

# 1 Algorithms

## Learning outcomes

By the end of this chapter you should be able to:

- explain what an algorithm is and create algorithms to solve specific problems
- use sequence, selection and iteration in algorithms
- use input, processing and output in algorithms
- express algorithms using flowcharts and pseudo-code
- analyse, assess and compare different algorithms
- create, name and use suitable variables
- use arithmetic, relational and Boolean operators
- use conditional statements.



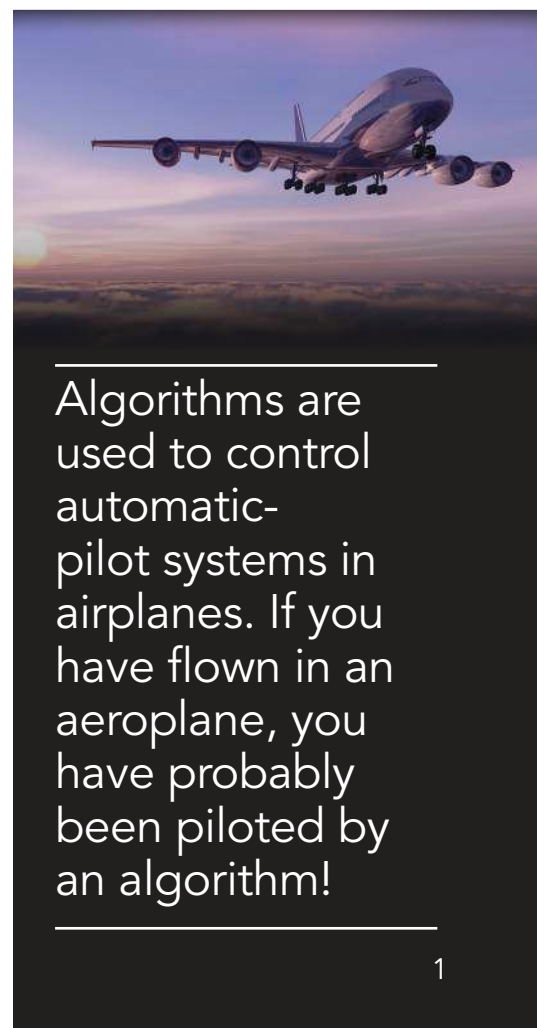
**Challenge:** create an algorithm to help a taxi company calculate fares

- By the end of this chapter, you should have a thorough knowledge of how algorithms can be used to solve complex problems and how they can be displayed using flowcharts and pseudo-code.
- An algorithm is a step-by-step procedure for solving a problem. It can be followed by humans and computers.
- Your challenge is to create an algorithm to help a taxi company calculate fares.

## Why algorithms?

Algorithms run our world! In every area algorithms are used to decide what action should be taken in a particular circumstance. As computers can consider all the possibilities far more quickly than a human brain, they are becoming more important to the running of the world. Here are just a few examples:

- In a game of chess, when each player has made three moves, there are over 9 million possible moves available; after four moves there are over 288 billion. Computers have the ability to consider all these possible moves far more quickly than humans. That is why no chess grandmaster has beaten a top computer chess algorithm since 2005.
- Algorithms are used by financial organisations to trade shares on the stock market. A computer following an algorithm can decide which deal to make far more quickly than a human and a split second difference can be worth millions of pounds.
- Closely guarded algorithms are used for internet searches to make them quicker and the results more relevant to the user. They will even auto-complete the search terms based on previous searches.



Algorithms are used to control automatic-pilot systems in airplanes. If you have flown in an aeroplane, you have probably been piloted by an algorithm!

## GCSE Computer Science for AQA

## What is an algorithm?

An algorithm is a step-by-step procedure for solving problems. It is a set of instructions that can be followed by humans and computers.

We use algorithms to carry out everyday tasks often without thinking about them. For example, an algorithm to solve the problem of getting ready for school might be:



### Watch out

In an algorithm, the order in which the tasks are carried out is very important to its success or failure. For example, this algorithm would not be very successful if 'shower' was placed after 'get dressed'. The sequence is very important.



### Key terms

**sequence:** the order in which tasks are carried out

**sub-tasks:** small steps making up a larger task

Get out of bed.
Shower.
Get dressed.
Turn on kettle.
Put bread in toaster and turn on.
Wait for kettle to boil and make tea.
Wait for bread to toast, butter it and add marmalade.
Drink tea and eat toast.
Gather school books and put in bag.
Put on shoes and coat.
Leave the house.

