SATs·Papers.co.uk

PAST PAPERS · QUESTION PAPERS · MARK SCHEMES & ANSWERS · PAPER NOTES

16+ PAST PAPER PACK

Rugby School 16+ Computer Science

Complete Past Paper Pack

CONTENTS

RUGBY SCHOOL

Sixth Form Entrance Examination

COMPUTER SCIENCE SPECIMEN PAPER

Time allowed: **1 hour**

Name: _____

INSTRUCTIONS TO CANDIDATES

- Write your name above.
- Do not open the paper until instructed to do so.
- A calculator **is not allowed** for this paper.
- Write clearly and fully where the questions request it.
- There are three sections in this paper. You must answer all questions in each section.
- The maximum mark for this paper is [**55** marks]

Section A - Hardware and Software

1. Describe the difference between **primary** and **secondary** storage in a computer. Provide one example for each. **[4]**

2. Give three examples of the roles of an operating system. **[3]**

3. (a) Match the following type of media to their corresponding average capacity. **[3]**

• BluRay Disk	• 512 GB - 6 TB
• Hard Disk	• 4.7 GB - 8.5 GB
• DVD	• 25 GB - 50 GB
• Solid State Drive	• 4GB - 2 TB
• CD-ROM	• 700 MB

(b) State what the abbreviation **ROM** stands for in "CD-ROM". **[1]**

(c) Arrange the following storage types from slowest to fastest access speed: magnetic, optical and solid state. **[2]**

(d) Explain why you would expect a BluRay disk to have a higher access speed than a CD-ROM. **[2]**

4. Extra information stored with an image is called *metadata*. Give two examples of image metadata. **[2]**

5. (a) Define what a CPU is in a computer. **[1]**

(b) Briefly explain the steps involved in the Fetch-Decode-Execute cycle. **[4]**

6. Ryan brags that he uses his new headphones to only listen to "FLAC, because it is a lossless file format". Explain what Ryan means by *lossless file format*. **[3]**

Turn over for next section

Section B - Programming

7. Given that $a = 7$ and $b = 6$, state what appears on the screen when each of the following snippets of code is executed.

(a)
```
if a > 7 or b = 1 then
    print("Yellow")
else
    print("Blue")
end if
```

[1]

(b)
```
if NOT(a > b) or (b ≤ 7) then
    print("Red")
else if NOT(a > b) or (b ≥ 7) then
    print("Blue")
else
    print("Green")
end if
```

[1]

8. The following pseudocode is intended to find and display the largest number in an array of ten positive integers.

```
1: max = numbers[0]
2: for n = 0 to 9 do
3:     if max > numbers[n] then
4:         max = numbers[n]
5:     end if
6: end for
7: print(max)
```

The pseudocode contains an error and does not work as intended. State the line of code that contains the error and suggest a correction. **[2]**

9. Describe and explain two advantages of writing code using sub-routines. **[4]**

10. An estate agent keeps details of all the properties they have available for rent.

PropertyID	Type	MonthlyRent	Beds	Furnished	DistanceToStation
1	Apartment	£800.00	2	Y	0.3
2	Semi	£475.00	2	N	1.5
3	Apartment	£1150.00	3	N	0.5
4	House	£1500.00	4	Y	0.2
5	Apartment	£900.00	2	Y	0.3
6	Apartment	£1250.00	3	Y	0.2
7	Semi	£550.00	3	Y	2.4
8	House	£600.00	3	N	0.6

List the Property IDs of the properties that will be found by the following SQL queries. **[4]**

(a) SELECT *
FROM tblRental
WHERE MonthlyRent <= 550.00 OR Furnished = 'Y'

(b) SELECT *
FROM tblRental
WHERE Type = "Apartment" AND DistanceToStation < 0.3

11. Jimmy produces the following algorithm.

```
1: limit = input("Please enter an upper limit")
2: x = 0
3: while x < limit do
4:     print(x)
5:     x = x + 2
6: end while
```

(a) Write down the outputs for the algorithm for an input of 9. **[2]**

(b) Jimmy intended this algorithm to print the first 9 non-negative even numbers in this case. Explain what this algorithm does instead. **[2]**

(c) Suggest how line 4 and line 5 can be changed to make the algorithm work as intended without changing any other part of it. **[3]**

12. Write pseudocode that will perform the following:

Ask a user to enter a number.

