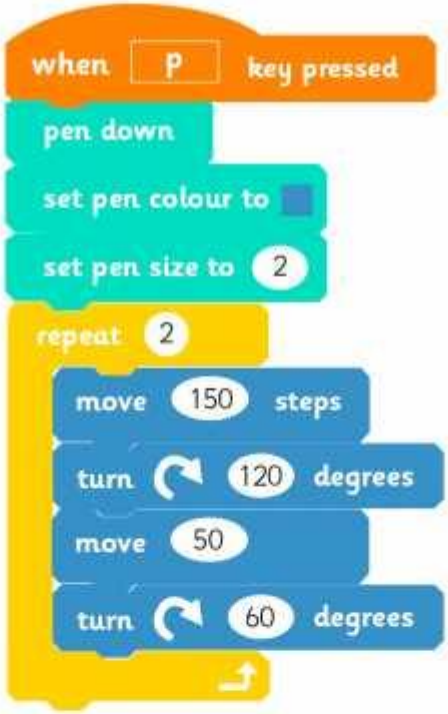
Scratch: Polygons and Patterns

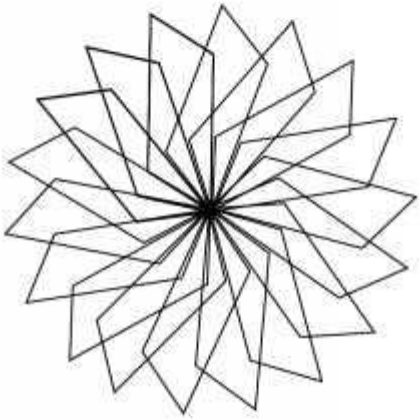
Type your aims and success criteria here.

Use Scratch for the following tasks:

1. Create algorithms for different quadrilaterals. Look at the example below for a parallelogram.



2. Using the algorithms you have created for the quadrilaterals, draw some interesting patterns.



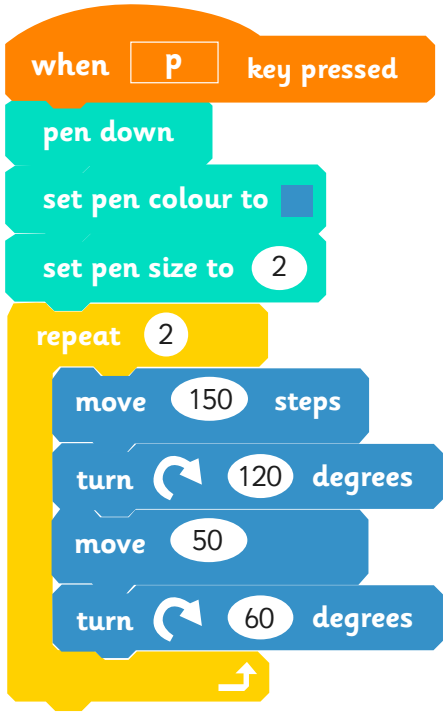
Note to Parents

Online versions of Scratch are available. There are also free downloadable applications – either v1.4 or v2 – both from the Scratch website. An algorithm is a set of precise instructions.

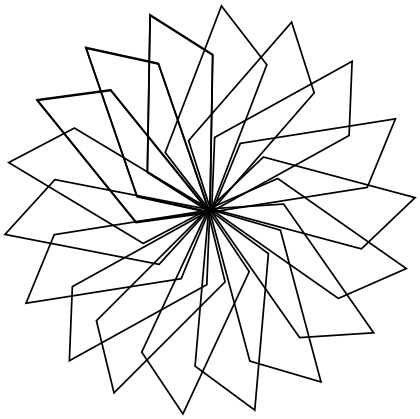
Scratch: Polygons and Patterns

Use Scratch for the following tasks:

1. Create algorithms for different quadrilaterals. Look at the example below for a parallelogram.



2. Using the algorithms you have created for the quadrilaterals, draw some interesting patterns.



Note to Parents

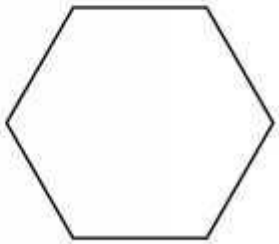
Online versions of Scratch are available. There are also free downloadable applications – either v1.4 or v2 – both from the Scratch website. An algorithm is a set of precise instructions.

Turtle Logo: Polygons and Patterns

Type your aims and success criteria here.