The algorithm shows the **sequence** of tasks. Different people will design different algorithms, as they will do things in a different order, meaning there can be many solutions to the same problem. Some of these tasks could be further divided into **sub-tasks** as they might be made up of smaller steps.

For example, 'showering' could involve many different steps including turning on the shower and setting the correct temperature. If all the possible sub-tasks were included, the complete algorithm would get very large and complicated.

### ACTIVITY 1.1

Create an algorithm for someone who has never made a cup of tea before to follow, in order to make a cup of tea successfully.

Compare it with other members of your group. There will probably be differences, for example some may include the addition of sugar.



Download Worksheet 1.1 from Cambridge Elevate

This example seems pretty easy. However, like a typical algorithm, it is simply a list of steps. Here is part of another algorithm which is the start of a recipe to make a perfect meringue.



1. Tip 4 large egg whites into a large clean mixing bowl (not plastic).
2. Beat them on medium speed with an electric hand whisk.
3. Keep beating until the mixture resembles a fluffy cloud and stands up in stiff peaks when the blades are lifted.
4. Now turn the speed up and start to add 115 g caster sugar, a dessertspoonful at a time until there is none left.
5. Continue beating for 3–4 seconds between each addition.

SEQUENCE

SEQUENCE

ITERATION

SELECTION

ITERATION

SELECTION

SEQUENCE

In addition to sequence, this algorithm has two new elements: **iteration** and **selection**.

Iteration means doing things over and over again. The cooks have to beat the mixture and then stop and ask themselves if it resembles a fluffy cloud. If it does not, they have to beat again, check again, beat again and check again until they are convinced they have made a fluffy cloud. There is also repetition when adding the sugar. It has to be added a spoonful at a time until there is none left.

Selection means making decisions. As well as doing things over and over again, the cooks have to make a decision. Does it resemble a fluffy cloud?



**Key terms**

**iteration:** a procedure or a set of statements or commands is repeated either for a set number of times or until there is a desired outcome

**selection:** a question is asked, and depending on the answer, the algorithm takes one of two courses of action



**Complete Interactive Activity 1a on Cambridge Elevate**

**ACTIVITY 1.2**

Using sequence, selection and iteration write an algorithm that a person (who has never done this before) could follow in order to successfully prepare a bath with the water at the correct temperature. Annotate the algorithm to indicate sequence, selection and iteration.

You could set it out as shown in the table below where the first four tasks have been done for you.

Put plug in the bath.	Sequence
Turn on hot tap.	Sequence
Is the water at the correct temperature?	Selection
Turn cold tap until water is at the correct temperature.	Iteration

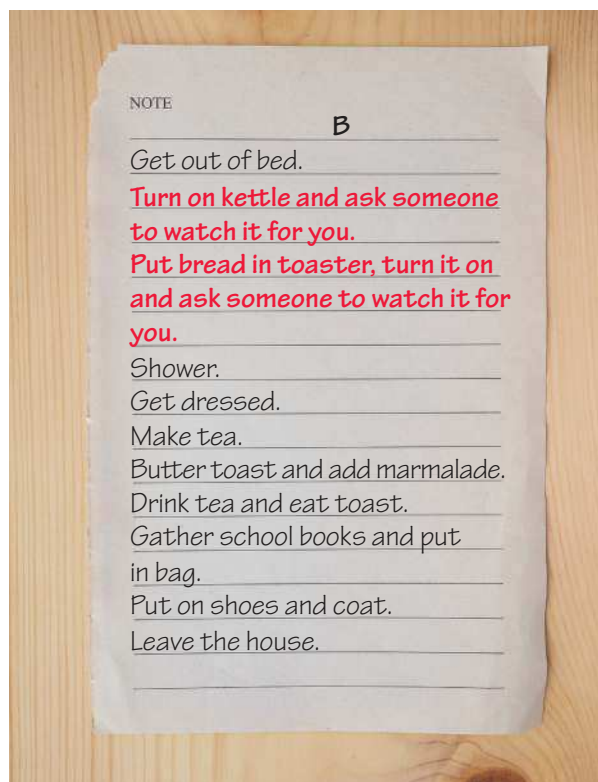
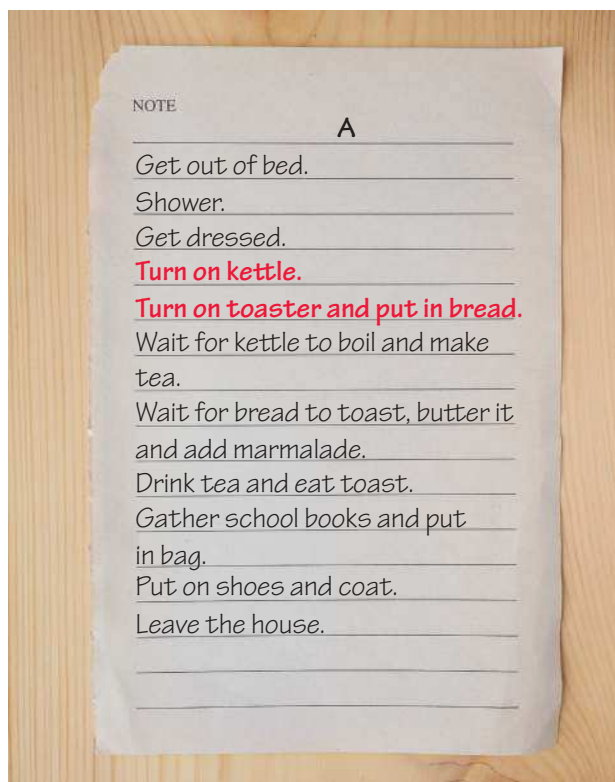
## GCSE Computer Science for AQA

