

|  |  |
| --- | --- |
| CANDIDATE NAME: |  |
| CANDIDATE NUMBER: |  |

School Of Coding

**GCSE OCR (9-1) Computer Science**

**Component 2 - Computational Thinking, Algorithms and Programming**

**Time:** 1 hour 30 minutes

**Instructions**

* Use black ink.
* Write your name at the top of this page.
* Answer all questions in the spaces provided.

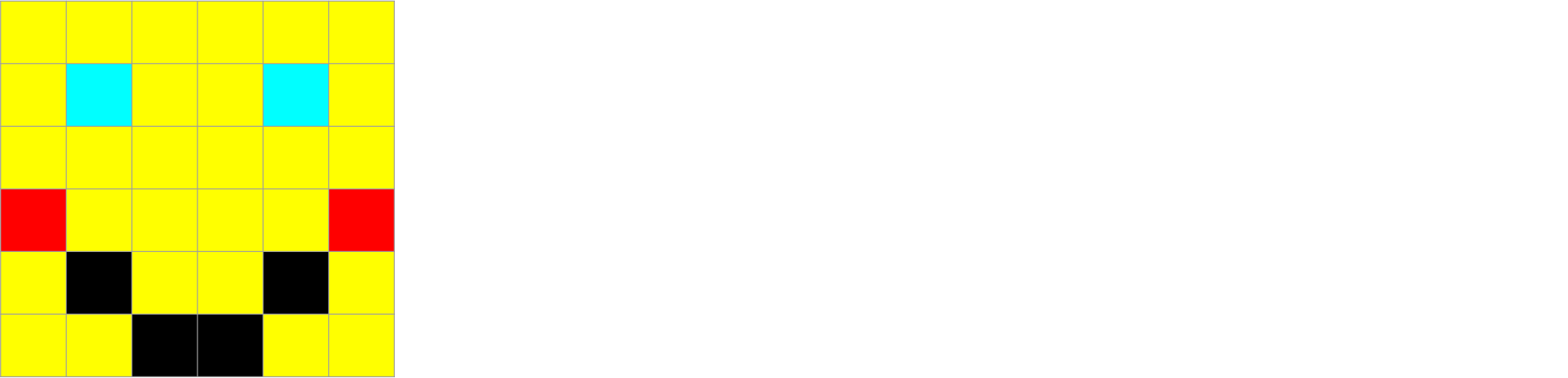
**Information**

* The marks for each question are shown in brackets.
* The maximum mark for this paper is 80.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **FOR EXAMINERS USE ONLY** | | | | | | | | |
| Question | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **Total** |
| Total Marks | 10 | 15 | 8 | 10 | 15 | 7 | 15 | 80 |
| Marks Awarded |  |  |  |  |  |  |  |  |

1. School Of Coding are adding images to their website.

They want the image in **Figure 1** to show up on the screen when a person gets a question wrong, and the image in **Figure 2** to show up on the screen when a person gets a question right.



**Figure 1 Figure 2**

* 1. Explain how a bitmap image is stored on a computer, with reference to the image in **Figure 1.**

[3]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

Two properties of images are colour depth and resolution.

* 1. Define the term “colour depth”.

[1]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

* 1. Define the term “resolution”.

[1]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

* 1. What is the colour depth of the image in **Figure 2**?

[2]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

* 1. Calculate the file size of the image in **Figure 2** in bytes. You do not need to account for metadata. Show your working.

[3]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

1. Molly from School Of Coding is trying to figure out how much each student needs to pay for their lesson.

The data for students is stored in a database. An example of the data held in this system is shown below in Figure 3.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **LessonType** | **NumberOfLessons** | **Paid** |
| Amrit | 1 to 1 | 5 | FALSE |
| Bob | 1 to 1 | 4 | FALSE |
| Crystal | Group | 7 | FALSE |
| Dev | Group | 9 | FALSE |
| Ethan | 1 to 1 | 12 | FALSE |

**Figure 3**

* 1. Write an SQL statement that will return the name of all students who have group lessons. The name of the table is Lessons.

