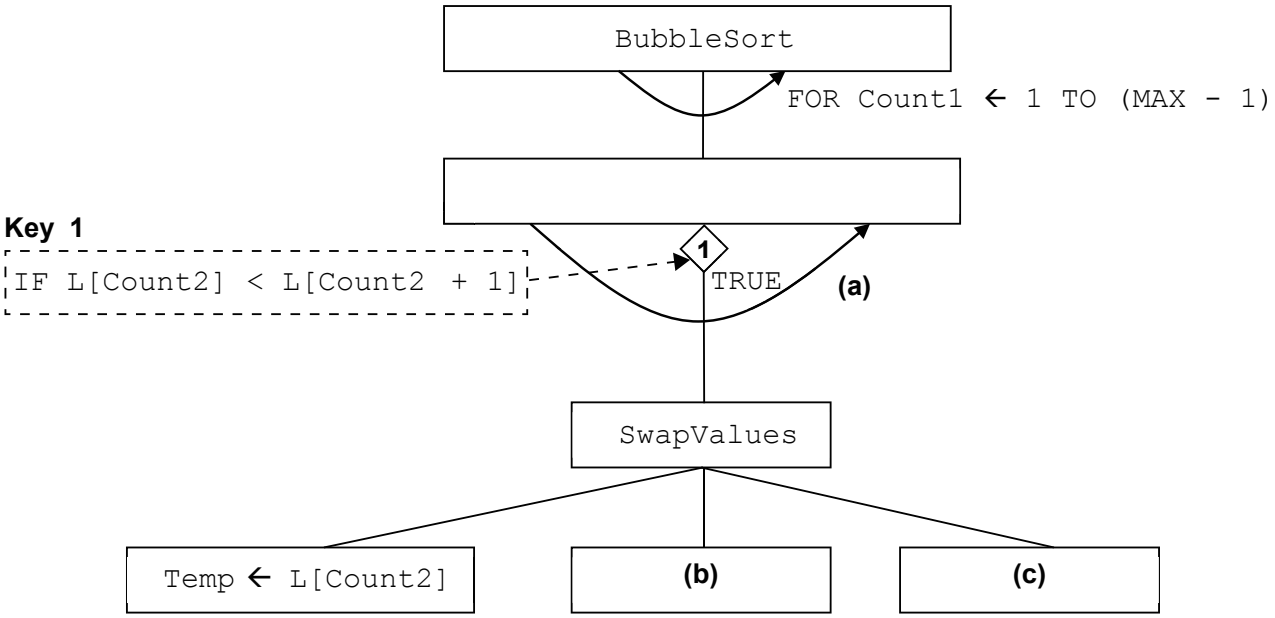
L									
[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
34	8	6	35	27	35	63	56	16	24

The numbers in L are to be sorted.

Figure 4 shows an **incomplete** structure chart that has been created while developing a solution to the problem of sorting the numbers in L.

The constant MAX has been used to represent the size of the array.

Figure 4



1 2 Describe the **goal** of this problem.

[1 mark]

1 3 What is meant by the **given** of a problem?

[1 mark]

1 4 The given and the goal are two components of a well-defined problem. State **two** other components of a well-defined problem.

[2 marks]

1 5 How should the curved arrow **(a)** in **Figure 4** be labelled?

[1 mark]

1 6 What should be written in box **(b)** in **Figure 4**?

[1 mark]

1 7 What should be written in box **(c)** in **Figure 4**?

[1 mark]

A **new** Bubble Sort routine is developed using the structure chart shown in **Figure 4**.

1 8 What value will be in `L[1]` when **this** Bubble Sort routine has finished executing on the array `L` shown in **Figure 3**?

[1 mark]

1 9 A Bubble Sort routine, based on the structure chart in **Figure 4**, always completes `MAX - 1` passes through the array. Often, this number of passes is not required, as the contents of the array will be sorted after fewer passes have been made.

If a pass is made through the array during which no swaps need to be made then the array has been sorted.

Describe the changes that need to be made to the Bubble Sort routine so that it will only complete the minimum number of passes through the array that are needed to fully sort the contents of the array.

[3 marks]

Turn over for the next question

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/file 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 algorithm, represented using pseudo-code in **Figure 5**, and the variable table, **Table 3**, describe the process of using a check digit to check if a value entered by the user is a valid 13 digit International Standard Book Number (ISBN).