### What makes a successful algorithm?

The two most important criteria are:

- **Correctness:** it successfully solves the problem
- **Efficiency:** it solves the problem in the least possible time.

List A is the 'getting up' algorithm we looked at earlier. List B is similar but with the sequence slightly altered.



#### Watch out

The algorithm is now more efficient and also safer; it would not have been safe to leave the kettle and toaster unattended. Computer scientists must consider health and safety issues when designing real-life solutions.

List B is more efficient as it could be implemented in less time. The kettle and the toaster are turned on before taking a shower and so the water will boil while the person is showering. There will be no waiting time.

### An algorithm for a computer

Now let's look at a simple algorithm that we could create for a computer to follow, instead of a human. Computers are ideal for obeying orders and carrying out actions over and over again. In fact, that is their main function.

It is important for the temperature in a shopping mall to be kept at a set value. It will keep the shoppers comfortable and it will help to prevent condensation on glass shop windows and slippery floor surfaces. Here is an algorithm intended to be used to control the temperature in a shopping mall and maintain a temperature of 20°C.

1. Check the temperature.
2. If the temperature is greater than 20°C, then turn off the heaters and open the ventilators.



3. If the temperature is less than 20°C, then turn on the heaters and close the ventilators.
4. Go back to instruction 1.

This is a simple algorithm but it includes the basic building blocks:

- *Sequence*: there is a list of instructions in the correct order.
- *Selection*: the 'if' statements in instructions 2 and 3 allow a decision to be made and an action to be taken.
- *Iteration*: instruction 4 tells the computer to go back to instruction 1 and so the sequence will run over and over again indefinitely.

### **i** Remember

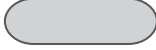
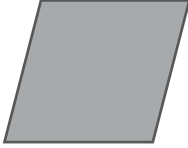
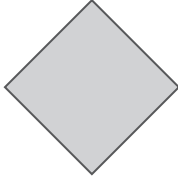

1. An algorithm is a step-by-step procedure for solving a problem in a finite number of steps.
2. The basic building blocks of algorithms are sequence, selection and iteration.
3. The criteria for a successful algorithm are correctness and efficiency.
4. An algorithm must be translated into a programming language before it can be executed by a computer.


## Flowcharts

Flowcharts can be used to represent algorithms visually.

They use symbols connected by arrows to show the flow of the algorithm.

