
INTERNATIONAL GCSE

COMPUTER SCIENCE

Paper 1 Programming

Friday 12 November 2021

07:00 GMT

Time allowed: 2 hours

Materials

For this paper you must have access to:

- a computer
- a printer
- appropriate software
- electronic versions of the **Skeleton Program** and the data file **examples.txt**
- a hard copy of the **Skeleton Program**.

Information

- The marks for questions are shown in brackets.
- The maximum mark for this paper is 80.
- No extra time is allowed for printing and collating.
- The question paper is divided into **three** sections.
- You may use a bilingual dictionary.
- You must **not** use an English dictionary.
- You are **not** allowed to use a calculator.

Instructions

- Type the information required on the front of your **Electronic Answer Document**.
- Answer **all** questions.
- Enter your answers into the **Electronic Answer Document**.
- 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).
- Include the evidence required for your answers to **Sections B** and **C** in your **Electronic Answer Document**.
- You **must save** your **Electronic Answer Document** at regular intervals.
- The questions in **Sections B** and **C** require you to make changes to the **Skeleton Program**.
- All of the programming questions in **Sections B** and **C** can be answered independently of each other. If you cannot answer one of the questions you can still attempt to solve later questions.
- You are advised to keep a backup copy of the original **Skeleton Program** so that you can go back to it if you accidentally make changes to the program which means it can no longer be compiled/executed while answering the questions in **Sections B** and **C**.

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

Section A (Non-programming questions)

You are advised to spend no more than **30 minutes** on this section.

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

You **must save** your **Electronic Answer Document** at regular intervals.

The questions in this section are about programming and how the **Skeleton Program** works.
Do **not** make any changes to the **Skeleton Program** when answering these questions.

0 1 . 1 Name a user-defined subroutine in the **Skeleton Program** that returns a string. **[1 mark]**

0 1 . 2 There are many selection statements in the **Skeleton Program**.
Explain what is meant by a selection statement. **[1 mark]**

0 1 . 3 Name a user-defined subroutine in the **Skeleton Program** that uses definite iteration. **[1 mark]**

0 2 This question is about the `DecimalToBinaryConvert` subroutine.

0 2 . 1 A `WHILE` loop is used in the `DecimalToBinaryConvert` subroutine.
Explain why a `WHILE` loop is used instead of a `FOR` loop. **[1 mark]**

0 2 . 2 Integer division is used in the `DecimalToBinaryConvert` subroutine.
Explain why integer division is used. **[1 mark]**

0 3

The subroutine `BinaryToDecimalConvert` is represented using pseudocode in **Figure 1**.

Figure 1

```

SUBROUTINE BinaryToDecimalConvert(BinaryString)
  DecimalNumber ← 0
  FOR Position ← 0 TO LEN(BinaryString) - 1
    DecimalNumber ← DecimalNumber * 2
    Bit ← BinaryString[Position]
    IF Bit = '1' THEN
      DecimalNumber ← DecimalNumber + 1
    ENDIF
  ENDFOR
  RETURN INT_TO_STRING(DecimalNumber)
ENDSUBROUTINE

```

0 3 . 1

Complete the trace table for the subroutine call `BinaryToDecimalConvert('1100')` and state the value returned.

Copy the contents of all the blank cells in **Table 1** and your answer to **value returned** into your Electronic Answer Document.

[6 marks]**Table 1**

BinaryString	DecimalNumber	Position	Bit
1100			

Value returned: _____

0 3 . 2

The value '1100' was used to test the `BinaryToDecimalConvert` subroutine and is an example of **normal (typical)** test data.

Name the other **two** types of test data.

[2 marks]**Turn over ►**

0 3 . 3 To work correctly, the `BinaryToDecimalConvert` subroutine must be passed a string consisting only of 1s and 0s.

Currently, the `GetBinary` subroutine does not check that the string consists only of 1s and 0s before calling `BinaryToDecimalConvert`.

Four programming techniques are:

- authentication
- concatenation
- string conversion
- validation.

State the name of the programming technique from the list above which could be used to make the `GetBinary` subroutine check that the string consists only of 1s and 0s.

[1 mark]

0 3 . 4 Name a local variable in the `BinaryToDecimalConvert` subroutine that has a character data type.

[1 mark]

0 4 This question is about the `LoadQuiz` subroutine.

