Topic 7: Algorithm Design and Problem-Solving

EAL keyword sheet - Cambridge IGCSE Computer Science 0984. Write the word in your own language in the last column.

Keyword	Meaning	Example sentence	In my language
algorithm	A set of clear steps to solve a problem.	*An algorithm tells the computer what to do.*	
computational thinking	Ways of thinking that help you solve problems like a computer scientist.	*Decomposition is part of computational thinking.*	
decomposition	Breaking a big problem into smaller, easier parts.	*I used decomposition to plan the game.*	
abstraction	Keeping only the important details and ignoring the rest.	*A map uses abstraction.*	
pattern recognition	Finding things that are the same or similar in different problems.	*Pattern recognition saves time.*	
analysis	Studying a problem to find out exactly what is needed.	*Analysis is the first stage of development.*	
design	Planning how the solution will work before writing code.	*I made a flowchart in the design stage.*	
coding	Writing the program using a programming language.	*Coding starts after the design.*	
testing	Running a program to check that it works correctly.	*Testing found two errors.*	
flowchart	A diagram that shows the steps of an algorithm using shapes and arrows.	*A diamond shape in a flowchart shows a decision.*	
pseudocode	A way to write an algorithm in simple, code-like English.	*Pseudocode is not a real programming language.*	
structure diagram	A diagram that shows how a problem is split into smaller parts.	*The structure diagram has three levels.*	
trace table	A table used to follow the values of variables as an algorithm runs.	*I completed the trace table.*	
dry run	Testing an algorithm by hand, without a computer.	*A dry run helps find logic errors.*	
linear search	Checking each item in a list, one by one, until you find the one you want.	*A linear search starts at the first item.*	

Keyword	Meaning	Example sentence	In my language
bubble sort	Sorting by comparing neighbouring items and swapping them if they are in the wrong order.	*A bubble sort repeats until no swaps are needed.*	
swap	To change the places of two items.	*Swap the two numbers.*	
total	To add up values.	*Total the marks of all the students.*	
count	To find how many times something happens.	*Count how many students passed.*	
maximum / minimum	The largest / smallest value.	*Find the maximum score in the list.*	
average	The total divided by the number of values.	*The average of 4 and 6 is 5.*	
validation	A check that data is sensible and allowed. It does not check that data is correct.	*Validation can stop a person entering 'abc' as an age.*	
range check	Checks that a number is between a lowest and highest value.	*A range check allows ages from 0 to 120.*	
length check	Checks the number of characters is allowed.	*A length check makes sure a password has 8 or more characters.*	
type check	Checks that data is the correct data type, e.g. a number.	*A type check rejects text in a number field.*	
presence check	Checks that something has been entered.	*A presence check stops empty boxes.*	
format check	Checks that data is in the right pattern, e.g. a date.	*A format check makes sure the date is DD/MM/YYYY.*	
verification	Checking that data was copied or entered correctly.	*Double entry is a type of verification.*	
double entry	Typing the data twice and comparing both entries.	*Double entry is used when you create a password.*	
normal data	Data that the program should accept.	*Normal data for an age is 15.*	
abnormal data	Data that the program should reject.	*Abnormal data for an age is 'ten'.*	
extreme data	The largest or smallest values that are still allowed.	*Extreme data for ages 0-120 is 0 and 120.*	
boundary data	Data at the edge of the accepted range, and the values just outside it.	*Boundary data includes 120 and 121.*	