














# Controlling Devices: Flowol: Subroutines

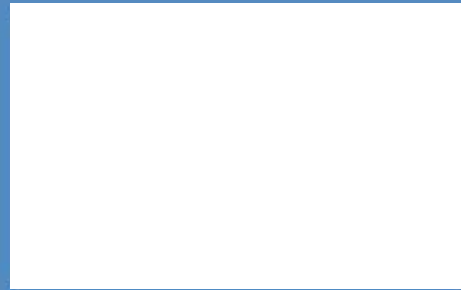
<p><b>Aim:</b> Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.</p> <p>Use sequence, selection, and repetition in programs; work with variables and various forms of input and output.</p>	<p><b>Success Criteria:</b> I can create a subroutine separate to a main flowchart program. I can call a subroutine from the main flowchart program. I can call multiple subroutines within a program.</p>	<p><b>Resources:</b> <a href="#">Lesson Pack</a> PC /laptop with Flowol 4 software</p>
<p>Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.</p> <p>Children are introduced to the concept of subroutines by editing an already familiar flowchart (Lighthouse).</p> <p>I can create a flowchart program containing a subroutine.</p>	<p><b>Key/New Words:</b> Flowol, flowchart, algorithm, control, output, mimic, simulation, insert, symbol, start, stop, delay, process, decision, input loop, subroutine.</p>	<p><b>Preparation:</b> Differentiated <a href="#">Subroutines Activity Sheets</a> - as required <a href="#">Pelican Crossing Activity Sheet</a> - as required</p>

**Prior Learning:** Children will have created flowcharts using inputs and outputs in lessons 1 to 4.

## Learning Sequence

	<p><b>Programming People:</b> Choose a child and give them a short sequence of instructions to perform (e.g. stand up, wave arms, turn around). Explain that every time they hear their name in the lesson, they must perform these actions. Do the same for another child with a different set of instructions. Test by beginning a short story or monologue which mentions the name of each child in turn and check that they perform their actions on cue.</p>	
	<p><b>Subroutines in Programs:</b> Explain that the actions of the selected children (saying their names again could prompt a repeat of their actions!) are similar to a subroutine in a program. Instead of all the instructions being in one long sequence, a flowchart can be designed to 'call' a subroutine. This is like a separate mini program that is performed on command.</p>	
	<p><b>Lighthouse Subroutine:</b> Show the example of the subroutine for the Lighthouse mimic using the <a href="#">Lighthouse 3 Flowol file</a>. Demonstrate how to start a subroutine by using the Start /Sub symbol, for a separate flowchart sequence, and giving the subroutine a name. This brings a new symbol into the menu that allows the subroutine to be called from the main flowchart.</p>	
	<p><b>Programming a Subroutine:</b> Children design a flowchart to incorporate a subroutine, called from the main program. The subroutine could be for a more complicated sequence of actions for the light and foghorn.</p> <div style="display: flex; justify-content: space-around;"> <div data-bbox="215 1444 566 1736"> <p> Follow instructions to copy and create a version of the flowchart program to control the lighthouse so that it contains a subroutine. Use the <a href="#">Subroutines Activity Sheet</a> for a copy of flowcharts.</p> </div> <div data-bbox="614 1444 981 1680"> <p> Design or edit an existing flowchart program to control the lighthouse so that it contains a subroutine. Use the <a href="#">Subroutines Activity Sheet</a> for additional guidance.</p> </div> <div data-bbox="1029 1444 1380 1848"> <p> Design or edit an existing flowchart program to control the lighthouse so that it contains a subroutine. Use the <a href="#">Subroutines Activity Sheet</a> for additional guidance. Then, design a new set of subroutines for the Pelican Crossing mimic using the <a href="#">Pelican Crossing Activity Sheet</a>.</p> </div> </div>	
	<p><b>Predict, Test and Debug:</b> Choose an example solution that some of the children have created for the lighthouse mimic and, if possible, the pelican crossing mimic. Examine the subroutines used. Predict whether the program will work, then run and test. Debug if necessary.</p>	