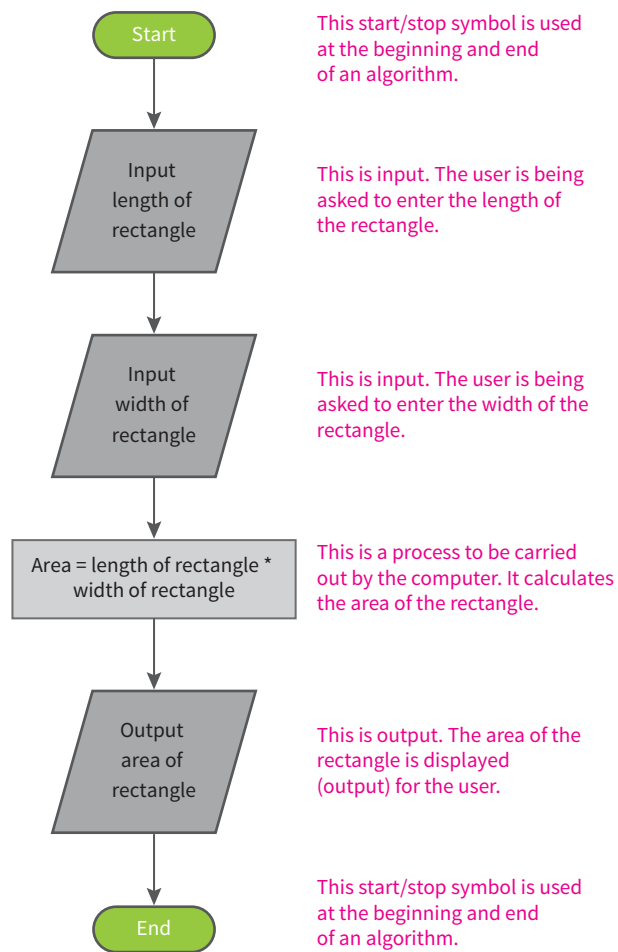
The symbols used are:

This represents the start or end point of the flowchart. You always start and finish your flowchart with this symbol.	
You use this to represent data input or data output. For example it could be a number or name entered by a user.	
You use this where a decision has to be made. It is also called selection. It will contain a question, for example: 'Is the temperature greater than 20°C?' If it is, then an arrow will point to a task to be carried out and if it is not, then an arrow will point to a different action.	
You use this to represent a process that must be carried out by the algorithm. In this example it is used as a result of one of the questions that has been asked, e.g. 'Turn on the heaters' or 'Turn off the heaters'.	

 Watch the flowcharts animation on Cambridge Elevate


**WORKED EXAMPLE**

Here is the flowchart of an algorithm to calculate the area of a rectangle:



There are two inputs of the dimensions of the rectangle, a process to calculate the area and an output of the area.

In this example, there is just sequence: a list of tasks to be performed.