0 4. **1** Describe what a text file is.

[2 marks]

0 4. **2** The variable `Filename` could have been defined as a constant.

Describe what a constant is.

[2 marks]

0 5 This question is about the `ShiftRight` subroutine.

0 5. **1** In the `ShiftRight` subroutine there are two separate `FOR` loops.

Explain the purpose of these `FOR` loops.

[2 marks]

0 5. **2** In the `ShiftRight` subroutine, if the user enters:

- any positive shift amount, the subroutine will work as expected
- any negative shift amount, an error will occur.

Explain why an error will occur if the user enters a negative shift amount.

[2 marks]

Turn over for Section B

Turn over ►

Section B (Short programming questions)

You are advised to spend no more than **45 minutes** on this section.

0 6

The **Skeleton Program** is to be improved so that it displays a message when the user quits the program.

The message `Goodbye` should be displayed after the user has picked the quit option from the main menu.

Change the subroutine `Main` so that the message `Goodbye` is displayed after the user has picked the quit option.

Ensure that there is a blank line before the message `Goodbye`.

Test your change has worked by running the **Skeleton Program** and then:

- enter `Q`

Evidence that you need to provide

Include the following evidence in your Electronic Answer Document.

0 6 . 1

All of the PROGRAM SOURCE CODE for the amended subroutine `Main`.

[3 marks]**0 6 . 2**

SCREEN CAPTURE(S) showing the results of the requested test.

[1 mark]

0	7
---	---

The **Skeleton Program** is to be improved so that when the user loads and takes a quiz the program counts the number of correct answers completed in the quiz and displays this total at the end of the quiz.

Change the `LoadQuiz` subroutine so that as the user completes the quiz it keeps a count of the number of correct answers. When the quiz ends, the following message should be displayed:

```
You managed to get  $X$  correct answers
```

where X is the number of correct answers.

Test your change has worked by running the **Skeleton Program** and then:

- enter 4
- enter 10
- enter 1100
- enter 14
- enter 101101
- enter 40
- enter 1100010

Evidence that you need to provide

Include the following evidence in your Electronic Answer Document.

0	7	.	1
---	---	---	---

All of the PROGRAM SOURCE CODE for the amended subroutine `LoadQuiz`.

[5 marks]

0	7	.	2
---	---	---	---

SCREEN CAPTURE(S) showing the results of the requested test.

[1 mark]

Turn over for the next question

Turn over ►

0	8
---	---

The **Skeleton Program** is to be improved so that it checks the number of places to shift, which is entered after the `Right shift` a binary number option is selected from the menu.

Change the `ShiftRight` subroutine so that:

- the number of places to shift is checked to make sure that it is positive and less than or equal to the number of digits in the binary number entered:
 - if this check passes, the subroutine carries out the required shift using the existing code
 - if this check fails, the message `Invalid Shift Amount` is displayed and the subroutine should exit without carrying out a shift.

Test your change has worked by running the **Skeleton Program** and then:

- enter 5
- enter 110011
- enter -2
- enter 5
- enter 110011
- enter 7
- enter 5
- enter 110011
- enter 3

Evidence that you need to provide

Include the following evidence in your Electronic Answer Document.

0	8	.	1
---	---	---	---

All of the PROGRAM SOURCE CODE for the amended subroutine `ShiftRight`.

[6 marks]

0	8	.	2
---	---	---	---

SCREEN CAPTURE(S) showing the results of the requested test.

[2 marks]

0	9
---	---

The **Skeleton Program** is to be improved so that when a binary number is entered it is checked to ensure that it only contains 1s and 0s.

Change the `GetBinary` subroutine so that after getting the user input it:

- uses a variable `Valid` to represent whether the entered string is a valid binary number or not
- checks all the characters in the entered string and if any character is not a 1 or a 0 it assigns the value `false` to `Valid`
- displays the message `Invalid binary number` if the entered string is not a valid binary number
- returns:
 - the entered string if it is a valid binary number
 - a string consisting of a single 0 if it is not a valid binary number.

Test your changes by running the **Skeleton Program** and then:

- enter 1
- enter 1234
- enter 1
- enter banana
- enter 1
- enter 10010

Evidence that you need to provide

Include the following evidence in your Electronic Answer Document.

0	9	.	1
---	---	---	---

All of the PROGRAM SOURCE CODE for the amended subroutine `GetBinary`.