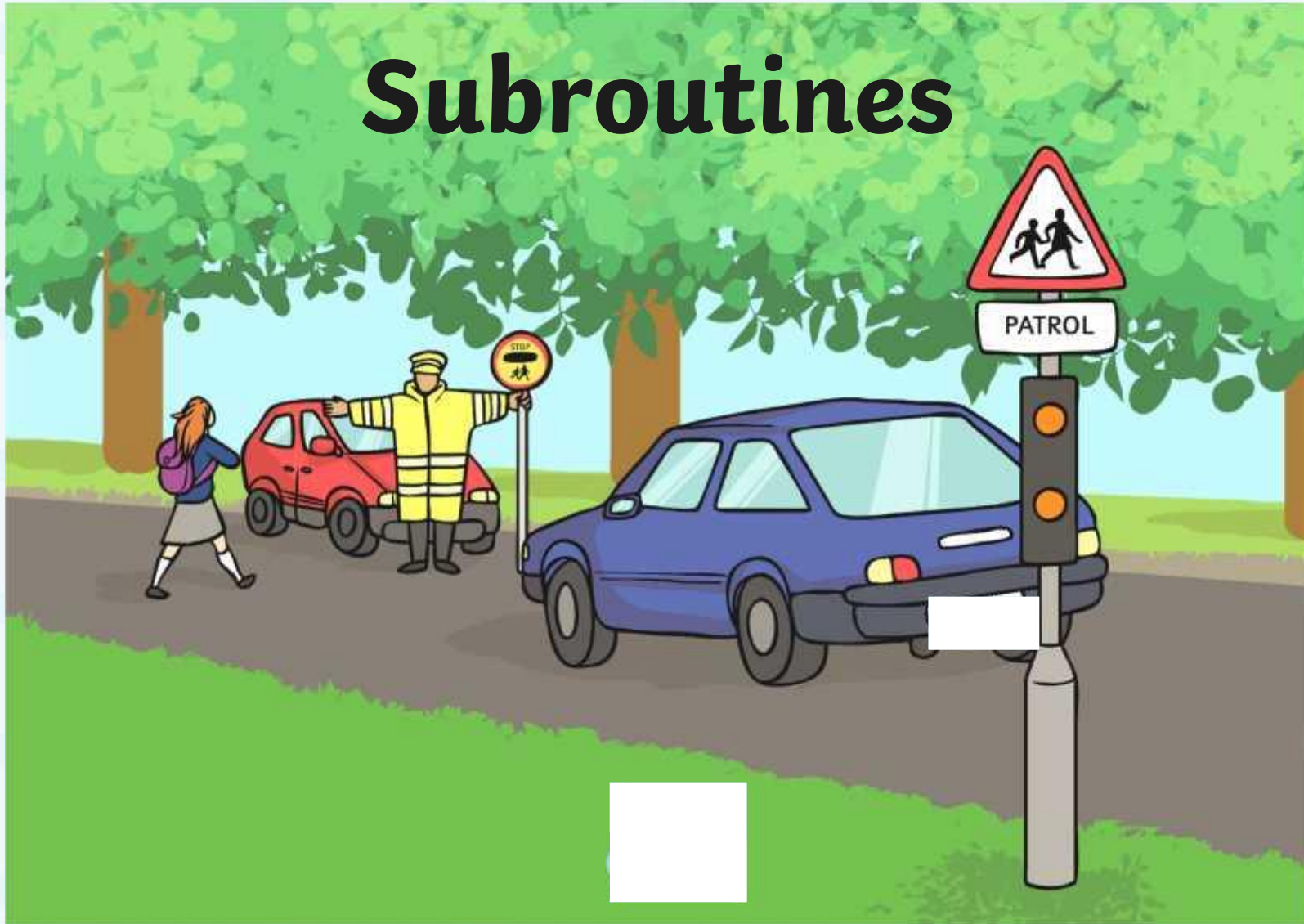
**Taskit**  
**Paintit:** Use your work on the lighthouse to paint a lighthouse scene. Create either a day or night time scene.



# Computing

## Controlling Devices: Flowol

# Subroutines



# Aim

- I can create a flowchart program containing a subroutine.

# Success Criteria

- I can create a subroutine, separate to a main flowchart program.
- I can call a subroutine from the main flowchart program.
- I can call multiple subroutines within a program.

# Programming People



Instead of a computer controlled device, we are going to imagine programming instructions for two people.

