DIGITAL TECHNOLOGIES HUB

Australian Curriculum V9.0

Algorithms

The precise sequences of steps and decisions needed to solve a problem, often involving iterative (repeated) processes

ACARA, 2022

Teaching about algorithms F–10

View your year band for an overview of the content related to algorithms. It also includes suggested related content so you can integrate with another learning area.

Approaches include:

Complexity

Match an algorithm to an outcome
- Provide different algorithms and match to the task it solves, then follow.

Arrange and follow a series of steps in correct order
- Provide steps of an algorithm to order correctly.

Fill in the gap of a missing step
- Provide an algorithm with a step missing in the sequence.

Identify elements of an algorithm
- Which part is where we make a decision, is anything repeated?

Compare two different algorithms for the same problem
- Do they each work? Is one more efficient than the other?

Fix bugs in algorithms
- Review and modify an algorithm that includes too many assumptions, is inaccurate or is inefficient.

Create an algorithm
- Use computational thinking to solve a problem and create an algorithm.

Write an algorithm in pseudocode
- Convert algorithms from Pseudocode into code.

Trace and validate algorithms
- Use input data to test and check output against expected values.

Foundation

This concept does not appear in the Australian Curriculum: Digital Technologies in Foundation.

There is related content in Design Technologies and English.

Related content

Sequence pictures to retell a story.

English	Retell and adapt familiar literary texts through play, performance, images or writing | English AC9EFLE05

Related content

Follow a series of steps when designing a solution to a problem using available materials.

Design Technologies	Generate, communicate and evaluate design ideas, and use materials, equipment and steps to safely make a solution for a purpose | Design Technologies AC9TDEFP01

Years 1–2

I can follow and describe ordered steps. They can include simple choices and parts that repeat.

An algorithm describes a sequence of steps and decisions. It can be spoken as instructions, written as a list, or presented as a series of images.

While following an algorithm, often by physically acting out the steps, we can see if the algorithm achieves the intended outcome.

For example, design and follow **a series of steps** and **decisions** to make a sandwich or rice paper roll.

The order of certain steps may affect the outcome. For example, 'Slice the vegetables' is before 'Spread the ingedients onto the rice paper and roll'.

Branching is when the steps include a decision with a *yes* or *no* answer.

Have you fed the dog?

No → Put food in the bowl

Yes → Don't put food in the bowl

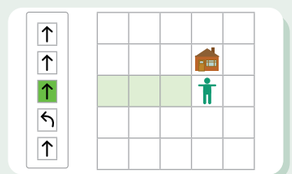
Sometimes a single step is repeated multiple times. This is called **iteration**. We can look at more efficient ways of describing repeated steps.

Pick up the basketball
Bounce the ball 5 times
Take a shot at goal.
Collect the ball.
Place the ball back.

Achievement standard	Students follow and describe basic algorithms involving a sequence of steps and branching.
Content descriptions	Follow and describe algorithms involving a sequence of steps, branching (decisions) and iteration (repetition) | AC9TDI2P02

Related content

Give and follow simple directions to move from one place to another using familiar reference points.



Mathematics	Give and follow directions to move people and objects to different locations within a space | Mathematics AC9M1SP02

DIGITAL TECHNOLOGIES HUB

Australian Curriculum V9.0

Algorithms

The precise sequences of steps and decisions needed to solve a problem, often involving iterative (repeated) processes

ACARA, 2022

Years 3–4

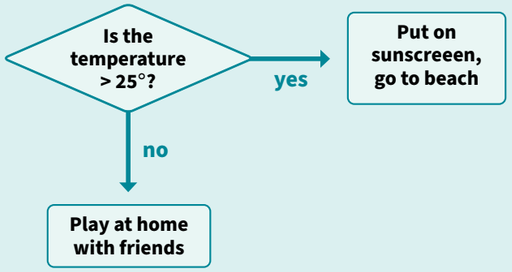
I can follow and describe the steps that include decisions based on comparing data.

An algorithm can describe a sequence of steps and decisions that include comparing data.

Sequencing refers to placing the steps to be performed in order, one after another.

At certain steps in the sequence we may want an input to make a decision. To get a yes or no answer, branching decisions may result from a **comparison**.