If the number is between 0 and 10, output the word **blue**.

If the number is between 10 and 20, output the word **red**.

If the number is between 20 and 30, output the word **green**.

If it is not in the accepted ranges above, output a message to say that this is not a correct colour option. **[6]**

Turn over for next section

Section C - Implications of Computer Use

13. Artificial Intelligence is increasingly used in everyday life of people, but also at more technical levels to provide advice on medical, financial and other matters.

(a) Provide an example of how AI is used in everyday life, or at a technical level. **[1]**

(b) Discuss the potential ethical and cultural issues associated with the application you have stated. Marks will be awarded for clarity of argument and knowledge of relevant information. **[4]**

End of paper

Paper Notes: 16+ Computer Science Specimen Paper (16+ Computer Science Specimen Paper)

Compiled by SATs-Papers.co.uk to help you get the most from this paper.

Overview

This is a **Sixth Form Entrance Examination** specimen paper in **Computer Science**, published by **Rugby School** for candidates seeking entry into Year 12. The paper carries a total of **55 marks** and is designed to be completed in **one hour** without a calculator.

The paper tests a broad range of computer science fundamentals across three distinct sections: **Section A** covers hardware and software concepts including storage types, operating systems, and the CPU fetch-decode-execute cycle; **Section B** focuses on programming logic, pseudocode analysis, SQL queries, and algorithm debugging; **Section C** examines the implications of computer use, specifically the ethical and cultural dimensions of artificial intelligence applications.

This specimen paper is particularly useful for students preparing for Rugby School's Sixth Form entry, as it provides a clear indication of the level of knowledge and analytical thinking expected. The mix of short-answer questions, technical explanations, and extended written responses means candidates must demonstrate both recall of facts and the ability to discuss complex issues in a structured manner.

How this paper is organised

The paper comprises **three sections** with a total of **13 questions**, all of which must be answered. Marks are allocated question by question, with individual sub-questions weighted according to complexity, ranging from single-mark definitions to six-mark programming tasks.

Section A (Hardware and Software) contains six questions totalling approximately 25 marks and covers storage, operating systems, the CPU, and file formats. **Section B (Programming)** features six questions worth around 25 marks and includes code tracing, pseudocode debugging, SQL queries, and algorithm design. **Section C (Implications of Computer Use)** consists of one extended question worth 5 marks, requiring an example of AI use followed by a discussion of ethical and cultural issues.

Candidates must write clearly and fully where requested, and the instructions emphasise that all questions are compulsory. The paper layout is clean, with generous space for written responses and clear subsection labelling throughout.

Topics covered

- Primary versus secondary storage, including definitions, examples (RAM, hard disk), and differences in speed and volatility
- Roles and functions of an operating system, such as memory management, file handling, and user interface provision
- Storage media capacities and characteristics, including BluRay, DVD, CD-ROM, hard disk drives, and solid state drives
- Storage access speeds, comparing magnetic, optical, and solid state technologies, with explanations of why newer media formats are faster
- Central Processing Unit (CPU) definition and the Fetch-Decode-Execute cycle, explaining each stage in the instruction processing sequence
- Metadata associated with digital images, such as date taken, camera model, GPS coordinates, and image resolution
- Lossless versus lossy file compression, with specific reference to FLAC and the concept of perfect data reconstruction
- Pseudocode and algorithm logic, including conditional statements (if/else), Boolean operators (AND, OR, NOT), and loop structures (while, for)
- Debugging and error correction in pseudocode, particularly comparison operators and loop conditions that produce unintended outputs
- SQL query construction and interpretation, including SELECT statements, WHERE clauses, comparison operators, and logical conditions (AND, OR)
- Subroutines and modular programming, discussing benefits such as code reusability, maintainability, and ease of debugging
- Algorithm design using nested conditionals and range checking to produce structured output based on user input
- Ethical and cultural implications of artificial intelligence, covering topics such as bias, accountability, privacy, and societal impact in medical, financial, or everyday applications

