BABAJI
VIDHYASHRAM

Grade 2

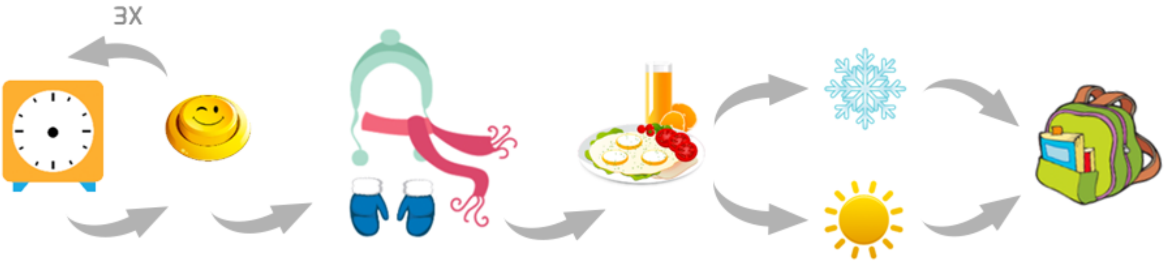
Algorithm

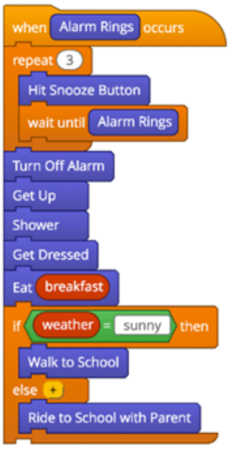
The word “algorithm” are all around us, governing everything from the technology they use to the mundane decisions they make every day. Algorithms are fascinating and, although some are quite complex, the concept itself is actually quite simple.

What is an Algorithm?

An algorithm is a detailed step-by-step instruction set or formula for solving a problem or completing a task. When you think of an algorithm in the most general way, algorithms are everywhere.

A recipe for making food is an algorithm, the method you use to solve addition is an algorithm, and the process of folding a shirt or a pair of pants is an algorithm. Even your morning routine could be considered an algorithm! In fact, here’s what your morning might look like written out as an algorithm.





```
when Alarm Rings occurs
repeat 3
    Hit Snooze Button
    wait until Alarm Rings
Turn Off Alarm
Get Up
Shower
Get Dressed
Eat breakfast
if weather = sunny then
    Walk to School
else +
    Ride to School with Parent
```

What is coding?

Coding is the process of using a computer language to get a computer how you want it to. Every line of code tells the computer to do something.

An algorithm is nothing but a set of instructions to be followed. When you multiply two and two, then that is also an algorithm because you are following a certain step-by-step procedure.

The Aadhar card, which is evolved as the country's universal identity card, is based on biometrics, which in turn use an algorithm to store and retrieve fingerprints and iris scans.

How to write an algorithm?

1. Determine the outcome of your code.

   What is the specific problem you want to solve or the task you want to accomplish?

   Once you have a solid idea of what you are aiming to accomplish, you can determine the steps it will take to get there.

2. Decide on a starting point.

   Finding your starting and ending point are crucial to listing the steps of the process.

3. Find the ending point of the algorithm.

   As with the starting point, you can find the end point of your algorithm.

4. List the steps from start to finish.

   Start with broad steps. To use a real-world example, let's say your goal is to have ice cream for dinner. You have determined that the striating pound is to find a recipe, and that the end result is that you will have an ice cream and ready to eat by 7 pm. Your steps may look like this:

- Search for the recipe online.
- Look for the ingredients you already have in the kitchen.
- Make a list of ingredients you will need from the store.
- Buy the missing ingredients.
- Return home.
- Prepare the ice cream.
- Remove the ice cream from the fridge.

5. Determine how you will accomplish each step.

Now that you have a step-by-step outline, it's time to think about how you might code each step.

For example, the first step in our ice cream algorithm is **Search for a recipe online.** But what is involved in this search? Be specific. For example:

- Turn on your computer.
- Check to make sure you are connected to the internet. Connect to the internet if you are not already.
- Open a web browser.
- Enter your search items.
- Click a recipe link.
- Determine whether the recipe meets your needs.
- Repeat some of these steps until you find the right recipe.

6. Review the algorithm.

Now that you have written your algorithm, it's time to evaluate the process. Your algorithm is designed to accomplish something specific, and you will need it to start writing your program. Ask yourself the following questions, and address each as necessary:

- Does the algorithm solve the problem/accomplish the task?
- Does it have clearly defined inputs and outputs?
- Should the end goal be redefined to be more general? More specific?
- Can any of the steps be simplified?
- Is the algorithm guaranteed to end with the correct result?

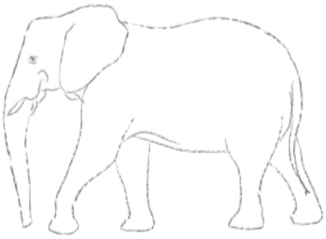
Algorithms are precise step-by-step instructions on how to accomplish a desired task. A formal algorithm for use with computers or in mathematics must be very detailed and resolve the ambiguities that we take for granted in everyday life.

You can use algorithms to help describe things that people do everyday. Let's create an algorithm of planting a seed.

Number the steps in the order from the beginning till the last for the following picture.

PUT POT IN SUNLIGHT

PUT SEED IN HOLE


HUG AN ELEPHANT

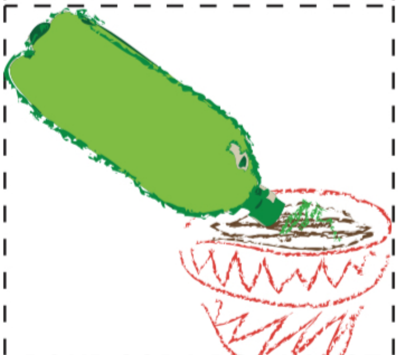
PUT GLUE ON SEED

FILL POT WITH SOIL

POKE HOLE IN SOIL

WATER POT

COVER SEED WITH SOIL


POUR SODA POP IN POT

If-then-else statements

An **if** statement can be followed by an optional **else** statement, which executes when the expression is false.

Example:

```
If
{
  I have homework
}

then
{
  I have to finish it
}

else
{
 My teacher will punish
}
```

If the expression evaluates to **true**, then the **if block** will be executed, otherwise, the **else block** will be executed.

Q. Write any 3 if-then-else statements your own.

- - - EXERCISES - - -

1) **Assignment**

You need to pick an everyday or interesting task and create an algorithm that explains how to do it.

- You need to specify what (if anything) is input (like ingredients in a recipe)
- The process by which you operate on that input to produce or accomplish a task
- What the output of the algorithm is

What to turn in

Type your algorithm up using the following example as a template. Name the task your algorithm describes, the input and output, and then list the steps involved, numbering each, followed by what your algorithm assumes.

Example

Algorithm: Calling a friend on the telephone

Input: The telephone number of your friend.
Output: None

Steps:

1. Pick up the phone and listen for a dial tone
2. Press each digit of the phone number on the phone
3. If busy, hang up phone, wait 5 minutes, jump to step 2
4. If no one answers, leave a message then hang up
5. If no answering machine, hang up and wait 2 hours, then jump to step 2
6. Talk to friend
7. Hang up phone

2) Write an algorithm to add any two numbers.

3) Write an algorithm of how to commute to computer lab from your class.