## First person

**When your name is called you need to:**

- Stand up
- Wave your hand
- Turn around
- Sit back down



## Second person

**When your name is called you need to:**

- Stand up
- Jump up
- Touch your toes
- Sit back down

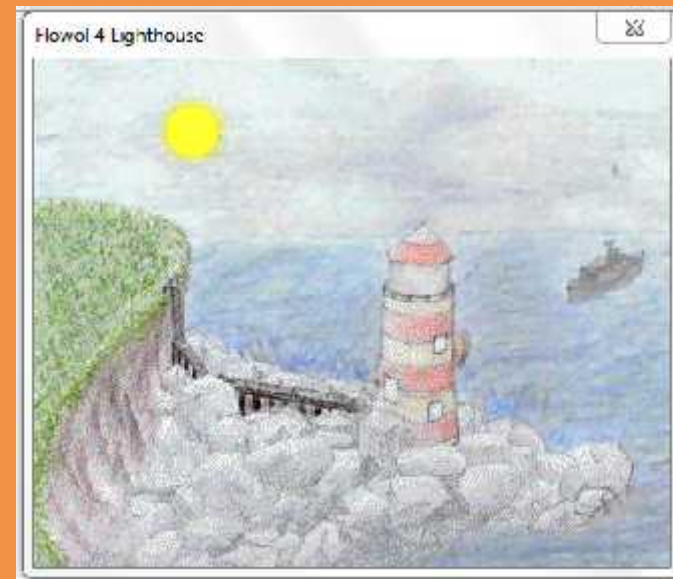


# Subroutines in Programs

The 'programming people' actions are similar to a subroutine in a computer program.

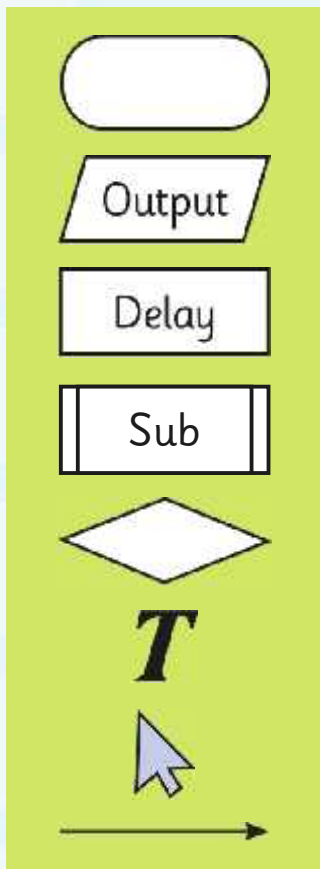
Instead of all the instructions being in one long sequence, a flowchart can be designed to 'call' a subroutine.

This is like a separate mini program that is performed when called by name.

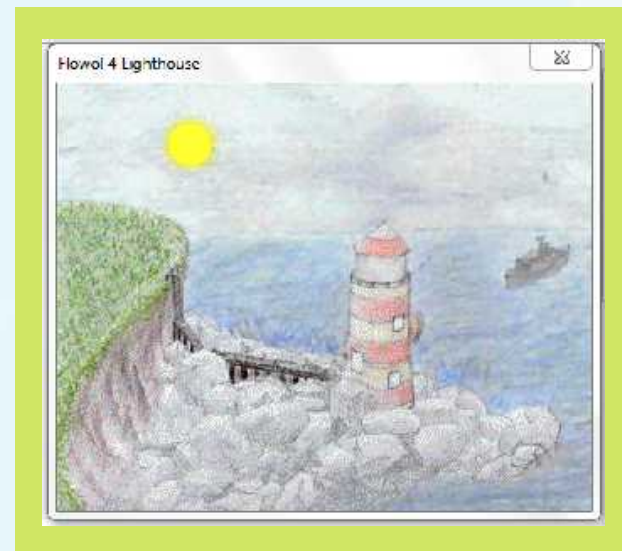


# Lighthouse Subroutine

Look at the example of the subroutine for the Lighthouse mimic in Flowol.



A new subroutine can be started by using the Start/Sub symbol, for a separate flowchart sequence, and for giving the subroutine a name.



This brings a new symbol into the menu that allows the subroutine to be called from the main flowchart.

# Programming a Subroutine



Can you design a flowchart to incorporate a subroutine, called from the main program?

Begin by using the Lighthouse mimic in Flowol. The subroutine could be for a more complicated sequence of actions for the light and foghorn.

Use the activity sheet to help you.

**Subroutines**

The aim of this activity is to create a main program and a separate subroutine that can be called from within it.

This activity will require you to use all the boxes and symbols, if the sun is not shining (off).

Create new functions with the STARTSLB symbol.

After placing it in the workspace use a SER symbol and give it a name, eg Flash.

Now create your own function sequence for turning the lamps and foghorn on and off with different intervals. It could be the same as the one you could experiment and adapt it to make it your own!