How to use this paper for revision

- Revise the fundamental differences between primary storage (fast, volatile, directly accessible by the CPU) and secondary storage (slower, persistent, used for long-term data retention).
- Practise tracing through pseudocode step by step with given values to predict outputs, paying close attention to the order in which conditions are evaluated.
- Memorise the stages of the Fetch-Decode-Execute cycle and be able to describe what happens in each stage, including the role of registers and the control unit.
- Familiarise yourself with SQL syntax, particularly the WHERE clause and how AND and OR operators combine multiple conditions to filter database records.
- Work through algorithm debugging exercises by testing code with sample inputs and identifying where logic fails to produce the intended result.
- Review the ethical dimensions of AI applications, preparing to discuss issues such as algorithmic bias, transparency, job displacement, and data privacy with clear examples.
- Make sure you understand lossless versus lossy compression, and be able to explain why certain file formats (FLAC, PNG) preserve original data while others (MP3, JPEG) do not.

Common mistakes to avoid

- Confusing the comparison operators in pseudocode, especially mixing up 'greater than' (>) with 'less than' (<) when checking for maximum or minimum values.
- Forgetting that OR conditions evaluate to true if either condition is met, leading to incorrect predictions about which branch of an if-statement will execute.
- Misunderstanding the loop termination condition, such as assuming 'while x < limit' will iterate 'limit' times when in fact it depends on how x changes inside the loop.
- Listing only superficial ethical issues for AI (e.g. 'privacy') without explaining how or why they arise in the specific application mentioned.
- Writing SQL queries without carefully checking the logical combination of conditions, particularly when AND and OR are used together without parentheses for clarity.
- Confusing ROM (Read-Only Memory) with RAM, or not recognising that CD-ROM refers to a disc that can be read but not written to by standard drives.

Exam technique

Begin by reading through the entire paper to identify questions where you feel most confident, then tackle those first to secure early marks and build momentum. For **Section A**, concise technical definitions are expected, so avoid lengthy waffle and focus on accuracy and key terms such as 'volatile', 'non-volatile', 'access speed', and 'instruction cycle'.

In **Section B**, show your working when tracing code or debugging pseudocode. Write down intermediate values or conditions so that even if your final answer is incorrect, you can earn method marks. For the SQL questions, double-check each condition in the WHERE clause and think through which rows will match before writing your answer. When designing pseudocode in Question 12, structure your code with clear indentation and logical flow, testing mentally with boundary values (0, 10, 20, 30) to ensure each range is handled correctly.

Section C awards marks for clarity of argument and knowledge, so plan a short structure before writing. State your AI example clearly, then organise your discussion into distinct points (e.g. bias, accountability, cultural differences in privacy expectations) with evidence or reasoning for each. Aim for two or three well-developed points rather than a list of superficial mentions.

What to revise alongside this paper

Students preparing for this paper should also revise **data representation**, including binary, hexadecimal, and how text, images, and sound are encoded digitally. Understanding the Von Neumann architecture and the role of registers, the ALU, and buses will deepen your grasp of the CPU and fetch-decode-execute cycle.

For programming, practise writing and testing more complex algorithms involving nested loops, arrays, and functions. Familiarity with a real programming language (Python, Java, or C#) will help you translate pseudocode concepts into working code and vice versa. Review **data structures** such as lists, stacks, and queues, and consider how they are used in algorithm design.

Finally, explore wider **computing ethics** topics, including the impact of automation on employment, data protection legislation (GDPR), and the role of computing in society. Reading case studies of AI in healthcare, criminal justice, or autonomous vehicles will provide concrete examples to draw on in extended written responses.

