# Testing

2.3.2 · Algorithms and programming (Paper 2) · revision notes

Testing checks that a program works and meets its requirements. You need to know why we test, the difference between iterative and final (terminal) testing, how to identify syntax and logic errors, and how to choose normal, boundary, invalid and erroneous test data. Expect test-plan tables and questions that use test results to find an error.

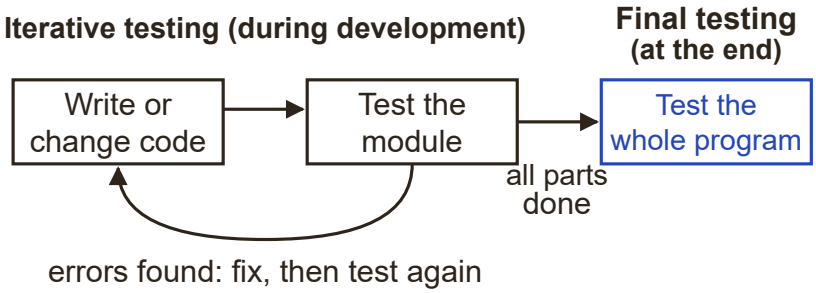
## Key facts

- Purpose: find and fix errors; check the program meets the requirements
- Final (terminal) testing: test the whole program at the end
- Logic error: the program runs but gives unexpected output
- Boundary: at the edge of the valid range (accepted)
- Erroneous: wrong data type (rejected)
- Iterative testing: test each part during development, fix, test again
- Syntax error: breaks the language's grammar rules, so the code cannot be translated or run
- Normal: sensible, valid data (accepted)
- Invalid: correct data type but outside the range (rejected)
- Test plan: test data, type of test, expected result (then actual result)

## Notes

### Why and when to test

- The **purpose of testing** is to find errors so they can be fixed, and to check the program meets its requirements and copes with unexpected input without crashing.
- **Iterative testing**: each part (module) is tested while it is being developed. Errors are fixed and the part is tested again, over and over.
- **Final (terminal) testing**: the whole program is tested at the end of development, to check every part works together and all the requirements are met.

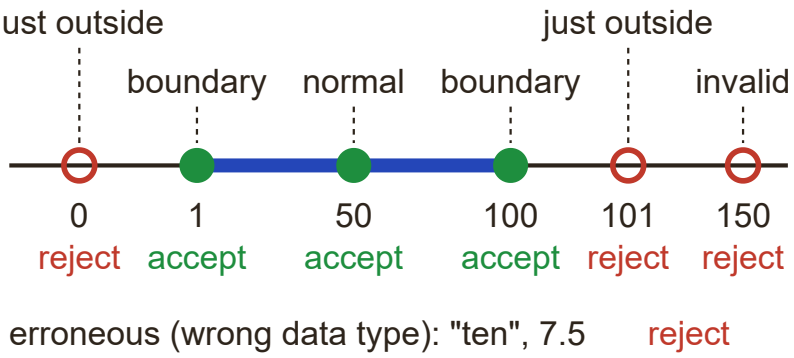



Iterative testing happens over and over during development; final testing happens once at the end

- **Syntax errors** break the grammatical rules of the programming language, so the program cannot be translated or run; the translator reports them. **Logic errors** let the program run but give the wrong result; they are found by testing.

### Types of test data (rule: a whole number from 1 to 100)

- **Normal**: sensible data that should be accepted, such as 50.
- **Boundary**: data at the very edge of the valid range, such as 1 and 100, which should be accepted. Values just outside, such as 0 and 101, are also worth testing: they should be rejected.
- **Invalid**: data of the correct data type that is outside the allowed range, such as 150, which should be rejected.
- **Erroneous**: data of the wrong data type, such as "ten" or 7.5, which should be rejected.




Test data for the rule 'a whole number from 1 to 100' (not to scale)

- Boundary tests matter because a small slip, such as < instead of <=, only shows up at the limit.