Then create your main program:

Use a new STARTSLB symbol and place it to the side, selecting STONE.

Use a decision symbol to check whether the sun is shining (on).

If YES, then the lighthouse lamp can be turned OFF.

If NO, then you can now insert a SUB symbol to call the sub routine (flowchart) that you created.

Run and test your program, checking what happens when the sun is on or off!

Check whether sun is a variable?  
If YES, lamp is turned off.  
If NO, subroutine Flash is called.

The activity sheet contains a flowchart with a 'Start' oval, a decision diamond 'Is sun shining?', and two paths: 'Yes' leading to a 'Flash' parallelogram, and 'No' leading to a 'Turn off lamp' parallelogram. A 'Subroutine Flash' box is shown to the right, containing a 'Turn on lamp' parallelogram and a 'Turn off foghorn' parallelogram. A lighthouse illustration is at the bottom right.



# Predict, Test and Debug



We are going to choose an example solution that some of you have created to share the lighthouse mimic and, if possible, the pelican crossing mimic.

Firstly can you examine the program and subroutines used to predict whether the program will work?

Then we will run, test it and debug if necessary.



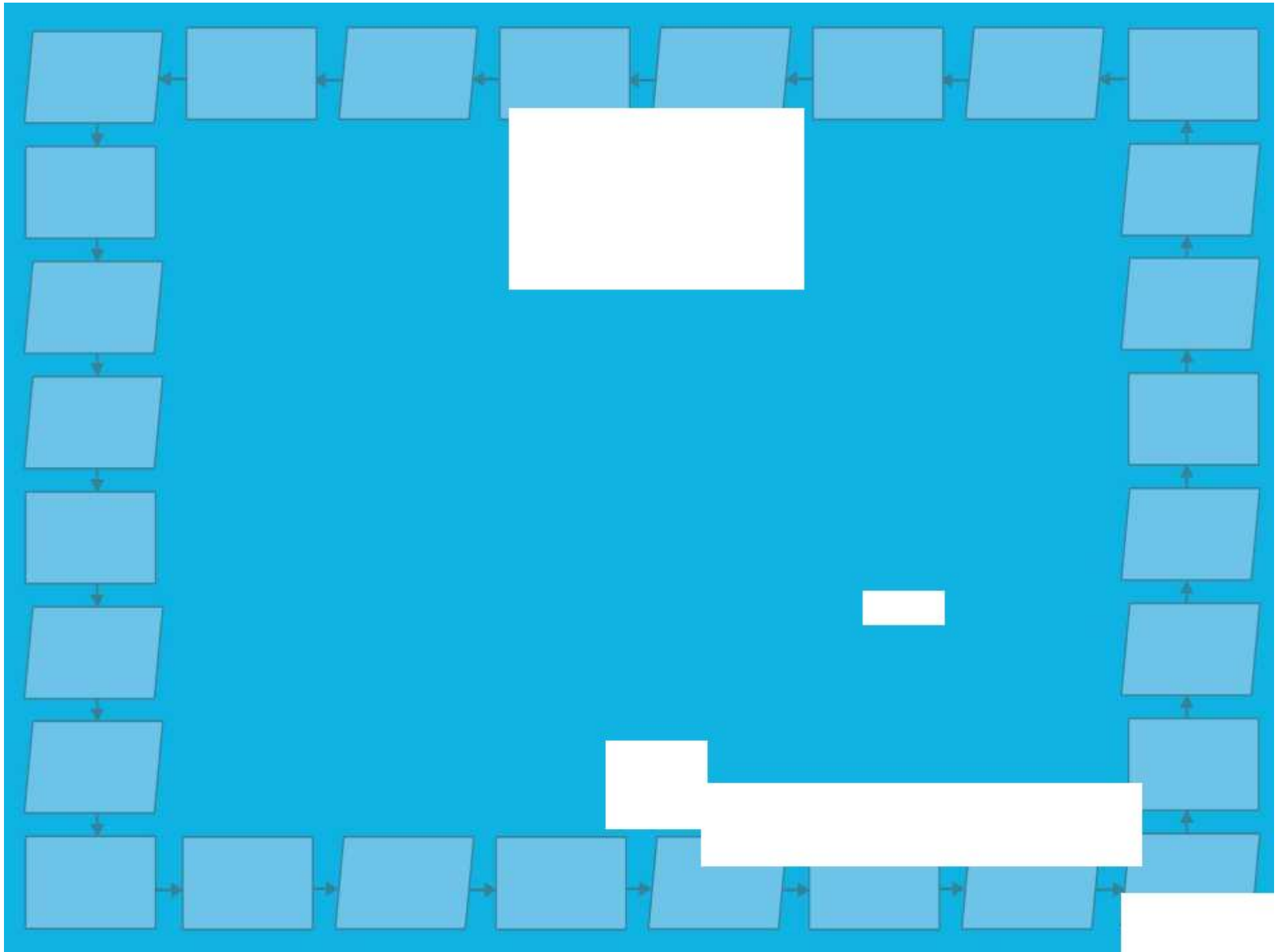
# Aim



- I can create a flowchart program containing a subroutine.

# Success Criteria

- I can create a subroutine, separate to a main flowchart program.
- I can call a subroutine from the main flowchart program.
- I can call multiple subroutines within a program.





# Subroutines

The aim of this activity is to create a main program and a separate subroutine that can be 'called' from within it.

First create a 'subroutine' to control the lamp and foghorn, if the Sun is not shining ('off').

Start a new flowchart with the START/SUB symbol.

After placing it in the workspace area, select SUB and give it a name, e.g. Flash.

Now create your own flowchart sequence for turning the lamps and foghorn on and off with delays in between. It could be like the one here or you could experiment and adapt it to make it your own!

Next create your main flowchart.

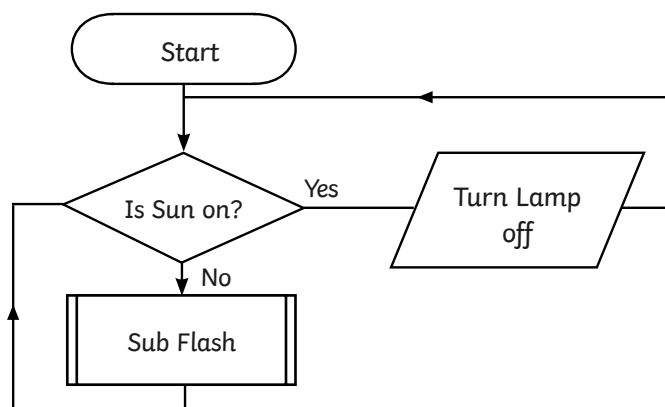
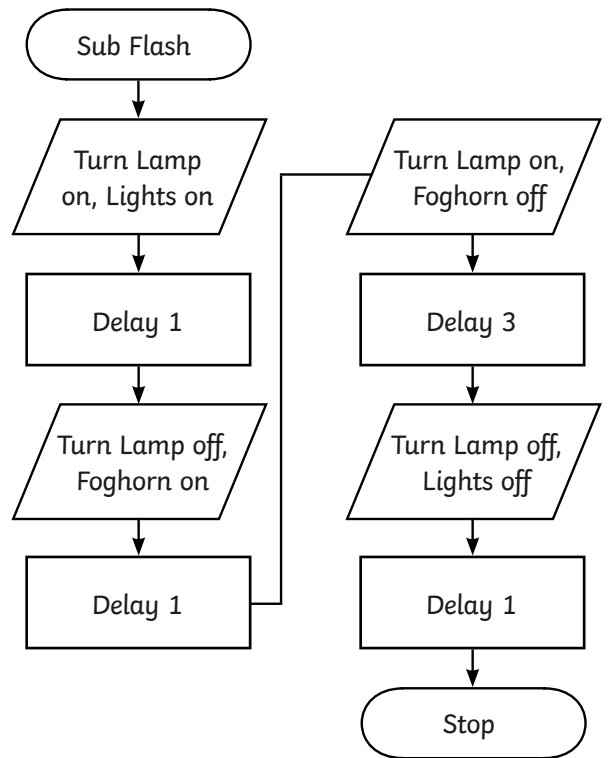
Use a new START/SUB symbol and place it at the side, selecting START.

Use a decision symbol to check whether the Sun is shining ('on').