Key terms

Primary storage, **Secondary storage**, **Operating system**, **CPU (Central Processing Unit)**, **Fetch-Decode-Execute cycle**, **Metadata**, **Lossless compression**, **Lossy compression**, **Pseudocode**, **Boolean operators (AND, OR, NOT)**, **SQL (Structured Query Language)**, **Subroutine**, **Algorithm**, **Ethical issues**, **Artificial Intelligence (AI)**

RUGBY SCHOOL

Sixth Form Entrance Examination

COMPUTER SCIENCE SPECIMEN PAPER

MARK SCHEME

Section A - Hardware and Software

1. Describe the difference between **primary** and **secondary** storage in a computer. Provide one example for each. **[4]**

   Primary is **volatile** eg RAM/Cache

   Secondary is **non-volatile**, e.g. HDD/SSD/Optical

   1 mark for discussing persistence/volatility or another applicable difference for each, 1 mark for a suitable example of each

2. Give three examples of the roles of an operating system. **[3]**

   Management of resources, Management of peripheral devices, Management of memory, Provision of a user interface or equivalent.

   1 mark for each example, up to a max. of 3 marks

3. (a) Match the following type of media to their corresponding average capacity. **[3]**

• BluRay Disk	• 512 GB - 6 TB
• Hard Disk	• 4.7 GB - 8.5 GB
• DVD	• 25 GB - 50 GB
• Solid State Drive	• 4GB - 2 TB
• CD-ROM	• 700 MB

   BluRay Disk -> 25 GB - 50 GB
   Hard Disk -> 512 GB - 6 TB
   DVD -> 4.7 GB - 8.5 GB
   Solid State Drive -> 4GB - 2 TB
   CD-ROM -> 700 MB

   3 marks for all correct, 2 marks for 3 correct, 1 mark for 2 correct

   (b) State what the abbreviation **ROM** stands for in "CD-ROM". **[1]**

   Read only memory

   1 mark

   (c) Arrange the following storage types from slowest to fastest access speed: magnetic, optical and solid state. **[2]**

   Optical -> Magnetic -> Solid State

   1 mark per each correct comparison

   (d) Explain why you would expect a BluRay disk to have a higher access speed than a CD-ROM. **[2]**

   A BluRay needs to hold more data as the quality of the media stored on a BluRay disk is generally higher. This data is to be accessed in the same time frame as the data on a CD-ROM so the access speed needs to be faster.

   2 marks for any reasonable description

4. Extra information stored with an image is called *metadata*. Give two examples of image metadata. **[2]**
   e.g. resolution, colour depth, date, etc

   1 mark for each correct example of image metadata

5. (a) Define what a CPU is in a computer. **[1]**

- Central Processing Unit
- The CPU is the primary component of a computer that processes instructions.

1 mark per bullet point (or equivalent description) to a max of 1 mark.

(b) Briefly explain the steps involved in the Fetch-Decode-Execute cycle. **[4]**

- Instructions (or data) are fetched from memory
- The instruction is decoded by the Control Unit
- ….and executed
- These steps are repeated.

1 mark per bullet point

6. Ryan brags that he uses his new headphones to only listen to "FLAC, because it is a lossless file format". Explain what Ryan means by *lossless file format*. **[3]**

The data file has been compressed…
This reduces the size of the file
There is no loss of audible information as the file can be decrypted to replicate the original.

1 mark per bullet point (or equivalent description) to a max of 3 marks.

Turn over for next section

Section B - Programming

7. Given that $a = 7$ and $b = 6$, state what appears on the screen when each of the following snippets of code is executed.

(a)
```
if a > 7 or b = 1 then
    print("Yellow")
else
    print("Blue")
end if
```

[1]

Blue

1 mark

(b)
```
if NOT(a > b) or (b ≤ 7) then
    print("Red")
else if NOT(a > b) or (b ≥ 7) then
    print("Blue")
else
    print("Green")
end if
```

[1]

Red

1 mark

8. The following pseudocode is intended to find and display the largest number in an array of ten positive integers.

```
1: max = numbers[0]
2: for n = 0 to 9 do
3:     if max > numbers[n] then
4:         max = numbers[n]
5:     end if
6: end for
7: print(max)
```

The pseudocode contains an error and does not work as intended. State the line of code that contains the error and suggest a correction. **[2]**

Line 3: if max<number[n] then

1 mark correct line number and 1 mark for the correct code

9. Describe and explain two advantages of writing code using sub-routines.

[4]

Reusable code
…….reduces development time
Simplifies testing
……can be tested in isolation from the rest of the program
Easier to maintain
……smaller sections can be readily understood by other programmers