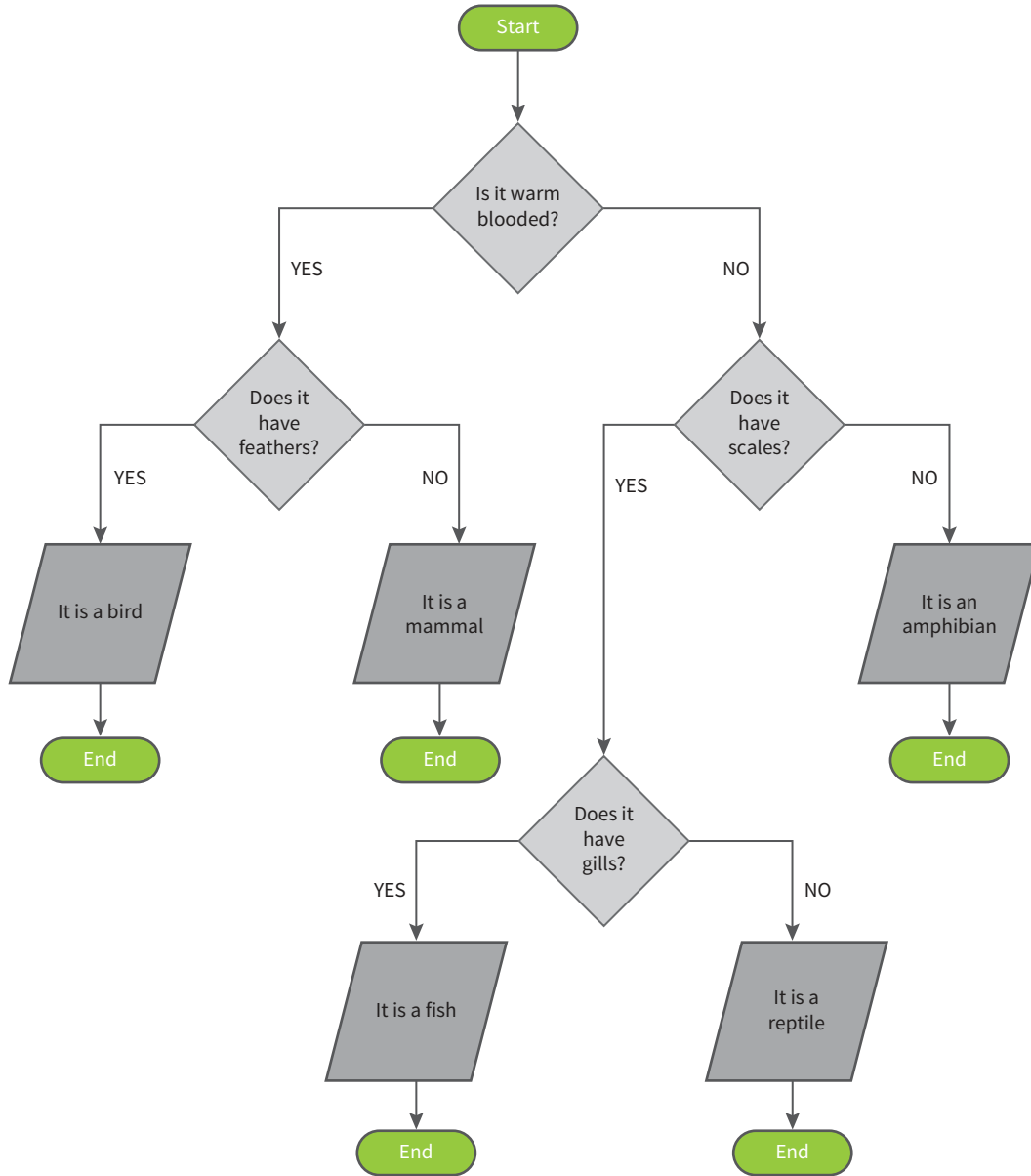
 Complete Interactive Activity 1b on Cambridge Elevate

**ACTIVITY 1.3**

At the end of each day, an ice cream seller calculates how much money he has collected. Assuming that the ice creams all cost the same amount, draw a flowchart of an algorithm that would output the total amount collected during the day.

**WORKED EXAMPLE**

Here is a flowchart of an algorithm to identify a vertebrate animal. It includes sequence and selection.



 **Complete Interactive Activity 1c on Cambridge Elevate**

**ACTIVITY 1.4**

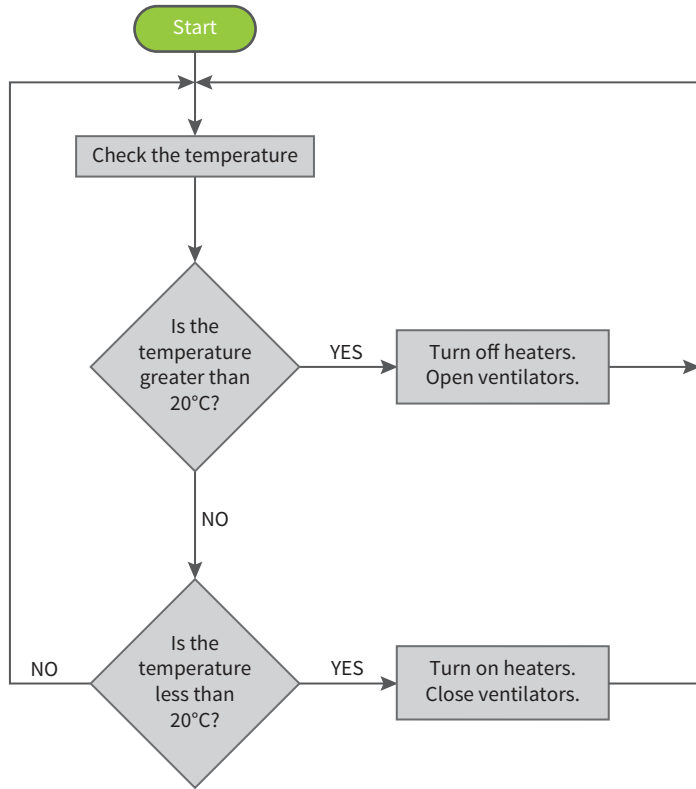
A teacher is marking his students' test papers on a computer. If they achieve over 50 per cent, he would like the message 'Well done!' displayed. If they achieve over 90 per cent, they should also receive a second message stating 'This is an excellent result.' If they score 50 per cent or lower, the message will be 'You must try harder next time.'