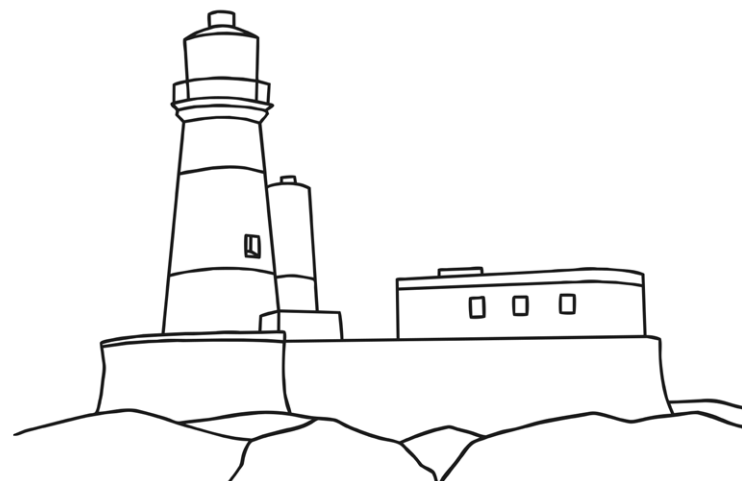
If YES, then the lighthouse lamp can be turned OFF.

If NO, then you can now insert a SUB symbol to call the subroutine if the Sun is not shining.

Run and test your program, checking what happens when the Sun is 'on' or 'off'.



Checks whether Sun is on (shining).  
 If YES, lamp is turned off.  
 If NO, subroutine Flash is called.





# Subroutines

The aim of this activity is to create a main program and a separate subroutine that can be 'called' from within it.

First create the beginning of your main flowchart.

Use a decision symbol to check whether the Sun is shining ('on').

If YES, then the lighthouse lamp can be turned OFF.

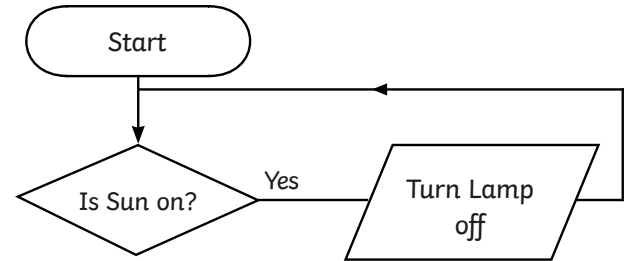
Instead of continuing with this flowchart, you can create a separate 'subroutine' to control the lamp and foghorn, if the Sun is not shining ('off').

Start a new flowchart at the side with the START/SUB symbol.

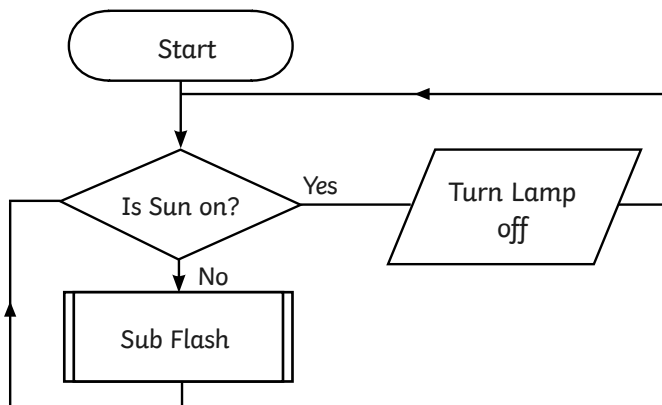
After placing it in the workspace area, select SUB and give it a name, e.g. Flash.

Now create your own flowchart sequence for turning the lamps and foghorn on and off with delays in between.

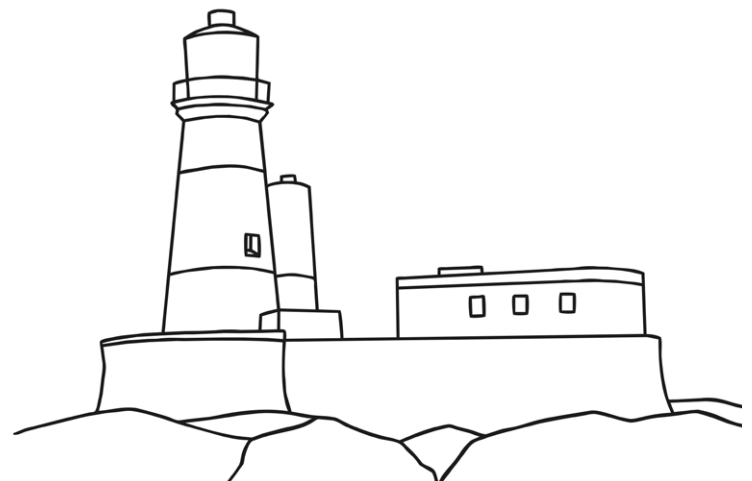
Finally, in your main flowchart, you can now insert a SUB symbol to call the subroutine if the Sun is not shining. Run and test your program, checking what happens when the Sun is 'on' or 'off'.



