9.1 Computational solution skills

Name: _____ Class: _____ Date: _____ **Total: 20 marks**

KEY WORDS success criterion • normal data • boundary data • invalid data • test plan

Objective

Build the skills behind the **computational solution**: testable success criteria, a test plan, explained code and an evaluation. Practice only; the assessment criteria come from your teacher.

You must be able to:

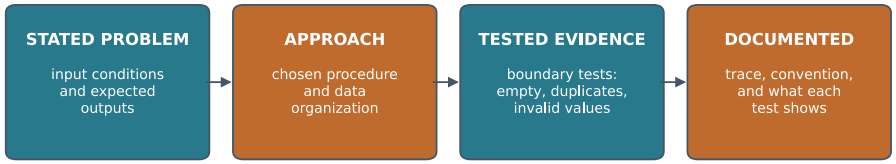
- turn a vague requirement into a testable **success criterion**
- choose **normal data**, **boundary data** and **invalid data** for a test
- write a **test plan** row that points to a criterion
- explain a code extract: what it does in the product and why it was chosen
- judge a criterion as met, partly met or not met, using evidence

1 Key knowledge and worked examples

Study these first. Each one shows the method for a question type used later.

■ Key knowledge: from problem to evidence

The computational solution pipeline



Running is not the requirement: the delivered contract includes the boundary behaviour.

Weak criterion	Strong criterion
It is easy to use.	A result is entered in under 30 s, in at most three clicks.
It stores players.	It stores name, year group and rating, kept after closing.
It is secure.	Only the coach, with a password, can change results.

A strong criterion names **who**, **what**, and **how it is checked**.

■ Worked example: a test plan row

Criterion: "The program accepts a year group from 7 to 13."

Data	Type	Expected result
10	normal	accepted
7 and 13	boundary	accepted
6, 14, "ten"	invalid	rejected with a message

Test both edges, and one value just outside each: most errors hide there.

■ Worked example: evaluating a criterion

Criterion: "No two players meet in two weeks running." Test 4 first failed (one pair repeated); after a fix, no pair repeated over four trial weeks.

1. **Judgement**: met, after one fix.
2. **Evidence**: test 4, before and after, with screenshots.
3. **Improvement**: let the coach see last week's pairs, as asked in feedback.

2 Practice

2.1 Rewrite "The program should be fast" as a testable success criterion. [2]

2.2 A criterion says "The program accepts a quantity from 1 to 20." Give one normal, one boundary and one invalid test value. [3]

2.3 A write-up lists “variables, loops, if statements, functions” as its techniques. Explain why this earns little. [2]

3 Exam-style questions

3.1 For “age 11 to 18 accepted”, which value is boundary data? [1]

- **A** 15
- **B** 18
- **C** 19
- **D** “eleven”

3.2 When should the test plan be written? [1]

- **A** after the program is finished
- **B** before coding, from the success criteria
- **C** only if a test fails
- **D** after the evaluation

3.3 Which is an acceptable way to use code from an AI tool? [1]

- **A** paste it in without comment
- **B** acknowledge the tool, show what was changed, and explain the code
- **C** rename the variables so it looks original
- **D** use it only in the test plan

3.4 A school canteen wants an app where students pre-order lunch by 10:00, choosing one main dish and up to two sides.

(a) Write **two** testable success criteria for the app. [2]

(b) For the rule “up to two sides”, write a test plan with one normal, one boundary and one invalid test, each with its expected result. [3]

(c) The app sorts the day’s orders by collection time before printing them for the kitchen. Explain why this technique was chosen. [2]

(d) In testing, an order placed at 10:00 exactly was rejected, although the requirement says “by 10:00”. Evaluate this result and state what the student should do. [3]

Solutions

Each answer shows the steps, then the **result**.

2.1

- names a measurable time, e.g. “within 2 seconds”
- names the action and size, e.g. “pairings for up to 40 players appear after pressing *Pair*”

2.2

- normal: any value from 2 to 19, e.g. **10**
- boundary: **1** or **20**
- invalid: **0**, **21**, or a non-number such as “five”

2.3

- it names features without saying how or why they are used in this product
- it should show a code extract of a key technique and explain what it does and why it was chosen

3.1 B

- 18 is the top edge of the allowed range

3.2 B

- it fixes what must be shown to work before the product exists

3.3 B

- acknowledging, adapting and explaining is academic integrity

3.4

(a) Any two testable criteria, e.g.:

- an order with one main and up to two sides is saved, and the student sees a confirmation
- orders after 10:00 are refused with a message; the kitchen list prints in under 5 s

(b)

- normal: one side, **accepted**
- boundary: two sides, **accepted**
- invalid: three sides, **rejected with a message**

(c)

- the kitchen prepares meals in the order they are collected
- so sorting by collection time gives a list that can be worked through from the top

(d)

- the criterion is **not met**: “by 10:00” includes 10:00
- the likely cause is a boundary error, `<` used where `<=` was needed
- fix the comparison, re-run the test and record both results as evidence