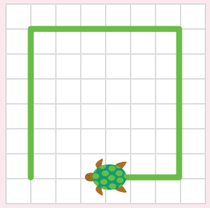
The **operator** may be: < (less than), > (greater than), or = (equal to).



With **iteration**, some steps can happen a set number of times.

To move along a square pathway, repeat these steps 4 times:

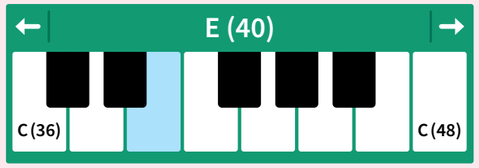
Move forward 5 steps, then turn right 90°.



Set instrument to piano.

Play these musical notes, repeat 14 times.

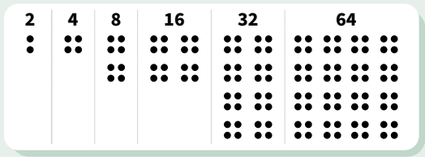
- **E for 0.5 beat**
- **B for 0.5 beat**
- **G for 0.5 beat**
- **E for 0.5 beat**
- **B for 0.5 beat**
- **C# for 0.5 beat**
- **G for 0.5 beat**



Achievement standard	Students follow and describe simple algorithms involving branching and iteration.
Content descriptions	Follow and describe algorithms involving sequencing, comparison operators (branching) and iteration | Digital Technologies AC9TDI4P02

Related content

To get 64, start with 2, then repeat this step 5 times: multiply by 2.



Mathematics	Follow and create algorithms involving a sequence of steps and decisions that use addition or multiplication to generate sets of numbers; identify and describe any emerging patterns | Mathematics AC9M4N09

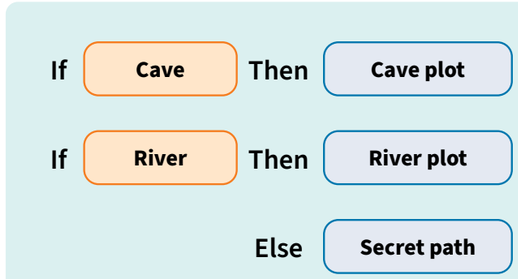
Years 5–6

I can design algorithms with decisions that can lead to multiple outcomes.

An algorithm can describe a sequence of steps and include multiple decisions.

An algorithm can be simple or complex, but it always follows a **sequence** from a starting point.

In algorithms, if/then statements allow for different paths. For example, in a 'choose your own adventure' story, IF the choice is ‘cave’, THEN the story changes to cave plot. IF the choice is ‘river’, THEN the story changes to a river plot. Otherwise (ELSE), the story continues with the secret path plot.



Yes/No questions can be used as a sorting algorithm to identify one object from a group, for example, sorting a group of animals using a series of Yes/No decisions (branching).

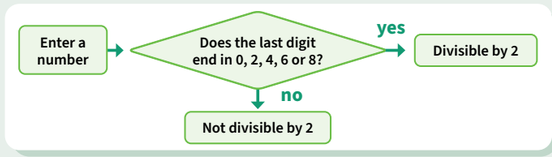
With **iteration**, some steps may be repeated only as long as a condition holds.

- Keep heating UNTIL temperature = 22 ˚C .
- Keep playing a game UNTIL 3 lives are lost, keep count of lives. Subtract one for each unsuccessful attempt.

Achievement standard	Students design algorithms involving complex branching and iteration.
Content descriptions	Design algorithms involving multiple alternatives (branching) and iteration | Digital Technologies AC9TDI6P02

Related content

Create a flow chart to determine if numbers are divisible by 2.



Mathematics	Create and use algorithms involving a sequence of steps and decisions and digital tools to experiment with factors, multiples and divisibility; identify, interpret and describe emerging patterns | Mathematics AC9M5N010

DIGITAL TECHNOLOGIES HUB

Australian Curriculum V9.0

Algorithms

The precise sequences of steps and decisions needed to solve a problem, often involving iterative (repeated) processes

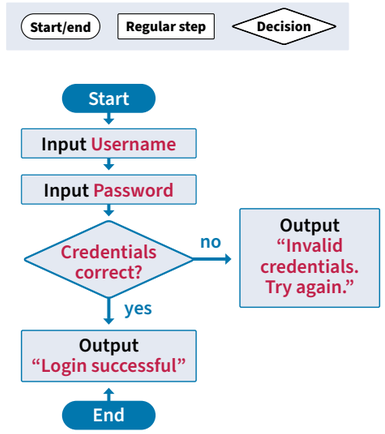
ACARA, 2022

Years 7–8

My algorithms involve multiple decisions and are designed using established conventions. I can manually step through them to understand their execution.

