

2014 Paper Model solution

NOTE: The paper itself can be obtained from the AQA exam board website, [here](#). The official mark scheme (that the examiners themselves used when marking the paper) can be picked up from [here](#).

1a. Binary to denary conversion.

128	64	32	16	8	4	2	1
1	0	0	1	0	1	1	1

Check: $128 + 16 + 4 + 2 + 1 = \mathbf{151}_{10}$

1b. 125_{10} in hexadecimal...

Step 1: Convert into a binary number...

$125 - 128$	< 0	so...	0
$125 - 64$	$= 61$	so...	1
$61 - 32$	$= 29$	so...	1
$29 - 16$	$= 13$	so...	1
$13 - 8$	$= 5$	so...	1
$5 - 4$	$= 1$	so...	1
$1 - 2$	< 0	so...	0
$1 - 1$	$= 0$	so...	1

125 in denary is the binary number: 0111 1101.

Step 2: Draw a conversion table (it's slow, but I personally do it this way, as I know I'll have the right answer)

Binary	Dec	Hex
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

Step 3: Look up the appropriate nibbles in the table...

0111 = 7, 1101 = D. **ANSWER: 7D**

1c. Hexadecimal allows a 4-bit nibble to be encoded in a single (human-friendly) character. Programmers entering hex characters will be faster than those entering binary data.

1d. If ASCII uses 7 bits to encode characters, then there are 128 (2^7) different combinations that can be assigned to characters.

1e. To convert 'real-world' sound into a digital format that a computer can understand, the order is as follows:

1. Microphone picks up the sound waves.
2. They are converted into an analogue electrical signal.
3. The value of the signal is read at specific points and rounded to a level.
4. A binary representation of this level is stored.

1f. A black and white image only requires a single bit to represent either black (0) or white (1). As such, a data stream of 0s and 1s could be used to render an image.

Notes: If had been asked about a 4-colour system, I'd need 2 bits per pixel to provide me with 4 different binary combinations to express the colours (i.e. 00, 01, 10 and 11).

Most images in the real world use 24-bit colour depth. 8 bits each are assigned to the amount of red, green and blue to be used in each pixel. This allows a possible colour palette of around 16.7 million (16,777,216 to be precise) different colours.

2ai. Volatile memory is that which is wiped when the power supply is disconnected; RAM is volatile, for instance.

2a.ii. In a desktop PC or laptop, the BIOS (startup instructions) are stored in non-volatile memory.

2b. Cache memory is an ultra-fast small amount of memory stored on the CPU. This stores frequently accessed instructions allowing the CPU to access it directly. As cache works faster than RAM, it improves the performance of the CPU.

2c. To improve CPU performance...

CPUs with additional cores will be faster when performing several tasks simultaneously.
CPUs with higher clock speeds (measured in GHz) will perform tasks more quickly.

Notes: You could also go for...

CPUs with more modern (i.e. faster) cache memory can deliver frequently accessed instructions to the CPU cores faster, so will perform more quickly.

CPUs with a larger word-size (e.g. 64-bit) can handle larger numbers of bits per instruction.

CPUs with a more modern architecture (e.g. more efficiently designed) will process instructions more efficiently.

3a. The function takes one parameter: **arr**.

3b. Iteration is a type of loop (a FOR loop). This starts on line **3**.

3c. Selection is the use of conditionals (IF statements) in a program. This starts on line 2.

3d. Variable assignment is assigning a value to a variable. This first happens on line 2.

3e. Scope refers to where in a program a particular variable can be accessed. In this specific case, 'i' only exists after it is created in the FOR loop in line 3. Once the loop has completed its work and moved beyond line 7, it is no longer available to use.

Notes: At a whole-program level, we often talk about a variable having either global or local scope. In a lot of languages, global variables are declared at the top of a program. These can be accessed inside any function or part of a program; this is convenient, as commonly used data is readily available. At the same time, there is always a risk that other parts of a program might inadvertently change the value of a variable and cause an error.

Local variables are those declared (created) inside a function. These can be accessed and changed inside the function itself, but as soon as the function finishes, the variable ceases to exist. Trying to access it will cause a runtime error.

4a. A network is a collection of computers that are connected together to allow them to communicate and share resources (*like files, printers, Internet access, scanners*).

4b. Client/Server

Displaying an HTML page is done by the web browser on the **client** machine.

Receiving messages is performed by **both** client and server (the server sends the appropriate message, the client receives them).

Starting the handshaking process is performed by the **client**, which requests a connection to the server.

5a. Physical devices that allow data to be input into a tablet computer are:

1. A microphone to allow sound to be entered.
2. Hardware buttons (E.g. Power, Vol+, Home) to access commonly used features.
3. Camera to allow photos/video to be captured.