[3]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

* 1. Name the data type that would be most suitable for each field:

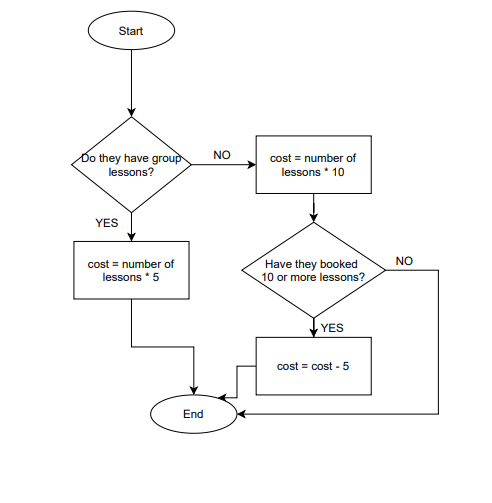
[3]

Name: ...…………………………………………………………………………………………………

NumberOfLessons:……………………………………………………………………….……………

Paid:………..……....……………………………………………………………………………………

Molly is using an algorithm to work out how much each student should pay for their lessons. The algorithm is shown as a flowchart in the diagram below.

* 1. Using data from **Figure 3**, what would be the cost Ethan has to pay?

[1]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

* 1. What would be the cost Crystal has to pay?

[1]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

* 1. The costs that a number of different people have to pay have been calculated and stored in an array named payments. The contents of the array are shown below.

**The value of payments[0] is 25.**

payments = [25, 45, 60, 95, 105, 30, 50]

Write an algorithm that:

* Outputs the sum of all the numbers in the array
* Outputs the highest number in the array

[7]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

* 1. Convert the denary number 63 into a binary number. Show your working.

[2]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

* 1. Convert the binary number 00111101 into a hexadecimal number. Show your working.

[2]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

* 1. Add the two following 8 bit numbers:

[2]

* 1. When adding the two 8 bit numbers 1101 1000 + 0110 0000, an overflow error occurs. Describe what is meant by an overflow error.

[2]

……………………………………………………………………………………………………………

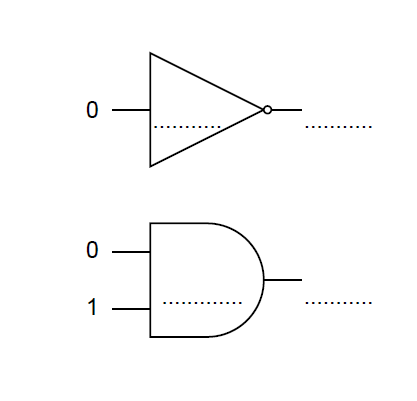
……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

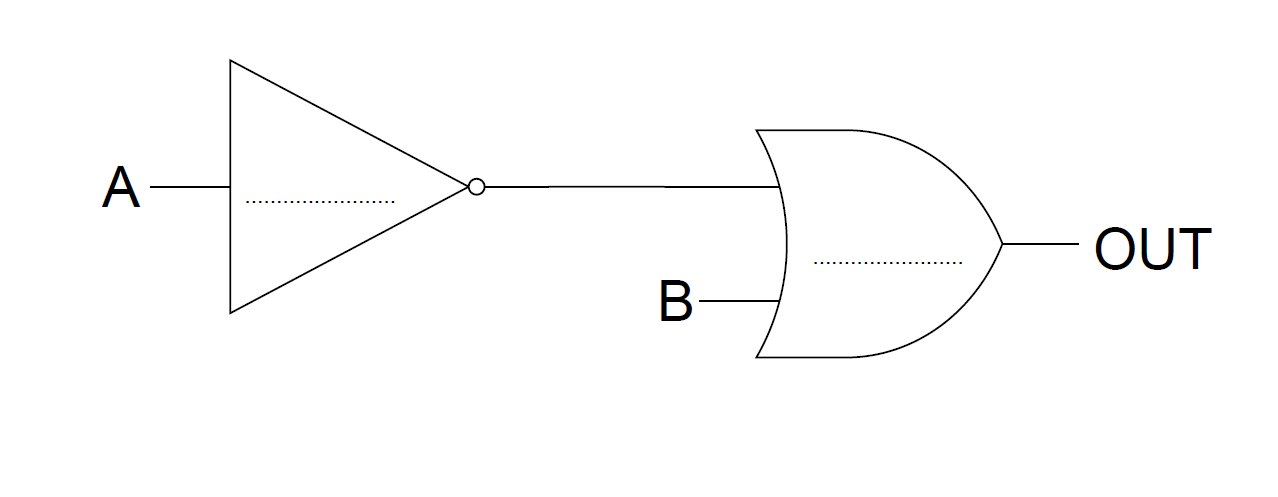
……………………………………………………………………………………………………………