An algorithm can describe a sequence of steps and decisions using a flowchart or pseudocode.

A **flowchart** is a diagram that represents a set of instructions using standard symbols.



Pseudocode isn't a programming language but a less formal text with basic conventions. It includes INPUT for questions and DISPLAY/OUTPUT for screen messages. It allows nested control structures like IF-THEN-ELSE within a FOR-NEXT loop.

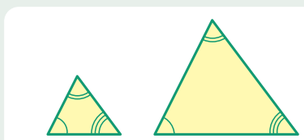
```
success ← false
WHILE NOT success DO
    username ← INPUT('Enter your username:')
    password ← INPUT('Enter your password:')
    IF username = storedUsername AND password = storedPassword
THEN
        success ← true
    ELSE
        DISPLAY 'Invalid credentials. Try again.'
    ENDIF
ENDWHILE
DISPLAY 'Login successful.'
```

To **trace an algorithm**, follow each step as if you were a computer or robot running the program. Take note of outputs and variable values as needed.

Achievement standard	Students design and trace algorithms and implement them in a general-purpose programming language.
Content descriptions	Design algorithms involving nested control structures and represent them using flowcharts and pseudocode | Digital Technologies AC9TDI8P05 Trace algorithms to predict output for a given input and to identify errors | Digital Technologies AC9TDI8P06

Related content

Create an algorithm to sort and classify triangles based on congruency.



Mathematics	Design, create and test algorithms involving a sequence of steps and decisions that identify congruency or similarity of shapes, and describe how the algorithm works | Mathematics AC9M8SP04

Years 9–10

The decisions in the algorithms I create are based on more complex and formalised conditions. I can also test them with appropriate inputs.

An algorithm can describe a sequence of steps and decisions using pseudocode or a flowchart and can show complex branching or looping.

To **validate** an algorithm, test it with varied input data that you have selected *intentionally*. Does the algorithm respond as it should, or does it need to be improved? Typically this is done by classifying ranges of input values and showing that the algorithm produces expected results for boundary values of the range and all values in between.

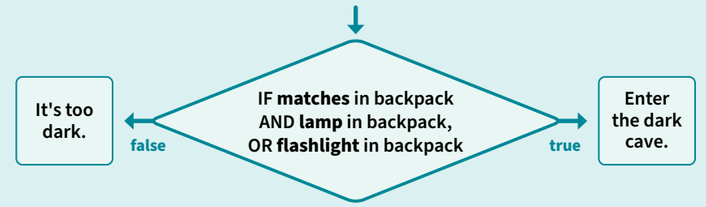
How many books were bought?
>> 0
Try again. Minimum is 1.

How many books were bought?
>> 1
OK. Total cost is $39.95.

How many books were bought?
>> 2
OK. Total cost is $79.90.

How many books were bought?
>> dog
OK. Total cost is $FATAL ERR!

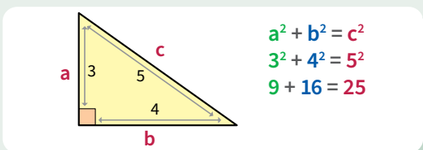
Sometimes the condition for branching or looping is more complex than a simple comparison check. The **logical operators** AND, OR and NOT allow combined conditions.



Achievement standard	Students design and validate algorithms and implement them, including in an object-oriented programming language.
Content descriptions	Design algorithms involving logical operators and represent them as flowcharts and pseudocode | Digital Technologies AC9TDI10P05 Validate algorithms and programs by comparing their output against a range of test cases | Digital Technologies AC9TDI10P06

Related content

Create an algorithm using pseudocode or flowcharts to generate Pythagorean triples.



Mathematics	Design, test and refine algorithms involving a sequence of steps and decisions based on geometric constructions and theorems; discuss and evaluate refinements | Mathematics AC9M9SP03