Figure 5

```
FOR Count ← 1 TO 13 DO
  OUTPUT "Please enter next digit of ISBN: "
  INPUT ISBN[Count]
ENDFOR
CalculatedDigit ← 0
Count ← 1
WHILE Count < 13 DO
  CalculatedDigit ← CalculatedDigit + ISBN[Count]
  Count ← Count + 1
  CalculatedDigit ← CalculatedDigit + ISBN[Count] * 3
  Count ← Count + 1
ENDWHILE
WHILE CalculatedDigit >= 10 DO
  CalculatedDigit ← CalculatedDigit - 10
ENDWHILE
CalculatedDigit ← 10 - CalculatedDigit
IF CalculatedDigit = 10
  THEN CalculatedDigit ← 0
ENDIF
IF CalculatedDigit = ISBN[13]
  THEN OUTPUT "Valid ISBN"
  ELSE OUTPUT "Invalid ISBN"
ENDIF
```

Table 3

Identifier	Data Type	Purpose
`ISBN`	`Array[1..13] Of Integer`	Stores the 13 digit ISBN entered by the user – one digit is stored in each element of the array.
`Count`	`Integer`	Used to select a specific digit in the ISBN.
`CalculatedDigit`	`Integer`	Used to store the digit calculated from the first 12 digits of the ISBN. It is also used to store the intermediate results of the calculation.

What you need to do

Write a program for the algorithm in **Figure 5**.

Test the program by showing the result of entering the digits `9, 7, 8, 0, 0, 9, 9, 4, 1, 0, 6, 7, 6` (in that order).

Test the program by showing the result of entering the digits `9, 7, 8, 1, 8, 5, 7, 0, 2, 8, 8, 9, 4` (in that order).

Save the program in your new **Question4** folder/directory.

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

2 0 Your PROGRAM SOURCE CODE.

[15 marks]

2 1 SCREEN CAPTURE(S) for the test when the digits `9, 7, 8, 0, 0, 9, 9, 4, 1, 0, 6, 7, 6` are entered (in that order).

Your evidence must show the result of the test and, as a minimum, the last three digits entered for the test.

[2 marks]

2 2 SCREEN CAPTURE(S) for the test when the digits `9, 7, 8, 1, 8, 5, 7, 0, 2, 8, 8, 9, 4` are entered (in that order).

Your evidence must show the result of the test and, as a minimum, the last three digits entered for the test.

[1 mark]

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.

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

Question 5 State the name of an identifier for:

2 3 a user-defined data type.

[1 mark]

2 4 a built-in function that converts a string into a different data type.

[1 mark]

2 5 an array variable.

[1 mark]

A variable can have one of a number of different roles. Some of the different possible roles a variable can have are:

- fixed value
- stepper
- gatherer
- transformation
- follower
- temporary
- most recent holder
- most wanted holder.

For each of the following variables state which of the possible roles best describes the role of the variable:

2 6 `SwapSpace` in the `ShuffleDeck` subroutine.

[1 mark]

2 7 `Choice` in the `PlayGame` subroutine.

[1 mark]

2 8 `NoOfCardsTurnedOver` in the `PlayGame` subroutine.

[1 mark]

An extra subroutine that could have been added to the **Skeleton Program** is `HasPlayerXGotARecentScore`. This subroutine would have looked at the contents of the `RecentScores` array and output a message saying if someone with a particular name has (or has not) got one of the recent scores.

Figure 6 shows a first attempt (written in pseudo-code) to develop an algorithm that the `HasPlayerXGotARecentScore` subroutine could be based on.

Figure 6

```
OUTPUT "Enter a name"
INPUT PlayerX
Found ← False
Position ← 1
WHILE Found = False DO
  IF RecentScores[Position].Name = PlayerX
    THEN Found ← True
    ELSE Position ← Position + 1
  ENDIF
ENDWHILE
IF Found = True
  THEN OUTPUT "Yes, they do have a recent score"
  ELSE OUTPUT "No, they do not have a recent score"
ENDIF
```

The algorithm shown in **Figure 6** is then implemented in a programming language. There is an error in the algorithm which means that when the program is run it sometimes works correctly and sometimes it does not.

2 9 Under what circumstances will a program based on the algorithm shown in **Figure 6** not work as intended?

[1 mark]

3 0 How should the algorithm in **Figure 6** be changed so that this problem is corrected?

You may answer this question by either describing the change(s) needed or by giving a new version of the relevant part(s) of the pseudo-code for the algorithm shown in **Figure 6**.

