

THEME B: COMPUTATIONAL THINKING · B1.1

Approaches to computational thinking — Mark scheme

33 marks · covers B1.1.1–B1.1.4

How to mark. Each creditworthy point is on its own line and ends with a semicolon (;). A line beginning with ... is an *expansion* of the line above it, so the pair is worth the marks shown. A slash (/) separates acceptable alternative wordings. Words in (brackets) are not needed for the mark. Mark positively: credit what the student has got right rather than penalising what is missing, and accept any valid alternative not listed here. Do not award more than the maximum for a part. Where a later part uses a wrong answer from an earlier part correctly, award follow-through marks (FT). HL marks parts that assess higher level only content.

Q	ANSWER	MARKS
1(a) B1.1.1	Award [2] max Award [1] per item, [2] max. Any two. A problem statement; Objectives and goals; Constraints and limitations; Input specifications; Output specifications; Evaluation criteria;	2
1(b) B1.1.1	Award [4] max Award up to [2] for the input specification and up to [2] for the output specification. Inputs For each student: a student identifier and their three club choices, given in order of preference; For each club: its name and the maximum number of places available; Valid ranges and formats, for example that the three choices must be different and must name clubs that exist; Outputs For each student, the single club they have been allocated to; A list per club of the students allocated to it, not exceeding its capacity; A list of any students who could not be allocated to any of their three choices; Award [1] max on each side for a vague answer such as 'the student's choices' with no detail of format or constraints.	4
2(a) B1.1.2	Award [2] max Award [1] per point up to [2] max. Decomposition means breaking a large, complex problem down into smaller sub-problems; Each sub-problem is easier to understand, to solve and to test on its own; The sub-problems can be worked on separately, and by different people, then combined;	2
2(b) B1.1.2	Award [2] max Award [1] per point up to [2] max. Abstraction means removing detail that is not relevant to the problem being solved;	2

Q	ANSWER	MARKS																								
	Only the features that matter to the solution are kept, which makes the problem simpler to reason about; ...(so) a general solution can be produced rather than one tied to unnecessary specifics;																									
2(c) B1.1.2	Award [2] max Award [1] per point up to [2] max. Must be applied to this system. Pattern recognition means spotting similarities between problems or within the data; The developers would notice that allocating students to clubs is the same kind of problem as allocating them to exam seats or timetable options; ...(so) an approach that already works for one can be reused rather than invented again; They may also spot patterns in the choices themselves, such as a few clubs being heavily oversubscribed every year;	2																								
3 B1.1.3	Award [5] max Award up to [3] for decomposition and up to [3] for abstraction, [5] max. Both must be applied to this system. Decomposition Split the system into separate parts: collecting the choices, validating them, running the allocation, and reporting the results; The allocation itself decomposes further into handling first choices, then second, then third, and finally the students left over; Each part can then be written and tested independently, and a fault can be traced to one part; Abstraction Represent each student only by the data the allocation needs: an identifier and three ranked choices; Details such as the student's address, year group or photograph are irrelevant here and are left out; Represent each club only by a name and a capacity; ...(so) the problem reduces to matching a list of ranked preferences against a list of limited-capacity slots, which is a general problem with known solutions;	5																								
4(a) B1.1.4	Award [1] max A diamond (rhombus); Accept a correct sketch of the symbol.	1																								
4(b) B1.1.4	Award [4] max Award [1] for each fully correct row after the seeded row. Award FT for rows that follow correctly from an earlier error. <table><tr><th>N</th><th>TOTAL</th><th>COUNT</th><th>OUTPUT</th></tr><tr><td></td><td>0</td><td>0</td><td></td></tr><tr><td>8</td><td>8</td><td>1</td><td></td></tr><tr><td>5</td><td>13</td><td>2</td><td></td></tr><tr><td>2</td><td>15</td><td>3</td><td></td></tr><tr><td>-1</td><td>15</td><td>3</td><td>5</td></tr></table> The output column stays empty until the loop ends. Accept a blank rather than a repeated value in the total and count columns of the final row, provided the values shown are correct.	N	TOTAL	COUNT	OUTPUT		0	0		8	8	1		5	13	2		2	15	3		-1	15	3	5	4
N	TOTAL	COUNT	OUTPUT																							
	0	0																								
8	8	1																								
5	13	2																								
2	15	3																								
-1	15	3	5																							
4(c)																										

Q	ANSWER	MARKS
B1.1.4	Award [1] max 5; total / count = 15 / 3 = 5. Award FT from the candidate's own trace table.	1
4(d) B1.1.4	Award [2] max Award [1] for the output and [1] for the reason. It outputs "No data"; The loop ends immediately, so count is still 0; the second decision (count > 0) is false, which also avoids dividing by zero;	2
5(a) B1.1.4	Award [2] max Award [1] per point up to [2] max. A connector joins one part of a flowchart to another without drawing a long flowline between them; It is used when the flowchart continues on another page, or to avoid lines crossing; Matching connectors carry the same label so the reader knows where the flow resumes;	2
5(b) B1.1.4	Award [2] max Award [1] for the point and [1] for the expansion. Any one reason. The logic can be checked and corrected before any code is written; ...(so) an error in the design is found when it is cheap to fix, rather than after implementation; A flowchart is language-independent; ...(so) the same design can be implemented in Java or Python, and can be discussed with people who do not program; It makes the flow of control explicit; ...(so) every branch and loop can be checked for a path that has been forgotten, such as an empty input;	2
6 B1.1.3	Award [4] max Award [1] per point up to [4] max. A developed real-world example is required for full marks. Computational thinking is a set of techniques for approaching problems: decomposition, abstraction, pattern recognition and algorithmic design; These describe how to think about a problem, not how to write it in a programming language; The output may be a plan, a process or a decision rather than a program; Example (credit any well-developed one) Planning a school timetable: the problem is decomposed into staffing, rooming and student options; irrelevant detail is abstracted away; recurring clashes are recognised as a pattern; and a repeatable procedure is designed to resolve them; ...(and) the result can be carried out by people with a whiteboard, with no code written at all; Diagnosing a fault in a car engine, or planning a supply route, follows the same four techniques; Award [2] max if the answer only lists the four concepts without an example, or gives an example that is really a programming task.	4

Total for this paper: 33 marks