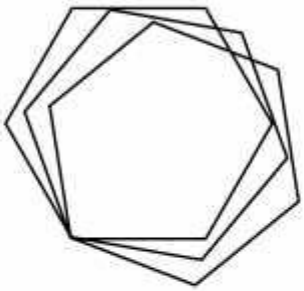
Use Turtle Logo for the following tasks:

1. Create algorithms for different regular polygons. Look at the example below.

Regular hexagon	repeat 6[fd 100 rt 60]	

To draw a different polygon change the number of times the algorithm repeats and the amount that it will turn. To find the correct angle to turn, divide 360 by the number of sides. For example, a square has 4 sides so 360/4 = 90. This means that the algorithm is repeat 4[fd 100 rt 90]

2. Create a pattern by drawing a hexagon and then turning the turtle a small amount (eg rt 10) and repeating the hexagon. Continue to repeat these commands until you have created a pattern.



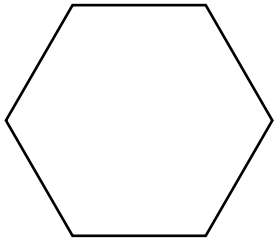
Note to Parents

Online versions of Logo are available, just search for "Turtle Logo" or "Turtle Academy".
An algorithm is a set of precise instructions.

Turtle Logo: Polygons and Patterns

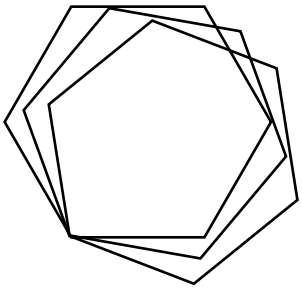
Use Turtle Logo for the following tasks:

1. Create algorithms for different regular polygons. Look at the example below.

Regular hexagon	repeat 6[fd 100 rt 60]	

To draw a different polygon change the number of times the algorithm repeats and the amount that it will turn. To find the correct angle to turn, divide 360 by the number of sides. For example, a square has 4 sides so 360/4 = 90. This means that the algorithm is repeat 4[fd 100 rt 90]

2. Create a pattern by drawing a hexagon and then turning the turtle a small amount (eg rt 10) and repeating the hexagon. Continue to repeat these commands until you have created a pattern.



Note to Parents
Online versions of Logo are available, just search for "Turtle Logo" or "Turtle Academy".
An algorithm is a set of precise instructions.

Programming Turtle Logo and Scratch: Backwards

Aim:	**Success Criteria:**	**Resources:**
Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts. Use sequence, selection, and repetition in programs; work with variables and various forms of input and output. Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs. This unit continues the learning from the Year 2 Turtle Logo units and links well to shape and direction in Maths. I can create and debug an algorithm using the move, rotate and repeat commands.	I can write commands in the correct order. I can write a variable value where required. I can correct any mistakes. I can use the commands fd, bk, lt, rt to move or rotate the turtle. I can use cs to clear the screen. I can use the repeat command.	**Lesson Pack** Desktop computer /laptop Turtle Logo application (installed or online) Whiteboards and pens or books, pens and pencils for recording.
	Key/New Words: Algorithm, instructions, commands, forward (fd), left (lt), right (rt), move, turn, clear screen (cs), variable.	**Preparation:** None needed

Prior Learning: It will be helpful if children can use and understand the commands; forward (fd), right (rt) and left (lt) alongside a variable.

Learning Sequence

	**What Can You Remember?** Ask the children what they can remember about programming Turtle Logo. Give the children a few minutes to remind themselves of how to draw a square, rectangle and a rectilinear letter L.	
	**Can You Go Backwards?** Talk partners discuss how to make the turtle move backwards then feedback their ideas to the class.	
	**Turtle Logo Commands:** Make sure all of the children can draw rectangles and squares using the repeat command. Demonstrate how backward or bk can be used to move backwards.	
	**Repeating Squares:** Demonstrate drawing a set of growing squares, all starting from the same place. **Snipping Tool:** Show how to snip an area of the screen and save the snipped picture. **Screenshot:** Alternatively, screenshot using print screen and copy into paint using select, crop and save.	
	**Growing Squares and Rectangles:** Using the differentiated **Backwards Activity Sheet** children draw repeating squares and rectangles.  Children are given the algorithms to copy and edit.  What happens when you draw the squares or rectangles backwards instead of forwards?  Can you create an algorithm to make this pattern in one go?	
	**Share:** Children share their pictures, patterns and algorithms and then continue with the **Activity Sheets** and respond to what they have shared or discovered after talking to their partner.	
	**Which Letter?** Children look at the algorithm on the **Lesson Presentation** and decide which letter is drawn by following the command. What do you need to do to draw 2 squares that are not touching?	

