Inputs, processes, outputs and structure diagrams

Algorithms and programming (Paper 2) · revision notes

Before you design an algorithm you identify what goes in (inputs), what the program does (processes) and what comes out (outputs). A structure diagram shows how a problem has been decomposed into smaller parts. Expect input/process/output tables and completing or drawing structure diagrams, often at the start of a Section B scenario.

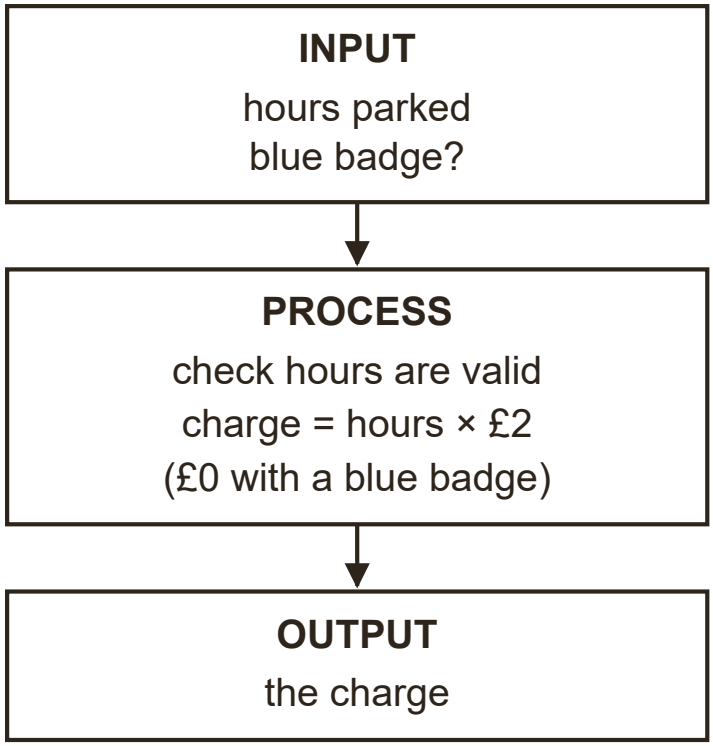
Key facts

- Input = data into the program (keyboard, file, sensor)
- Output = results out of the program (screen, printer, file, speaker)
- Top box = the whole problem; lower levels = smaller sub-problems
- No decisions, loops or arrows in a structure diagram
- Process = what is done with the data (calculate, compare, validate, search, sort)
- Structure diagram = hierarchy of boxes showing a problem broken into sub-problems
- Bottom-level boxes = small tasks, often one sub program each

Notes

Inputs, processes and outputs

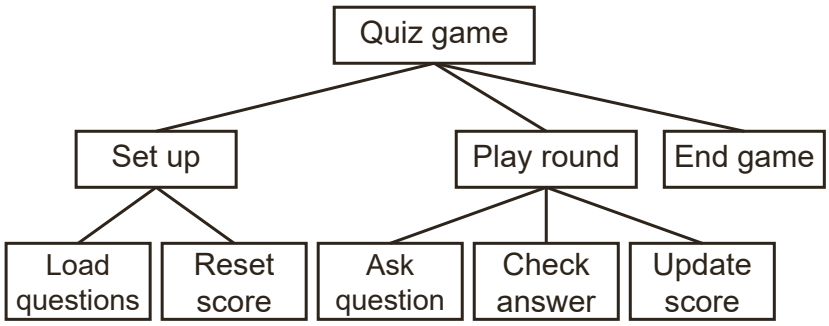
- **Input**: data that goes into the program, for example typed by the user, read from a file or taken from a sensor.
- **Process**: what the program does with the data, for example calculations, comparisons, validation, searching or sorting.
- **Output**: the information the program produces, for example shown on screen, printed, written to a file or played as a sound.
- Quick guide: if the user types it in, it is an input; if the user sees or receives it, it is an output; everything the program works out in between is a process.
- Example: a car park program. Inputs: the number of hours parked, and whether the driver has a blue badge. Processes: check the hours are valid; calculate the charge at £2 per hour, or £0 for a blue badge. Output: the charge.



Input, process and output for the car park program

Structure diagrams

- A structure diagram shows the **decomposition** of a problem (top-down design) as a hierarchy of boxes.
- The box at the top is the whole problem. Each level below breaks a box into smaller sub-problems, joined to it by lines.
- Read each level from left to right. The boxes at the bottom should be small enough to code as one module, such as a sub program.
- A structure diagram has no decision symbols, loops or arrows: it shows how the problem is broken down, not how the program flows from step to step.
- Example: *Quiz game* breaks down into *Set up*, *Play round* and *End game*. *Set up* breaks down into *Load questions* and *Reset score*; *Play round* into *Ask question*, *Check answer* and *Update score*.