1 mark for each description to a max of 2 marks and 1 mark for an appropriate explanation to a max of 2 marks

10. An estate agent keeps details of all the properties they have available for rent.

PropertyID	Type	MonthlyRent	Beds	Furnished	DistanceToStation
1	Apartment	£800.00	2	Y	0.3
2	Semi	£475.00	2	N	1.5
3	Apartment	£1150.00	3	N	0.5
4	House	£1500.00	4	Y	0.2
5	Apartment	£900.00	2	Y	0.3
6	Apartment	£1250.00	3	Y	0.2
7	Semi	£550.00	3	Y	2.4
8	House	£600.00	3	N	0.6

List the Property IDs of the properties that will be found by the following SQL queries. **[4]**

(a) SELECT *
FROM tblRental
WHERE MonthlyRent <= 550.00 OR Furnished = 'Y'

1	Apartment	£800.00	2	Y	0.3
2	Semi	£475.00	2	N	1.5
4	House	£1500.00	4	Y	0.2
5	Apartment	£900.00	2	Y	0.3
6	Apartment	£1250.00	3	Y	0.2
7	Semi	£550.00	3	Y	2.4

(b) SELECT *
FROM tblRental
WHERE Type = "Apartment" AND DistanceToStation < 0.3

6	Apartment	£1250.00	3	Y	0.2

4 marks for a & b answered correctly, 1 mark per question if correct records identified but not all the fields provided.

11. Jimmy produces the following algorithm.

```
1: limit = input("Please enter an upper limit")
2: x = 0
3: while x < limit do
4:     print(x)
5:     x = x + 2
6: end while
```

(a) Write down the outputs for the algorithm for an input of 9. **[2]**

0, 2, 4, 6 ,8

1 mark for 0, 2, 4 and 1 mark for 6, 8 (if other answers provided in addition, 1 mark max)

(b) Jimmy intended this algorithm to print the first 9 non-negative even numbers in this case. Explain what this algorithm does instead. **[2]**

- Prints even numbers
- Prints numbers from 0 up to 9

1 mark for each bullet

(c) Suggest how line 4 and line 5 can be changed to make the algorithm work as intended without changing any other part of it. **[3]**

Line 4: print(2x)
Line 5: x=x+1

3 marks for both bullet points, 1 mark for just one

12. Write pseudocode that will perform the following:

Ask a user to enter a number.

If the number is between 0 and 10, output the word **blue**.

If the number is between 10 and 20, output the word **red**.

If the number is between 20 and 30, output the word **green**.

If it is not in the accepted ranges above, output a message to say that this is not a correct colour option. **[6]**

```
Num=input("Enter a number:")
If num>=0 AND num<10 then
    print("blue")
Else if num>=10 AND num<20 then
    print("red")
Else if num>=10 AND num<20 then
    print("green")
Else
    print("not correct")
End if
```

2 marks for correct usage of inputs and outputs
1 mark for conditions to print correct colours (max 3 marks)
1 mark for outputting a message if outside accepted ranges

Turn over for next section

Section C - Implications of Computer Use

13. Artificial Intelligence is increasingly used in everyday life of people, but also at more technical levels to provide advice on medical, financial and other matters.

(a) Provide an example of how AI is used in everyday life, or at a technical level. **[1]**

1 mark for any reasonable response

(b) Discuss the potential ethical and cultural issues associated with the application you have stated. Marks will be awarded for clarity of argument and knowledge of relevant information. **[4]**

1 mark for SPAG
3 marks for relevant responses across ethical and cultural issues; max 2 marks if only one of the two issues are discussed

End of paper

Answer-Key Notes: 16+ Computer Science Mark Scheme (16+ Computer Science Mark Scheme)

Compiled by SATs-Papers.co.uk to help you mark this paper and learn from each answer.

How to use this answer key

This mark scheme shows both the correct answers and the reasoning expected at each mark threshold. Use it to mark objectively: award full marks when the student's answer contains all the bullet points, partial marks when some elements are present, and zero when the response misses the core requirement.

Distinguish between **careless errors** (a forgotten example, a reversed comparison operator) and **knowledge gaps** (unable to explain volatility, no understanding of the fetch-decode-execute cycle). Careless errors respond to practice and exam discipline; knowledge gaps require targeted revision.