[2 marks]

3 1 State the name of the algorithm that is being used as a basis for the development of the `HasPlayerXGotARecentScore` subroutine.

[1 mark]

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 6 This question refers to the subroutine `GetPlayerName`.

Add a validation check to the subroutine `GetPlayerName` so that it repeatedly attempts to get the name from the user until a name with at least one character in it is entered (the name cannot be left blank).

Each time an invalid value is entered the message "`You must enter a name`" should be displayed.

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

- run the **Skeleton Program**
- select option 2 from the menu
- play a game
- when the prompt "`Please enter your name: `" is displayed press the Enter key without entering a name
- then enter `Emily` as a name.

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

3	2	Your amended PROGRAM SOURCE CODE for the subroutine `GetPlayerName`. **[4 marks]**

3	3	SCREEN CAPTURE(S) showing the requested test. **[2 marks]**

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

The game is to be altered so that if two cards have the same rank then the suit of the cards determines which of the two cards is the higher. Spades is the highest suit, then hearts, then diamonds, then clubs. For example:

- if the last card was the 7 of Diamonds and the next card is the 7 of Hearts then the subroutine `IsNextCardHigher` should return a value of `True`
- if the last card was the 7 of Diamonds and the next card is the 7 of Clubs then the subroutine `IsNextCardHigher` should return a value of `False`.

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

- run the **Skeleton Program**
- select option 2 from the menu
- when asked if you think the next card will be higher enter `y`, then `n`, then `y`.

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

3 4 Your amended PROGRAM SOURCE CODE for the subroutine `IsNextCardHigher`.

[4 marks]

3 5 SCREEN CAPTURE(S) showing the requested test.

[2 marks]

Turn over for the next question

Question 8 This question will extend the functionality of the game.

The game is to be altered so that the player can play a joker. When asked if they think the next card will be higher the player can enter a `j` to play a joker instead of guessing `y` or `n`. When the player uses a joker it doesn't matter what the next card is as the player is considered to have predicted correctly whether the next card is higher or not.

The player can play a joker a maximum of two times in a game.

Task 1
Adapt the `GetChoiceFromUser` subroutine so that an appropriate message is displayed that informs the user how to play a joker.

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

3 6 Your amended PROGRAM SOURCE CODE for the subroutine `GetChoiceFromUser`.

[1 mark]

Task 2
Adapt the `PlayGame` subroutine so that the player can play a joker in the way described.

Test your program works by conducting the following test:

- run the **Skeleton Program**
- select option 2 from the menu
- when asked if you think the next card will be higher enter `j`, then `j`, then `j`.

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

3 7 Your amended PROGRAM SOURCE CODE for the subroutine `PlayGame`.

[7 marks]

3 8 SCREEN CAPTURE(S) showing the requested test.

[3 marks]

There are no questions printed on this page

Turn to page 16 for the next question

Question 9 This question will further extend the functionality of the game.

The game is to be altered so that the player is told the probability that the next card will be higher than the last card. Each time the player is asked to make a prediction, they should first be shown the probability that the next card will be higher than the last card.

The probability of the next card being higher than the last card can be calculated by performing the division:

$$\frac{\text{Number of cards not yet turned over that are higher than the last card turned over}}{\text{Number of cards not yet turned over}}$$

Additional marks will be awarded in Question 9 for writing code that demonstrates good practice by ensuring subroutines are self-contained and make use of interfaces.

Task 1
Create a new subroutine, `CalculateProbability`, which works out the probability of the next card being higher than the last card. It should return this calculated value to the calling routine. You may choose whether to make the new subroutine a function or a procedure.

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

3	9

Your PROGRAM SOURCE CODE for the subroutine `CalculateProbability`.
[11 marks]

Task 2

Adapt the `PlayGame` subroutine so that, before the second call to the `GetCard` subroutine, the message "`The probability of the next card being higher is X`" is displayed, where `X` is the value returned by the `CalculateProbability` subroutine.

Test your program works by conducting the following test:

- run the **Skeleton Program**
- select option 2 from the menu
- the probability of the next card being higher is displayed to the user
- when asked if you think the next card will be higher enter `y`
- the probability of the next card being higher is displayed to the user.

Evidence that you need to provide

Include the following in your Electronic Answer Document.

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

[3 marks]

4 1 SCREEN CAPTURE(S) showing the requested test.

[2 marks]

END OF QUESTIONS

There are no questions printed on this page

There are no questions printed on this page

There are no questions printed on this page