Draw a flowchart of an algorithm that would output these messages.

GCSE Computer Science for AQA

WORKED EXAMPLE

A flowchart to represent the algorithm to control the temperature of the shopping mall that we mentioned earlier would look like this. It contains sequence, selection and iteration.



There are three **processes**: one to check the temperature in the mall and two to either turn off the heaters and open the ventilators, or turn them on and close the ventilators.

There are two **decisions**: is the temperature greater or is it less than 20°C? Both are needed as the temperature could in fact be equal to 20°C.

There are only two possible answers for each decision question: *yes* or *no*, and the arrows show the relevant action to be taken depending on the answer.

*Iteration* is shown in the flow diagram as the arrows always lead the flow back to the first process. So the algorithm will repeat over and over again indefinitely.

As the algorithm repeats forever, no end symbol is required.



Tip

Look back at your answers for Activity 1.1.

ACTIVITY 1.5

Draw a flowchart to illustrate an algorithm for making a cup of tea. It should include sequence, selection and iteration.

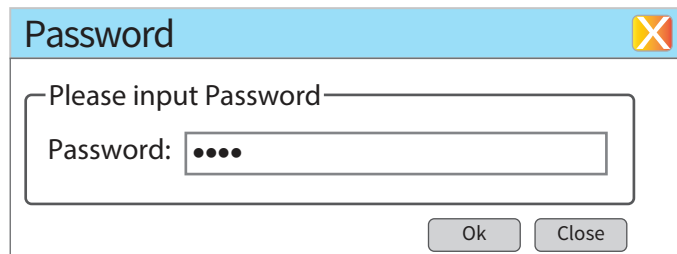
Input and output

In some of the previous examples, user input was required and information was output to the user.



Key terms

**decision**: when a question is asked (as in selection) the answer will lead to one or more varied alternative actions  
**process**: an activity that a computer program is performing




A common request for user input is to enter a password.



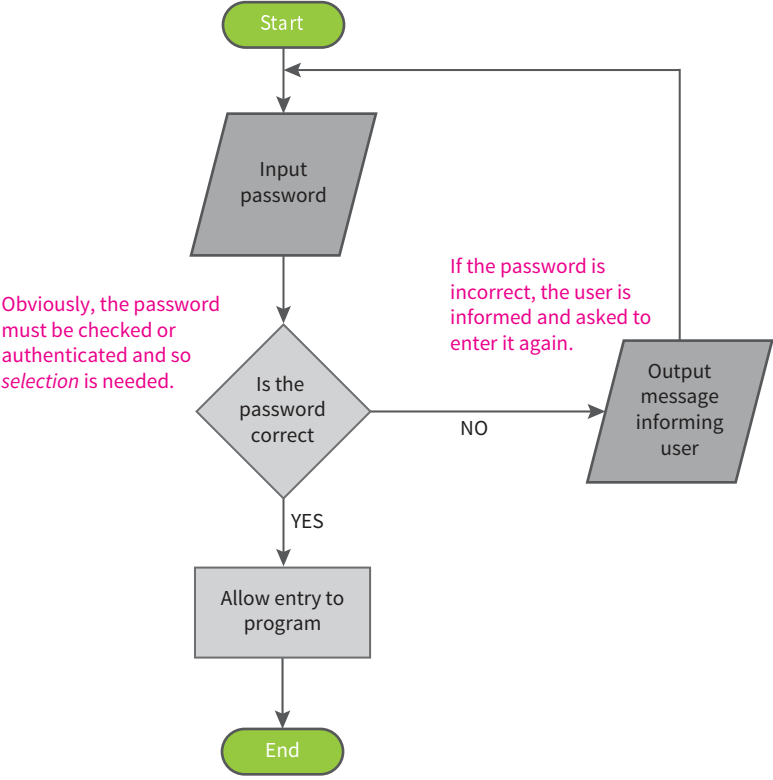
**WORKED EXAMPLE**

Here is a flowchart of an algorithm to **authenticate** a password.


When a user enters a password, it has to be confirmed that it is true; it has to be authenticated.