The worked examples below unpack questions where markers frequently award zero despite the student being close. Consult them when a response feels 'nearly right' but the mark scheme appears unforgiving, or when you want to understand what separates a one-mark from a two-mark explanation.

Score interpretation

This paper awards **55 marks** across hardware and software (22 marks), programming (28 marks), and implications of computer use (5 marks). A student scoring 44 or above (80 per cent) demonstrates the breadth expected for confident entry to A-level Computer Science, with fluency in fundamental concepts and the ability to trace and correct code.

Scores between 33 and 43 (60–79 per cent) indicate solid foundations but gaps in either hardware principles or programming logic. Check whether errors cluster in one section: if so, targeted revision of that topic is more efficient than retaking immediately.

Below 33 marks suggests the student is not yet secure in GCSE-level content. **Revisit primary versus secondary storage, the CPU cycle, lossless compression, conditional logic, and iteration** before attempting another full paper. Rugby School typically expects entrants to score comfortably above 70 per cent, so a borderline result should prompt careful discussion of readiness.

Worked examples

Hardware and software, Q1–6

Markers reward **precise terminology** (volatile, non-volatile, fetch-decode-execute) and concrete examples (RAM not 'memory', HDD not 'disk'). Partial credit is common: naming an OS role without explaining it earns one mark, but stops there. The BluRay access-speed question (Q3d) often receives zero because students describe capacity instead of *why* higher data density requires faster reads.

> **Q1** : `Primary is volatile (e.g. RAM/Cache); secondary is non-volatile (e.g. HDD/SSD/Optical)`
>
> The mark scheme awards **one mark for explaining the difference** (persistence or volatility) *and* one mark for a suitable example of each type. Writing 'Primary is faster' without mentioning volatility earns zero for that half; similarly, 'RAM and hard drive' without labelling which is which scores only one mark total.

> **Q3(c)** : `Optical → Magnetic → Solid State`
>
> Award **one mark for each correct pairwise comparison**. If a student writes 'Magnetic → Optical → Solid State', they earn one mark (Magnetic faster than Optical, Solid State fastest) but lose the second because Optical is not faster than Magnetic. Both comparisons must be right for full marks.

> **Q3(d)** : `BluRay holds higher-quality media, so more data must be read in the same time frame as a CD-ROM, requiring faster access speed.`
>
> Many students write 'BluRay has more capacity' and stop. That observation alone is **not an explanation of access speed**. The mark scheme wants you to connect higher data density to the need for faster reading during playback. Two marks for linking capacity, media quality, and speed; one mark if only capacity is mentioned; zero if the answer discusses only storage size.

> **Q5(b)** : `Fetch instructions from memory; decode in Control Unit; execute; repeat.`
>
> Four discrete steps, **one mark each**. Students who write 'The CPU fetches, decodes and executes instructions' in a single sentence often lose marks because they omit 'repeat' or fail to mention where decoding happens. List the steps separately to maximise credit.

Programming fundamentals, Q7–9

Boolean logic and tracing (Q7) are marked correct/incorrect with no partial credit: the output is either 'Blue' or it is not. The array-maximum question (Q8) awards **one mark for identifying line 3** and one for the corrected operator; writing 'change the if statement' without specifying the new condition earns only one mark. Sub-routines (Q9) require both description and explanation: 'Reusable code' alone is one mark; 'reduces development time' is the second.

> **Q7(b)** : `Red`
>
> With a = 7 and b = 6, evaluate **NOT(a > b)** = NOT(true) = false; then (b ≤ 7) = true. The compound condition false OR true = true, so the first print executes. Students who answer 'Blue' have misread which branch runs; students who answer 'Green' have ignored short-circuit evaluation or misapplied operator precedence.

> **Q8** : `Line 3: if max < numbers[n] then`
>
> The original code updates max whenever max > numbers[n], which **finds the minimum, not the maximum**. One mark for identifying line 3 as the error; one mark for reversing the operator to <. Writing 'Line 4 should say max = numbers[n]' earns zero because line 4 is correct as written.