Notes: You could also have had GPS receiver for location data or an accelerometer to detect the orientation of the device (also commonly used to control games/apps)

5b. Two differences that make Solid State media (SSD drives) better than magnetic media:

1. SSDs contain no moving parts, so can be used while a user is walking/moving the device around. Mechanical hard drives rely on a high-speed motor spinning the disk platter that can be damaged if moved knocked while in motion.
2. An added bonus of having no moving parts is that SSDs can read and write data more quickly than a mechanical drive.

Notes: You could also have gone for...

3. *Because they contain no moving parts, SSDs are silent when they operate.*
4. *Because they contain no moving parts, SSDs are physically smaller than a mechanical hard disk, and so can lead to smaller consumer devices (like 'phones).*

6. Optical media all use light; typically from a laser to read data.

Using playing music from a CD as an example, the disk is spun, and a laser/ tiny opto-electric sensor pair are slowly moved from the centre of the disk to the outside (the data are arranged in a gradual spiral). The surface of the media is covered in pits and bumps which the laser light beam strikes while the disk is spun. Where the laser beam strikes a 'land', the light beam is reflected onto the opto-electric sensor and this is read as a '1'. When the laser-beam strikes a pit, the beam is scattered and not detected by the sensor, denoting a '0'.

7. In the 'Compare' function above, there are three possible outcomes – 'x is larger than y', 'x is smaller than y' or 'x is the same as y'. This requires three possible return values, but a Boolean variable can only supply 2 (True or False).

7bi. Doing Compare(4,4) would return **0** because neither of the IF statements are true.

7bii. Doing Compare(1,-1) would return **1** because x (1) is larger than y (-1).

7biii. Doing Compare(Compare(4,4),Compare(1,-1)) would first evaluate to: Compare (0,1), based on the results from Q7bi and Q7bii. This would return **-1**, as x (0) is less than y (1).

Note: As these are only 1 mark questions that say "Give the Integer value", you don't need the commentary. I just included it to be helpful.

8a. Database terms...

A row of data within a table is called a record (Option E).

A Primary Key (D) uniquely identifies a row of data.

A Relationship (F) links between tables.

An index (C) is used to speed up searches.

Notes: While we're at it...

A field is the term for a single column of data in a table (e.g. FirstName, ID, DOB)

A foreign key is a link to a primary key in another table (needed for relationships)

8b. Option 3.

Notes: Let's see why the others were wrong.

WHERE DayTaught = Monday OR DayTaught = Wednesday

The DayTaught field is a string, so any search term needs to be wrapped in speech marks so the SQL parser can see where the term to be searched for ends.

WHERE DayTaught = 'Monday' OR 'Wednesday'

This says that the DayTaught must be Monday, and we use the Boolean OR operator, but the search term by itself doesn't mean anything. It should be in the form <FIELDNAME> = 'Wednesday'.

WHERE DayTaught = 'Monday' OR DayTaught = 'Wednesday'

The correct answer. We've used quotes around our string search term, used the correct logical operator (OR, as per the wording of the question on the paper) AND we've avoiding the mistake from the previous option.

WHERE DayTaught = 'Monday' AND DayTaught = 'Wednesday'

This says that the DayTaught must be both Monday and Wednesday simultaneously. As we can only store one of these values in the record, no record could ever match this.

8bii.

Notes: Here's how I did it...

SELECT Class.ClassID, Class.DayTaught

I know that whatever the result is, these will be the column headings. I can sketch the table format of the answer straight away.

Class.ClassID	Class.DayTaught

FROM Class, Teacher

I know that to complete this query, I'll be looking in both these tables. Noted.

WHERE Teacher.Room = '405' AND Class.TeacherID = Teacher.TeacherID

I deal with the relationship part of this first. As the command is saying that the TeacherID field in both class and Teacher are referring to the same thing, I can now mentally splice the two tables together.

As I look at the first row of the Class table, I imagine the row has two new columns (Surname and Room) on the end of it. As this row has the TeacherID, "tur", I add Turing and 405 to the end.

The second row (ClassID: Blue7) has now got Milner and 406 on the end, and so on...

Class.ClassID	Class.Size	Class.DayTaught	Class.TeacherID	Teacher.Surname	Teacher.Room
Red7	24	Monday	Tur	Turing	405
Blue7	29	Tuesday	Mil	Milner	406
Yellow7	29	Wednesday	Mcc	McCarthy	412a
Red8	31	Wednesday	Mcc	McCarthy	412a
Red9	26	Thursday	Mcc	McCarthy	412a
Blue9	24	Thursday	Mil	Milner	412a
CompA10	17	Thursday	Mil	Milner	406
CompB10	19	Friday	tur	Turing	405

I then think about the other part of the WHERE statement: **WHERE Teacher.room='405'**

This tells me to only pick out rows where Teacher.Room is 405. This only matches 2 rows.

Class.ClassID	Class.Size	Class.DayTaught	Class.TeacherID	Teacher.Surname	Teacher.Room
Red7	24	Monday	Tur	Turing	405

CompB10	19	Friday	tur	Turing	405
---------	----	--------	-----	--------	-----

Finally, I'm told what order to output the data in: **ORDER BY Class.ClassID ASC**

So I'm to populate the table I drew up earlier with these two rows, in ascending order, sorted by the ClassID. Therefore, my answer is:

Class.ClassID	Class.DayTaught
CompB10	Friday
Red7	Monday

9ai. This is a logical error.