 **Key term**


**authenticate:** confirm that a user's password has been entered correctly



In this flowchart there is iteration. If the password is incorrect, then in this particular algorithm, the user is asked to enter it again, and again, and again, forever or until it is correct.

 **Download Worksheet 1.2 from Cambridge Elevate**

Now assume that the user is given three attempts and then the account is locked.

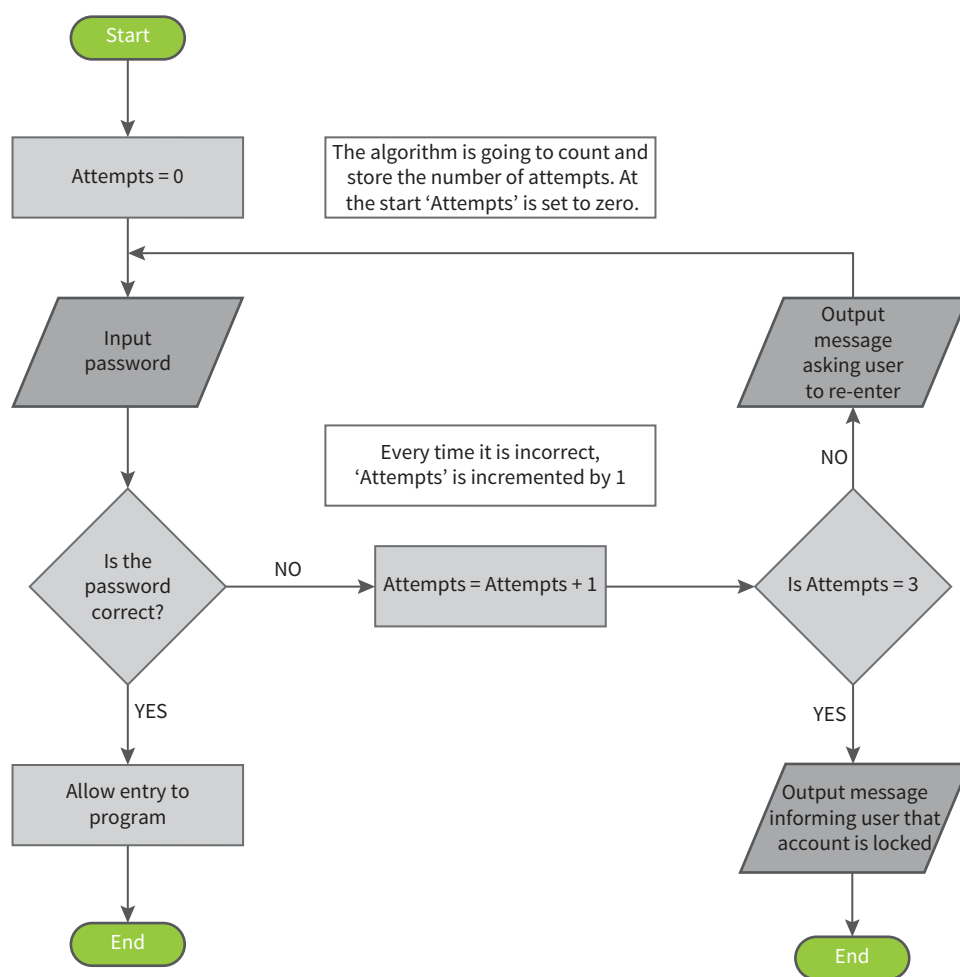
**Information: HandiTax 2015** 

Too many failed login attempts, account LOCKED

**WORKED EXAMPLE**

Here is a flowchart to authenticate a password and lock the account after three incorrect attempts.

The algorithm will have to keep a count of the number of attempts that have been made.



In this algorithm, we have used a container called 'attempts' to keep a count of the number of attempts that have been made.



**Key term**

**variable:** a container which is used to store values such as an 'attempts' counter

When incorrect attempts are made, the value of 'attempts' changes. It does not keep the same value throughout the algorithm, but it can change because it is a **variable**.

On the first attempt, it is changed to 1, on the second attempt it is changed to 2, and on the third attempt it is changed to 3.

If three attempts are made, then the container 'attempts' equals 3 and if there is still no correct password, then the account is locked.



**Download Worksheet 1.3 from Cambridge Elevate**

Containers like 'attempts' are used in algorithms to store values that can change as the algorithm is running. As the values they contain can change, they are called variables.