> **Q9** : `Reusable code reduces development time; simplifies testing by isolating functionality; easier to maintain because smaller sections are more understandable.`
>
> Award **one mark per advantage** (up to two) and one mark per explanation (up to two). 'Reusable' with no follow-up scores one; 'Reusable, which saves time' scores two. 'Easier to test and maintain' without explaining *why* scores two total (one per advantage), not four.

SQL and algorithms, Q10–12

SQL queries (Q10) reward **correct record IDs**; listing entire rows when the question asks for IDs alone drops you to one mark per query. Algorithm tracing (Q11a) awards marks in bands: one for recognising 0, 2, 4 and one for continuing to 6, 8; extra incorrect values reduce the mark to one total. The pseudocode question (Q12) is marked holistically: two marks for correct input/output structure, up to three for conditionals, one for the else clause.

Q10(a) : `1, 2, 4, 5, 6, 7`

The OR condition means **any property with MonthlyRent ≤ 550 OR Furnished = 'Y'** qualifies. Property 2 has rent £475 (below threshold, unfurnished); property 3 is unfurnished and above £550, so excluded. Students who list only furnished properties have misread OR as AND.

Q11(c) : `Line 4: print(2x); Line 5: x = x + 1`

To print the first nine even numbers (0, 2, 4, ... 16), iterate x from 0 to 8 and **print double the counter**. Three marks for both changes correct; one mark for correcting only one line. Writing 'change the loop to 0 to 18' ignores the constraint that no other part may change.

Q12 : `num = input("Enter a number:"); if num ≥ 0 AND num < 10 then print("blue"); else if num ≥ 10 AND num < 20 then print("red"); else if num ≥ 20 AND num < 30 then print("green"); else print("not correct"); end if`

Two marks for correct input and print syntax. **One mark per colour condition**, maximum three (note the boundaries: 10 belongs to red, 20 to green). One mark for the final else clause. Overlapping ranges (e.g. num ≥ 10 AND num ≤ 20 for red, num ≥ 20 for green) cause 20 to trigger both branches if the logic is careless; use < and ≥ consistently.

Implications of computer use, Q13

The AI question awards **one mark for any reasonable example** (voice assistants, recommendation algorithms, medical diagnostics) and four marks for the discussion. One mark is reserved for spelling, punctuation and grammar. The remaining three require coverage of *both* ethical and cultural issues; discussing only one caps the content marks at two. **Markers look for specific harms or benefits**, not generic statements like 'AI can be biased'.

Q13(a) : `Voice assistants (e.g. Alexa), facial recognition, or AI-powered medical diagnosis.`

One mark for naming any application. 'AI is used in phones' is too vague; 'predictive text' or 'spam filtering' is concrete. **Technical-level examples** (radiology AI, algorithmic trading) are equally acceptable.

> **Q13(b)** : `Ethical: bias in training data leading to discriminatory outcomes; privacy concerns from data collection. Cultural: displacement of jobs traditionally held by specific communities; differing cultural norms around acceptable AI use.`
>
> Three content marks require **both ethical and cultural dimensions**. An answer covering only bias and privacy (both ethical) can earn at most two content marks plus one for SPAG. 'AI might make mistakes' without explaining the ethical or cultural consequence scores low. Link your example from part (a) to concrete issues: if you chose medical AI, discuss misdiagnosis rates in underrepresented groups (ethical) and cultural trust in algorithmic versus human doctors.

Next steps

After marking, **categorise errors by type**: terminology (writing 'memory' for RAM, 'speed' for volatility), logic (reversed operators, misread Boolean conditions), or omission (forgetting to state that steps repeat, or missing the 'else' clause). Terminology errors fix quickly with a glossary and retrieval practice; logic errors need worked examples and tracing on paper. If the student scored below 60 per cent, prioritise one section at a time rather than retaking the full paper immediately.

If the score exceeds 45 marks, **extend understanding** by exploring why the mark scheme accepts alternative phrasings (Q1 allows 'persistence' or 'volatility') and writing model answers for the questions missed. Rugby School's A-level course assumes fluency in these fundamentals, so even high scorers benefit from eliminating small gaps now rather than encountering them under exam pressure later.
