7 Algorithm design and problem-solving – Systems Design
QUESTIONS

7 – Algorithm design and problem-solving

7 – Algorithm design and problem-solving				
1	Understand the program development life cycle, limited to: analysis, design, coding and testing			
2	(a) Understand that every computer system is made up of sub-systems, which are made up of further sub-systems			
2	(b) Understand how a problem can be decomposed into its component parts			
3	Explain the purpose of a given algorithm			
4	Understand standard methods of solution			
5	(a) Understand the need for validation checks to be made on input data and the different types of validation check			
5	(b) Understand the need for verification checks to be made on input data and the different types of verification check			
6	Suggest and apply suitable test data			
7	Complete a trace table to document a dry-run of an algorithm			
8	Identify errors in given algorithms and suggest ways of correcting these errors			
9	Write and amend algorithms for given problems or scenarios, using: pseudocode, program code and flowcharts			

More Guidance:

Candidates should be able to:	Notes and guidance
1 Understand the program development life cycle, limited to: analysis, design, coding and testing	• Including identifying each stage and performing these tasks for each stage: – analysis: abstraction, decomposition of the problem, identification of the problem and requirements – design: decomposition, structure diagrams, flowcharts, pseudocode – coding: writing program code and iterative testing – testing: testing program code with the use of test data
2 (a) Understand that every computer system is made up of sub-systems, which are made up of further sub-systems	
(b) Understand how a problem can be decomposed into its component parts	• Including: – inputs – processes – outputs – storage
(c) Use different methods to design and construct a solution to a problem	• Including: – structure diagrams – flowcharts – pseudocode
3 Explain the purpose of a given algorithm	• Including: – stating the purpose of an algorithm – describing the processes involved in an algorithm

7 Algorithm design and problem-solving continued

Candidates should be able to:	Notes and guidance
4 Understand standard methods of solution	• Limited to: – linear search – bubble sort – totalling – counting – finding maximum, minimum and average values
5 (a) Understand the need for validation checks to be made on input data and the different types of validation check	• Including: – range check – length check – type check – presence check – format check – check digit – the purpose of each validation check and writing algorithms to implement each validation check
(b) Understand the need for verification checks to be made on input data and the different types of verification check	• Including: – visual check – double entry check
6 Suggest and apply suitable test data	• Limited to: – normal – abnormal – extreme – boundary • Extreme data is the largest/smallest acceptable value • Boundary data is the largest/smallest acceptable value and the corresponding smallest/largest rejected value
7 Complete a trace table to document a dry-run of an algorithm	• Including, at each step in an algorithm: – variables – outputs – user prompts
8 Identify errors in given algorithms and suggest ways of correcting these errors	
9 Write and amend algorithms for given problems or scenarios, using: pseudocode, program code and flowcharts	• Precision is required when writing algorithms, e.g. `x > y` is acceptable but `x is greater than y` is not acceptable • See section 4 for flowchart symbols • See section 4 for pseudocode

5 The first stage of the program development life cycle is analysis. Two of the tasks in analysis are abstraction and decomposition.

(a) Describe what is meant by abstraction.

.....

.....

.....

..... [2]

(b) Identify **three** of the component parts when a problem has been decomposed at the analysis stage.

1 .....

2 .....

3 .....

[3]

(c) Identify and describe **one** other stage of the program development life cycle.

.....

.....

.....

..... [2]

7 Algorithm design and problem-solving – Systems Design
QUESTIONS

1 **(a)** **Four** descriptions of stages in the program development life cycle are shown.

Draw **one** line to link each description to its most appropriate program development life cycle stage.

Not all program development life cycle stages will be used.

Program development life cycle description	Program development life cycle stage
develop an algorithm to solve the problem by using structure diagrams, flowcharts or pseudocode	analysis
detect and fix the errors in the program	coding
identify the problem and its requirements	design
write and implement the instructions to solve the problem	evaluation
	testing

[4]

(b) Identify **three** of the component parts after a problem has been decomposed.

1 .....

.....

2 .....

.....

3 .....

.....

[3]

1 Tick (✓) **one** box to identify the first stage of the program development life cycle.

A Analysis ☐

B Coding ☐

C Design ☐

D Testing ☐

[1]

7 Algorithm design and problem-solving – Systems Design
QUESTIONS

3 Identify **three** different ways that the design of a solution to a problem can be presented.

1 .....

.....

2 .....

.....

3 .....

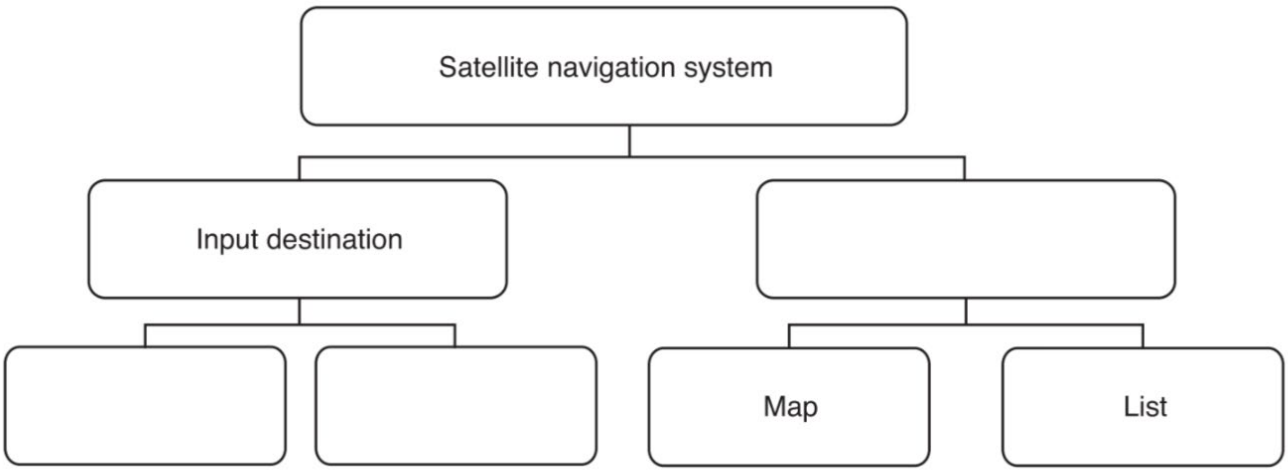
.....

[3]

3 A satellite navigation system works using destination details entered by the user, either a new destination or chosen from previously saved destinations. The satellite navigation system will then output directions to the destination in the form of either a visual map or a list of directions.

A satellite navigation system is an example of a computer system that is made up of sub-systems. This structure diagram shows some of its sub-systems.

Complete the diagram by filling in the empty boxes.

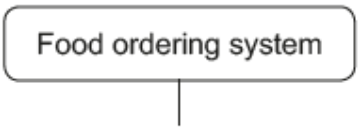


[2]

5 A food ordering system is an example of a computer system that is made up of sub-systems.

The food ordering system:
- allows the user to enter the details of the food they want to order and to pay for the order
- displays food available as pictures or as a list.

Complete the structure diagram for the given parts of the food ordering system.



[4]