Trace tables and common errors

Algorithms and programming (Paper 2) · revision notes

Trace tables let you follow an algorithm step by step, recording the value of each variable, to find its output or a logic error. You also need to identify syntax errors and logic errors in code and correct them. Expect 'complete the trace table' questions and 'identify and correct the error' questions.

Key facts

- Syntax error: breaks the rules of the language, so the code will not translate or run
- Common syntax errors: misspelt keyword, missing bracket, quotation mark or endif
- Trace table: one column per variable plus output; a new row each time a value changes
- Logic error: the program runs but gives the wrong result
- Common logic errors: < instead of <=, wrong loop range, wrong variable, total not reset
- Dry run = working through an algorithm by hand, line by line

Notes

Syntax errors and logic errors

- A **syntax error** breaks the rules (grammar) of the programming language, so the code cannot be translated and run. Examples: a misspelt keyword such as `pritn`, a missing bracket or a missing quotation mark.
- A **logic error** means the program runs, but it does not do what was intended: it gives an unexpected or wrong result. Examples: `<` instead of `<=`, adding the wrong variable, a loop that runs one time too many or too few, or a total that is never reset.
- The translator finds syntax errors for you. Logic errors are found by testing and by tracing the algorithm.

How to complete a trace table

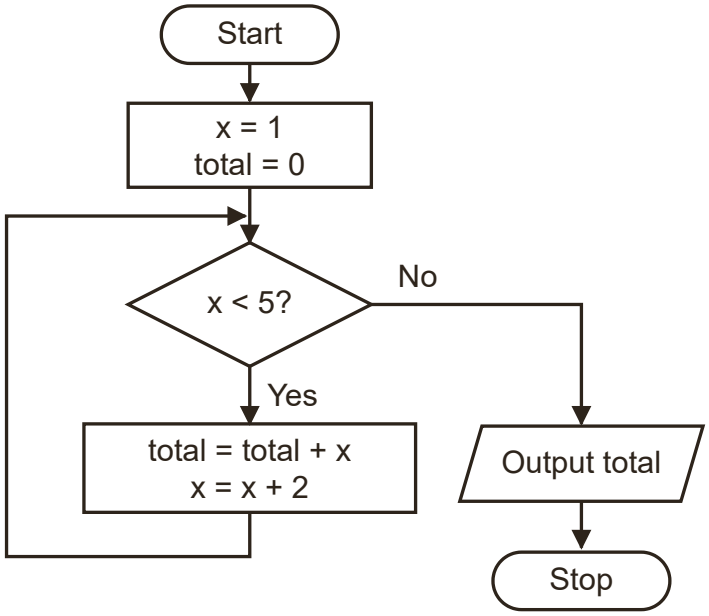
- Give each variable its own column, plus a column for any output.
- Work through the code one line at a time. Each time a variable changes, write its new value in its column on the next row down.
- For a loop, check the condition each time round, and stop as soon as it is false.
- Example:

```
01 x = 1
02 total = 0
03 while x < 5
04    total = total + x
05    x = x + 2
06 endwhile
07 print(total)
```

Trace table (a new row each time round the loop):

x	total	output
1	0	
3	1	
5	4	
		4

- When x is 5 the condition `x < 5` is false, so the loop stops and 4 is output.



The same algorithm as a flowchart: the condition is checked each time round the loop

Using trace tables to find errors

- Work out what the output *should* be, then trace the code. The line where the values start to go wrong is where the logic error is.
- Trace with boundary values too. If a pass needs 50 or more, try exactly 50: code that says `if mark > 50` instead of `if mark >= 50` gives the wrong result only for that value.
- Tracing an algorithm with sample data also shows you its **purpose**, for example that it counts, totals or finds the largest item.

How to answer each type of question

Complete a trace table 3 to 5 marks **Grade 5**

1. Write the starting values on the first row.
2. Go through the code line by line, recording each new value.
3. Check the loop condition every time round and stop when it becomes false.
4. Put the output in the output column at the point it is printed.

Example. Complete a trace table for this algorithm.

```
01 num = 3
02 result = 1
03 for i = 1 to 3
04    result = result * num
05    num = num - 1
06 next i
07 print(result)
```

[5]

Model answer.

i	num	result	output
	3	1	
1	2	3	
2	1	6	
3	0	6	
			6

Starting values num = 3, result = 1 (1); i is 1, 2, 3 (1); num is 2, 1, 0 (1); result is 3, 6, 6 (1); output 6 (1).

Identify and correct errors

2 to 4 marks | **Grade 6**

1. Look for syntax first: brackets, quotation marks, keywords, missing endif / next.
2. Then check the logic against what the program is meant to do: operators, loop ranges, values used.
3. Give the line number and rewrite the whole corrected line.

Example. The program should input five marks and output their average.

```
01 total = 0
02 for i = 1 to 5
03    mark = int(input("Enter mark")
04    total = total + mark
05 next i
06 average = total / 4
07 print(average)
```

Identify the line with a syntax error and the line with a logic error, and write each line correctly. [4]

Model answer.

Syntax error: line 03 (1), `mark = int(input("Enter mark"))` (1).

Logic error: line 06 (1), `average = total / 5` (1).

Describe the difference between syntax and logic errors

2 marks | **Grade 4**

1. Say what a syntax error does: breaks the language rules, so the code will not run.
2. Say what a logic error does: runs, but gives an incorrect result.

Example. Describe the difference between a syntax error and a logic error. [2]

Model answer. A syntax error breaks the rules of the programming language, so the program cannot be translated and run (1). A logic error lets the program run, but it produces an incorrect or unexpected result (1).

State the purpose of an algorithm

1 to 2 marks | **Grade 7**

1. Trace the code with a few simple values.
2. Describe what it does overall, not line by line.
3. Mention what is output.

Example. The array `nums` holds a list of numbers. State the purpose of this algorithm.

```
01 big = nums[0]
02 for i = 1 to nums.length - 1
03    if nums[i] > big then
04       big = nums[i]
05    endif
06 next i
07 print(big)
```

[2]

Model answer. It finds the largest number in the array nums (1) and outputs it (1).

Don’t lose marks

- Running a while loop one extra time: check the condition before each pass.
- Forgetting that a for loop includes its last value, so `for i = 1 to 3` runs three times.
- Updating variables in the wrong order when two lines depend on each other.
- Describing an algorithm line by line when asked for its purpose.
- Calling a wrong result a syntax error: if the program runs, it is a logic error.

More tips

Memory tricks

- Syntax = spelling and grammar (it won't run). Logic = it runs, but the answer is wrong.
- In a trace table, only write a value when it changes: it keeps the table clear.
- Check the last time round a loop carefully: that is where most tracing marks are lost.

Exam technique

- Use the line numbers given in the question when you identify an error.
- When correcting code, rewrite the whole line, not just the changed part.
- If a trace table has more rows than you need, leave the extra rows blank; do not invent extra steps.

What each grade needs

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

Grade 3 **Define syntax error and logic error**

A syntax error breaks the rules of the language; a logic error runs but gives the wrong result.

Grade 4

Spot a syntax error in code

Find missing brackets, missing quotation marks or misspelt keywords, and correct them.

Grade 5

Complete a trace table for a loop

Record each new value of every variable, and the output, in the correct order.

Grade 6

Find and correct a logic error

Use the expected output to find the line that gives the wrong result, then rewrite it.

Grade 7

State the purpose of an algorithm

Trace it with sample data and describe what it does overall, such as 'finds the largest value'.

Grade 8

Trace nested loops and selection accurately

Keep track of inner and outer loop counters and which branch runs each time.

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.