Checks whether Sun is on (shining).  
If YES, lamp is turned off.



Checks whether Sun is on (shining).  
If YES, lamp is turned off.  
If NO, subroutine Flash is called.





# Subroutines

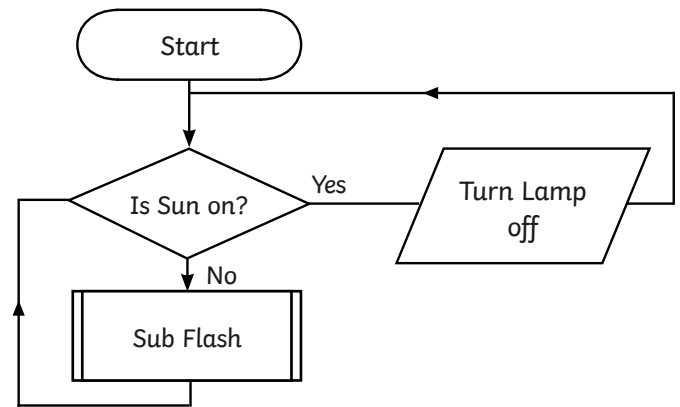
The aim of this activity is to create a main program and a separate subroutine that can be 'called' from within it.

## 1: Lighthouse

First create the beginning of your main flowchart. Use a decision symbol to check whether the Sun is shining ('on').

If YES, then the lighthouse lamp can be turned OFF. Instead of continuing with this flowchart, you can create a separate 'subroutine' to control the lamp and foghorn, if the Sun is not shining ('off').

Start a new flowchart at the side with the START/SUB symbol. After placing it in the workspace area, select SUB and give it a name, e.g. Flash. Now create your own flowchart sequence for turning the lamps and foghorn on and off with delays in between.



Checks whether Sun is on (shining).  
If YES, lamp is turned off.  
If NO, subroutine Flash is called.

Finally, in your main flowchart, you can now insert a SUB symbol to call the subroutine if the Sun is not shining. Run and test your program, checking what happens when the Sun is 'on' or 'off'.

## 2: Pelican Crossing

Start a new program using the Pelican Crossing mimic. Begin by exploring the Input (a Push Switch which goes off after half a second) and the outputs (Red, Amber, Green traffic lights, Red man, Green Man, Wait light).



Create a flowchart that can call 3 or 4 different subroutines, depending on the scenario. Think about how a pelican crossing works and what the lights show at any given time.