Notes: Let's recap...

Logical errors are ones which the program will run, but as the programmer's logic is flawed (like in the example in the question), you'll get the wrong result. These can be hard to track down in larger programs.

Runtime errors are where you ask the computer to do something it can't. Divide by zero, find a root of a negative number or return element 10 of an array with 5 elements in it for instance.

Syntax errors are where the programmer mis-types a command word (e.g. PrOnt("")).

9aai. It's line 4.

By taking 3 away from whatever number you feed repeatedly as long as the number you get is greater than 0, if the number you eventually end up with IS 0, then the number is divisible by 3. Otherwise, there's a remainder, and therefore the number isn't divisible.

9aiii. Change from WHILE n >= 0 to WHILE n>0.

9b. A runtime error. The computer can't treat 'eight' (a string) as 8 (an integer).

9c. Tools to help reduce the change of errors in a program include:

Using a variable watch.

Using a text editor that supports syntax highlighting.

Using an IDE with a step-through feature.

Notes: Here's a little more detail on the common ones.

A variable watch allows the programmer to see the values of the different variables in their program as it runs. This is helpful when debugging.

Syntax highlighting helps the developer see when they make syntax errors. You see this in IDLE when writing Python code. Comments are always coloured red, functions (like print()) are coloured purple, strings are green, etc). This helps show when typos are made, as the colours stop appearing as expected.

Break points are where a program can be run up to a certain line of code, then paused at that point. This is normally used with variable watch, allowing the developer to see the values of all the variables at that point in the program.

Step through is commonly used with break points and a variable watch. This is where the program is executed one line at a time, so that the developer can see the flow of the program and look at the variable contents slowly to ensure the flow of the program is as intended.

Code completion is where a developer can type the first few characters of a command word, then tap an auto-completion key (often Tab) to have the computer write the full command word out; this reduces the risk of typos.

10. **Unit testing** looks at the correctness of small blocks of code (e.g. individual functions). They are performed by the programmer as they write their program. This can reveal logical errors and syntax errors in the code that will prevent the module/function from working as intended.

Integration testing happens when all of the different parts of code, such as functions or modules, are complete. This tests whether the units of code work together correctly, and can also be performed by the programmer. This testing could highlight logical errors that happen as a result of one function passing the 'wrong' result into another. It may also show runtime errors if seemingly correct results cause future functions to attempt to divide by zero, for instance.

System testing takes place after the code has been individually tested and is done without having to have any knowledge of the code itself (black box testing). This tests that the entire system functions correctly. This could be performed by the developer or by the client before project sign-off. This could demonstrate whether or not the finished system meets the original client brief.

11. External code sources are those which someone else has written (often part of a community effort) and released for others to use.

One advantage of using external code is that as the work has likely been developed by a large number of people and extensively peer reviewed, it is likely to be robust and reliable.

Another advantage is time saving. Writing complex functionality into a program is highly time consuming (e.g. to use a specialised sorting algorithm), and using someone else's proven solution will allow the developer to move on to another part of their program, reducing development time.

Another advantage is that not all developers are experts in all areas of CS. If a developer wanted to send data over a network (for instance), having access to a set of functions that allows that would let the developer use features they otherwise wouldn't be able to implement.

Notes: Let's keep going...

A disadvantage is that with someone else's code, the developer is dependent on the author of the code for future updates and improvements. If there is a bug in the code, it'll be very hard (or perhaps impossible) to fix it without the original author(s) help.

Another disadvantage is that the code brought in from an external source may not exactly meet the needs of the developer, and as such the developer may have to compromise on what they want to achieve.

Another disadvantage is that not all external code is freely available, and may need paying for, which will increase development cost.

carriages	total	max	index
0	0	0	1
	100	100	2
	80		3
	150	150	4
	100		5
	0		6
3			

Notes: There are plenty of practice exercises completing trace tables over at www.BourneToCode.com in the GCSE notes that you can look at. Work through these until you feel confident at doing them.

12b. My algorithm...

```
journeyLength <-- INPUT "How many kilometres?"
```

```
IF journeyLength > 0 THEN  
    fuelAmount = journeyLength * 100
```

```
    IF fuelAmount < 1500 THEN  
        fuelAmount <-- 1500  
    END IF
```

```
    OUTPUT fuelAmount  
ENDIF
```

Notes: Read, then re-read any 'write an algorithm to...' questions to ensure you're providing exactly the functionality the examiner is asking for. When I wrote my initial solution, I set up a WHILE loop to keep prompting the user until they entered a value greater than 0. This isn't what the question asks for, and would have cost me a mark!

Also check your logic when writing conditionals (IF statements) carefully. Do you need to use >, >=, =, <, <= or <> (Not equal to) for the task at hand?