* 1. Fill in the name and the output for each logic symbol below.

[4]



* 1. Fill in the names for each symbol in **Figure 4**, then complete the truth table for the logic diagram in **Figure 4**.

[6]

**Figure 4**

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **OUT** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Emily has written a program in pseudocode for a retail store. The algorithm’s purpose is to ask the manager for the total earnings and the total costs for every day in a week. The algorithm will then add up the total profits for the week and output it after all 7 days have been entered.
2. for x = 1 to 7
3. totalProfit = 0
4. dayEarnings = int(input(“Enter the earnings for day “,x))
5. dayCosts = int(input(“Enter the costs for day”,x))
6. dayProfit = dayEarnings - dayCost
7. totalProfit = totalProfit + dayProfit
8. Primt(“Weekly profit is: “, totalProfit)
   1. This program contains a logic error and a syntax error.
      1. Explain what is meant by a logic error, and explain how Emily has made a logic error in the above program.

[3]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

* + 1. State which line contains a syntax error, and rewrite this line so that the syntax error is corrected.

[2]

Line Number:………………….………………………………………………………………….…….

Rewritten Line:.…………………………………………………………………………………………

* 1. Explain what is meant by a variable and give an example from the algorithm above.

[3]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

* 1. Complete the table to show whether the algorithm above uses, sequence, selection or iteration. You may tick more than one box.

[3]

|  |  |
| --- | --- |
| **Sequence** |  |
| **Selection** |  |
| **Iteration** |  |

* 1. Emily wants to test this algorithm.
     1. Give one example of normal test data for the daily earnings.

[1]

……………………………………………………………………………………………………………

* + 1. Give an example of erroneous test data for the daily earnings.

[1]

……………………………………………………………………………………………………………

* 1. Emily has used indentation to make her program more maintainable. Name one other thing that she can use to improve the maintainability of her program, and explain how this makes it more maintainable.

[2]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

* 1. Show the stages of a merge sort to sort the following numbers:

**32,6,7,90,30,2,17,102**

[5]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

* 1. A bubble sort is another type of sorting algorithm. State one benefit of using a bubble sort over a merge sort.

[1]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

* 1. Name one benefit of using a merge sort over a bubble sort.

[1]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

1. Gus needs to create an online system for School Of Coding to grade students. He will be using defensive design considerations when creating his program as he doesn’t want his program to crash easily.
   1. Name 2 defensive design considerations.

[2]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

* 1. Gus uses an IDE (Integrated Development Environment) to create his program. Name 2 features of an IDE and explain how each feature aids you when creating a program.

[4]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

* 1. One part of the online system needs to set up login details for new students.

Students have a username that is generated from parts of their first and last name, as well as their birth year. Students must also create a password which must be at least 10 characters long.

Write an algorithm that:

* Asks the user for their first name and surname.
* Creates a username by getting the first 2 letters of their first name and the first 3 letters of their surname.
* Outputs the created username.
* Asks the user to enter a password.
* If the password is less than 10 characters, output the message “too short”, otherwise output “account created”.

[9]

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

**END OF QUESTION PAPER**