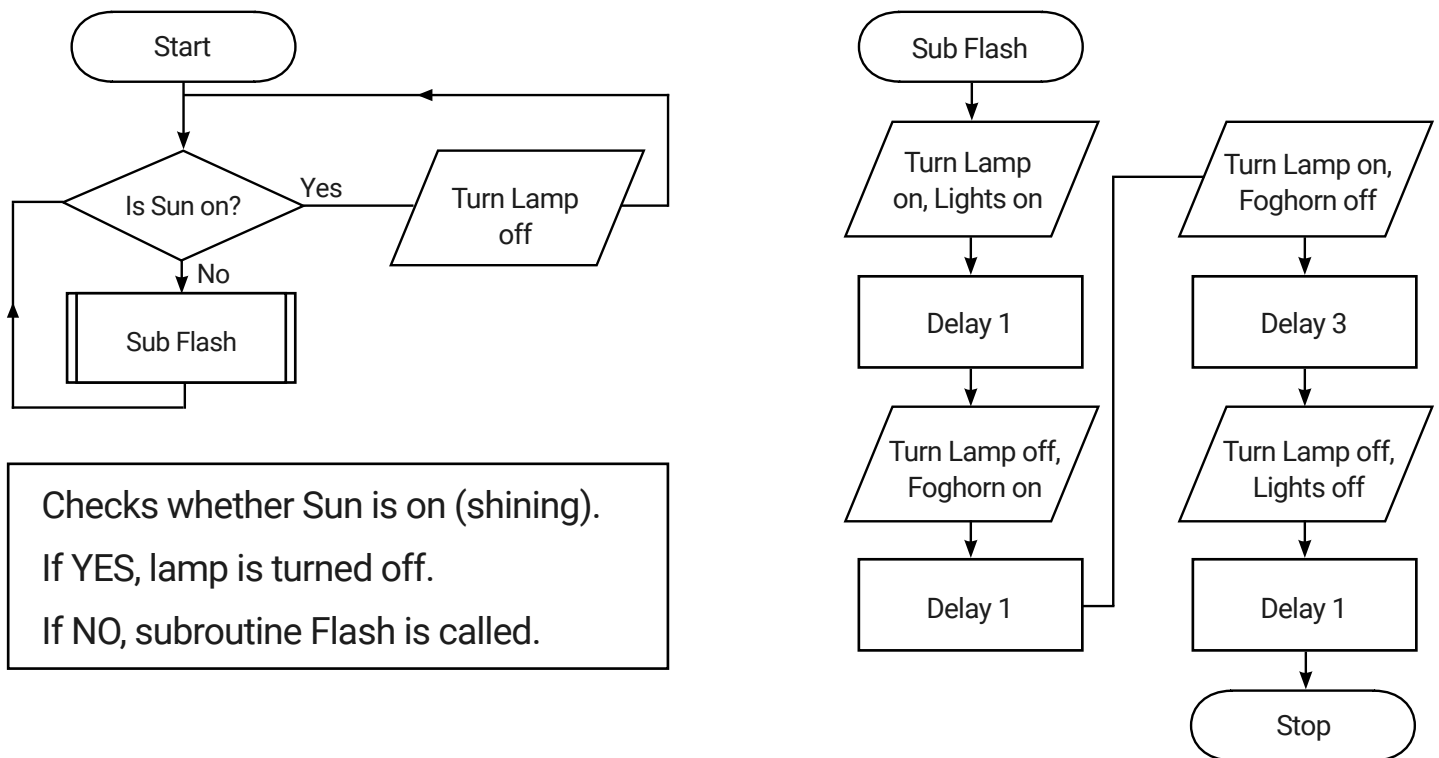
You may need subroutines for the cars to stop and go, and for the pedestrian to walk. There is the extra state of the Flashing green man, when you can continue crossing but the traffic lights are soon about to change back.

Experiment with the subroutines and remember to keep running and testing your program to check and debug!

# Adult Guidance

This lesson is designed for use with Flowol software (intended for use with Version 4, but adaptable to earlier versions) and the mimics for the Lighthouse (all children) and Pelican Crossing (Higher Ability children).

An example of a flowchart calling a subroutine for the Lighthouse mimic.



An example solution for the Pelican Crossing mimic is provided in the 'Flowol Files' folder. Children may have come up with alternative solutions.

Controlling Devices: Flowol | Subroutines

<b>I can create a flowchart program containing a subroutine.</b>		
I can create a subroutine separate to a main flowchart program.		
I can call a subroutine from the main flowchart program.		
I can call multiple subroutines within a program.		

Controlling Devices: Flowol | Subroutines

<b>I can create a flowchart program containing a subroutine.</b>		
I can create a subroutine separate to a main flowchart program.		
I can call a subroutine from the main flowchart program.		
I can call multiple subroutines within a program.		

Controlling Devices: Flowol | Subroutines

<b>I can create a flowchart program containing a subroutine.</b>		
I can create a subroutine separate to a main flowchart program.		
I can call a subroutine from the main flowchart program.		
I can call multiple subroutines within a program.		

Controlling Devices: Flowol | Subroutines

<b>I can create a flowchart program containing a subroutine.</b>		
I can create a subroutine separate to a main flowchart program.		
I can call a subroutine from the main flowchart program.		
I can call multiple subroutines within a program.		

Controlling Devices: Flowol | Subroutines

<b>I can create a flowchart program containing a subroutine.</b>		
I can create a subroutine separate to a main flowchart program.		
I can call a subroutine from the main flowchart program.		
I can call multiple subroutines within a program.		

Controlling Devices: Flowol | Subroutines

<b>I can create a flowchart program containing a subroutine.</b>		
I can create a subroutine separate to a main flowchart program.		
I can call a subroutine from the main flowchart program.		
I can call multiple subroutines within a program.		

Controlling Devices: Flowol | Subroutines

<b>I can create a flowchart program containing a subroutine.</b>		
I can create a subroutine separate to a main flowchart program.		
I can call a subroutine from the main flowchart program.		
I can call multiple subroutines within a program.		

Controlling Devices: Flowol | Subroutines

<b>I can create a flowchart program containing a subroutine.</b>		
I can create a subroutine separate to a main flowchart program.		
I can call a subroutine from the main flowchart program.		
I can call multiple subroutines within a program.		