[10 marks]

0	9	.	2
---	---	---	---

SCREEN CAPTURE(S) showing the results of the requested test.

[1 mark]

Turn over for Section C

Turn over ►

Section C (Longer programming questions)

You are advised to spend no more than **45 minutes** on this section.

1 0

The **Skeleton Program** is to be improved by adding new functionality to the `LoadExamples` subroutine.

Currently, if the subroutine reads in a line starting with B or D it converts the number after the letter on the line from binary to decimal or decimal to binary. If a line does not start with B or D an error message is displayed.

The `LoadExamples` subroutine is to be changed so that if it reads in a line starting with C it converts the binary string after it to a character.

To convert the binary string to a character the subroutine should:

- take the binary string and convert it to a decimal value
- check if the decimal value is between 35 and 126 inclusive:
 - if it is, convert this decimal value to the corresponding ASCII character then display the message
`S represents the character X`
where `S` is the binary string and `X` is the ASCII character
 - if it is not, display the message `Invalid character found`.

Examples of how the subroutine should work are shown in **Figure 2**.

Figure 2

If the line read in from the text file is:

```
C 1100010
```

the message:

```
1100010 represents the character b
```

should be displayed as the decimal value for the binary string 1100010 is 98 and b has the ASCII code 98.

If the line read in from the text file is:

```
C 11111111
```

`Invalid character found` should be displayed as the decimal equivalent of the binary string 11111111 is 255, which is not within the range 35 to 126.

To find the character from a decimal value you can use:

- in C# – the `Convert.ToChar()` function. For example, `Convert.ToChar(98)` is the character `b`
- in Python – the `chr()` function. For example, `chr(98)` is the character `b`
- in VB – the `Chr()` function. For example, `Chr(98)` is the character `b`.

Change the `LoadExamples` subroutine so that lines that begin with `C` are processed as described above.

Test your changes by running the **Skeleton Program** and then:

- enter `6`

Evidence that you need to provide

Include the following evidence in your Electronic Answer Document.

1 0 . 1 All of the PROGRAM SOURCE CODE for the amended subroutine `LoadExamples`.
[11 marks]

1 0 . 2 SCREEN CAPTURE(S) showing the results of the requested test.
[2 marks]

Turn over for the next question

Turn over ►

1	1
---	---

The **Skeleton Program** is to be improved by adding a new subroutine that allows the program to calculate the complement of a binary number.

The complement of a binary number is defined as the number obtained by inverting all the bits in the binary representation of the number. Inverting a bit means that if the bit is a 1 it becomes a 0 and if the bit is a 0 it becomes a 1.

Examples of the complement of two binary numbers are shown in **Figure 3**.

Figure 3

If the binary number is	00110101
then the complement is	11001010
If the binary number is	10101010
then the complement is	01010101

Task 1

Create a new subroutine called `Complement` that:

- has a parameter `BinaryString`
- creates a new string which is the complement of the parameter `BinaryString`
- returns this new string.

Task 2

Adapt the subroutine `DecimalToBinary` so that after displaying the binary string:

- it asks the user if they want to view the complement
- if the user answers Y, it should call the `Complement` subroutine with the appropriate parameter and display the returned value.

Task 3

Test your changes by running the **Skeleton Program** and then:

- enter 2
- enter 100
- enter Y

Evidence that you need to provide

Include the following evidence in your Electronic Answer Document.

- 1 1 . 1** All of the PROGRAM SOURCE CODE for the new subroutine `Complement`. **[10 marks]**
- 1 1 . 2** All of the PROGRAM SOURCE CODE for the amended subroutine `DecimalToBinary`. **[4 marks]**
- 1 1 . 3** SCREEN CAPTURE(S) showing the results of the requested test. **[1 mark]**

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

Copyright information

For confidentiality purposes, all acknowledgements of third-party copyright material are published in a separate booklet. This booklet is published after each live examination series and is available for free download from www.oxfordqaexams.org.uk.

Permission to reproduce all copyright material has been applied for. In some cases, efforts to contact copyright-holders may have been unsuccessful and Oxford International AQA Examinations will be happy to rectify any omissions of acknowledgements. If you have any queries please contact the Copyright Team.

Copyright © 2021 Oxford International AQA Examinations and its licensors. All rights reserved.

