

## Years 3–4

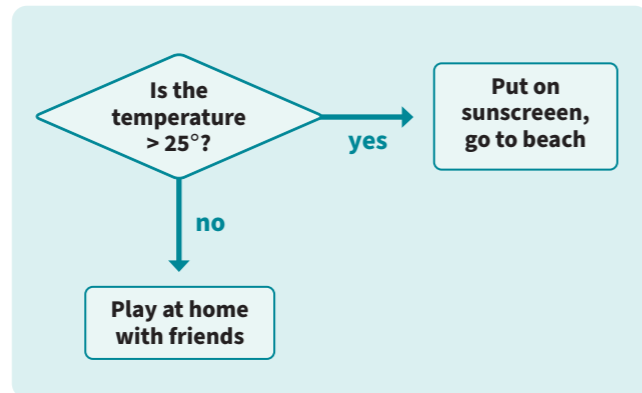
I can follow and describe the steps that include decisions based on comparing data.

**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).



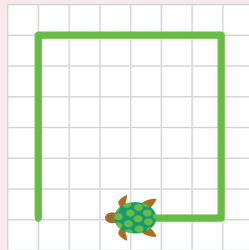
For suggested resources



<https://bit.ly/AlgorithmsYears3and4>

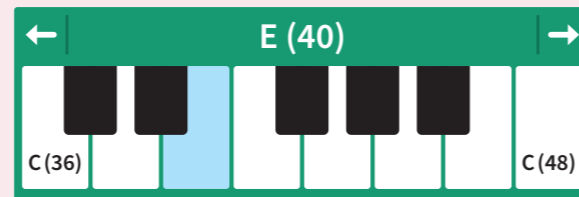
With **iteration**, some steps can happen a set number of times.

**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



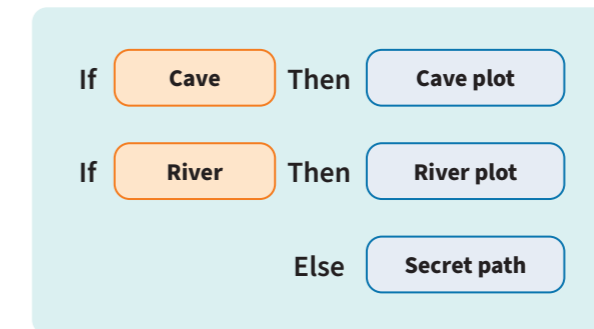
## Years 5–6

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.



For suggested resources



<https://bit.ly/AlgorithmsYears5and6>

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.

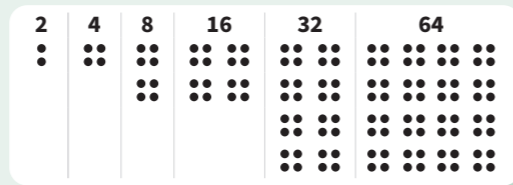


**Achievement standard** Students follow and describe simple algorithms involving branching and iteration.

**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.



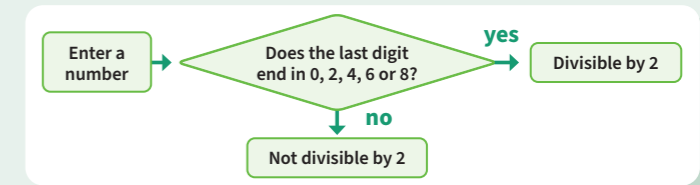
**Mathematics** Follow and create algorithms involving a sequence of steps and decisions that use addition or multiplication to generate sets of numbers; identify and describe any emerging patterns | Mathematics AC9M4N09

**Achievement standard** Students follow and describe simple algorithms involving 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