Why use them

- They show the structure of the whole program before any code is written.
- Each module can be given to a different programmer, and coded and tested on its own.
- They make it easier to spot tasks that repeat, which can be written once as a sub program and reused.

How to answer each type of question

Complete an input, process, output table

3 marks **Grade 4**

1. Underline the data the user enters: these are the inputs.
2. Underline what is shown or produced: these are the outputs.
3. Everything worked out or checked in between goes in processes.

Example. A program asks for the length and the width of a rectangular room in metres. It works out the floor area and the cost of carpet at £12 per square metre, then displays the cost.
Complete a table showing the input(s), process(es) and output(s) of the program. [3]

Model answer.

Input(s)	Process(es)	Output(s)
The length and the width of the room (1)	area = length × width; cost = area × 12 (1)	The cost of the carpet (1)

Draw or complete a structure diagram

3 to 4 marks | **Grade 6**

1. Put the name of the whole system in one box at the top.
2. Put the main tasks on the second level, left to right.
3. Break any task the question describes into smaller boxes under it.
4. Check that every box belongs under the box above it.

Example. A library app lets members log in, search for a book and reserve a book. Logging in involves entering a username, entering a password and checking the details. Searching can be by title or by author.
Draw a structure diagram for the app. [4]

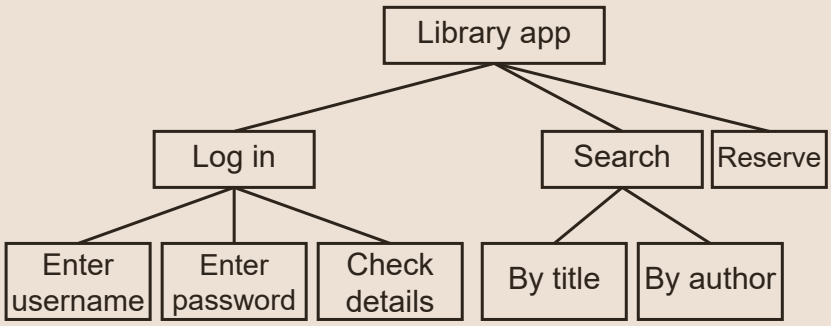
Model answer.

Top box: Library app (1).

Second level: Log in, Search, Reserve (1).

Under Log in: Enter username, Enter password, Check details (1).

Under Search: By title, By author (1).



Explain the benefit of a structure diagram

2 marks | **Grade 7**

1. Say what the diagram does: breaks the problem into separate modules.
2. Say how that helps the programmers, linked to the context.

Example. Explain **one** way a structure diagram helps a team of programmers who are writing a large program. [2]

Model answer. It breaks the problem down into separate modules (1) so each module can be given to a different programmer, and coded and tested separately (1).

Don’t lose marks

- Writing 'the user presses Enter' as an input instead of naming the data that is entered.
- Putting 'display the total' under processes: showing something is an output.
- Drawing a flowchart, with decisions and arrows, when a structure diagram is asked for.
- Putting a sub-task under the wrong parent box, or leaving out the top box.

More tips

Memory tricks

- IPO: In, Process, Out. Typed in = input; seen or received = output; worked out = process.
- A structure diagram is like a family tree for a program: the whole system at the top, its parts below.
- Check a structure diagram by reading any box and asking: is this part of the box above it?

Exam technique

- Use the words from the scenario for inputs and outputs, such as 'the number of tickets'.
- Keep each box short: a verb and a noun, such as 'Check password'.
- Section B often starts with an input, process, output question on the scenario: it is a quick way to pick up marks.

What each grade needs

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

Grade 3 — **Identify the inputs and outputs of a problem**
Pick out the data that goes in and the results that come out of a described program.

Grade 4 — **Identify the processes in a problem**
Name what the program does with the data, such as calculating, checking or sorting.

Grade 4 — **Describe what a structure diagram shows**
It shows a problem broken down into smaller sub-problems as a hierarchy of boxes.

Grade 5 — **Complete a partly drawn structure diagram**
Add the missing boxes at the correct level, under the correct parent box.

Grade 6 — **Draw a structure diagram for a scenario**
Put the whole system at the top and break it down over two or more levels.

Grade 7 — **Explain how structure diagrams help development**
Each module can be coded and tested separately, shared between programmers and reused.

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.