## Test plans and refining

- A **test plan** lists each test: what is being tested, the test data, the type of test and the **expected result**. When the test is run, the **actual result** is recorded.
- If the actual result is different from the expected result, there is an error to find and fix.
- **Refining** an algorithm means correcting or improving it after testing, then testing it again.

## How to answer each type of question

### Complete a test plan

3 to 5 marks **Grade 5**

1. Read the rule: which values should be accepted?
2. Pick data for each type asked: normal, boundary, invalid, erroneous.
3. Give the expected result for each: accepted, or rejected with an error message.
4. Every test needs both the data and the correct expected result.

**Example.** A program accepts a number of tickets from 1 to 8. Complete a test plan with one test of each type: normal, boundary, invalid and erroneous. Give the test data and the expected result for each. [4]

**Model answer.**

| Type of test | Test data | Expected result |
|---|---|---|
| Normal | 4 | Accepted (1) |
| Boundary | 8 (or 1) | Accepted (1) |
| Invalid | 9 (or 0) | Rejected, error message shown (1) |
| Erroneous | "two" | Rejected, error message shown (1) |

### Describe iterative and final testing

2 to 4 marks **Grade 5**

1. Iterative: during development, each part tested and retested as it is written.
2. Final: at the end, the whole program tested against the requirements.
3. If asked 'why both', say parts that work alone may not work together.

**Example.** Describe the difference between iterative testing and final testing. [2]

**Model answer.** Iterative testing is carried out during development: each part of the program is tested as it is written, and fixed and retested (1). Final testing is carried out on the whole program at the end of development, to check it meets all the requirements (1).

### Use test results to find an error

3 to 4 marks **Grade 7**

1. Compare each expected result with the actual result.
2. The test where they differ shows the error.
3. Look at the line that makes that decision and rewrite it correctly.

**Example.** A pass needs a mark of 40 or more.

```
01 mark = int(input("Enter mark"))
02 if mark > 40 then
03    print("Pass")
04 else
05    print("Fail")
06 endif
```

Tests: 55 outputs Pass; 40 outputs Fail; 12 outputs Fail.
Identify the test that shows an error, state the type of error and write the corrected line. [3]

---

**Model answer.** The test with 40 (boundary data): it should output Pass but outputs Fail (1). It is a logic error (1). Line 02 should be `if mark >= 40 then` (1).

## Don’t lose marks

- Mixing up invalid (right type, out of range) and erroneous (wrong type).
- Giving test data without an expected result, or with only 'works'.
- Choosing boundary data that is not actually on the limit, such as 2 for a range starting at 1.
- Saying testing proves a program has no errors: it can only show the errors it finds.
- Describing final testing as 'testing the last part of the code'.

## More tips

### Memory tricks

- Rule 1 to 100 test set: 50 (normal), 1 and 100 (boundary), 150 (invalid), "ten" (erroneous).
- Invalid = right type, wrong range.
  Erroneous = wrong type altogether.
- Iterative = as you go; final = at the end.

### Exam technique

- Write expected results as 'accepted' or 'rejected (error message shown)', not 'correct'.
- Use the exact limits from the question for boundary data.
- When finding an error from test results, quote the test data and both the expected and the actual result.

## What each grade needs

What you need to be able to do, from the first marks up to the top grade.

Grade 3

**State the purpose of testing**

To find and fix errors, and check the program meets its requirements.

Grade 4

**Give examples of each type of test data**

Normal, boundary, invalid and erroneous data for a given rule.

Grade 5

**Complete a test plan with expected results**

Give test data, its type and whether it should be accepted or rejected.

Grade 5

**Compare iterative and final testing**

Iterative happens during development; final testing is of the whole program at the end.

Grade 6

**Explain why boundary data is tested**

Errors often happen at the limits, for example < used instead of <=.

Grade 7

**Use test results to find and fix errors**

Compare expected and actual results to locate a logic error and correct the line.

---

Free notes, Full packs, practice papers and mock exams at chhetriacademy.com. Written for this site; not taken from past papers. Stuck? 1-to-1 tutoring with a free 30-minute trial lesson.