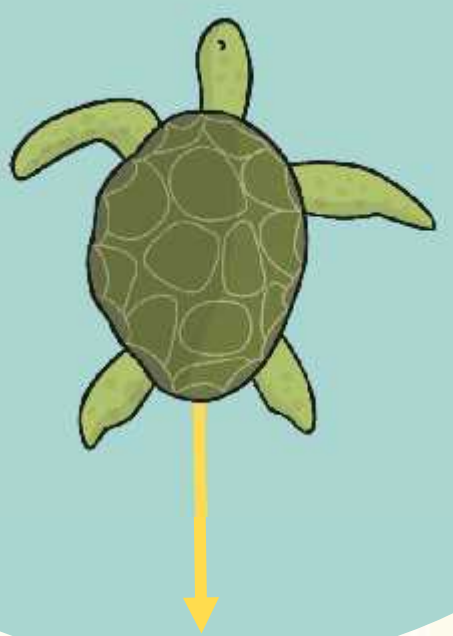
Task*it*

***Rotate*it:** Children make algorithms repeating shapes after rotating the turtle.

Computing

Programming Turtle Logo and Scratch

Backwards



Aim

- I can create and debug an algorithm using the move, rotate and repeat commands.

Success Criteria

- I can write commands in the correct order.
- I can write a variable value where required.
- I can correct any mistakes.
- I can use the commands fd, bk, lt, rt to move or rotate the turtle.
- I can use cs to clear the screen.
- I can use the repeat command.

What Can You Remember?

Can you use the basic Turtle Logo commands to write down the algorithm for drawing a square, rectangle or other rectilinear shape?

Can you remember the commands for moving forward, or rotating right and left?

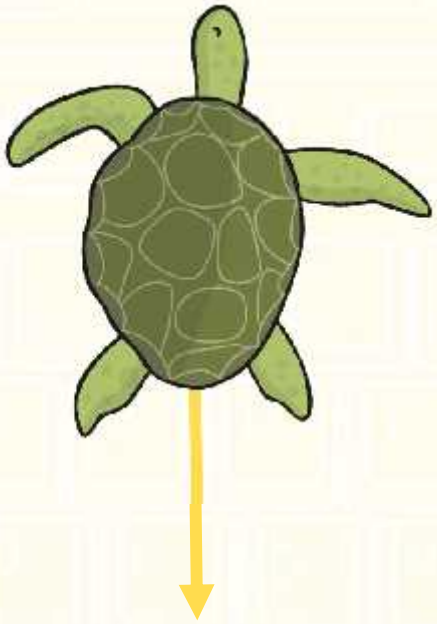
Shapes With Turtle Logo

Write an algorithm for:

- a square with sides of 120
- a rectangle of sides 50 and 80
- an L shape

Can You Go Backwards?

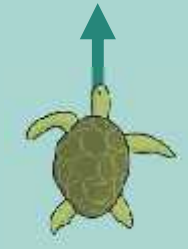
Can you give the turtle a command to go backwards?



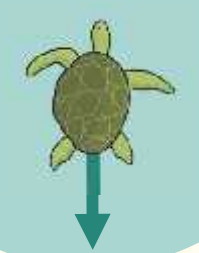
backward 100
Or
bk 100
(or any other number)

Turtle Logo Commands

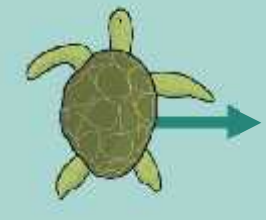
fd 100
move forward



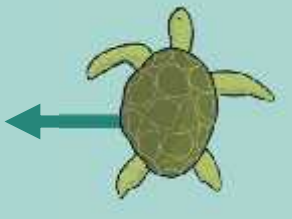
bk 100
move backward



rt 90
turn right



lt 90
turn left



cs
clear screen

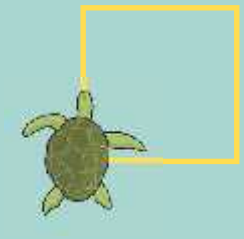
repeat

fd 4 rt 90

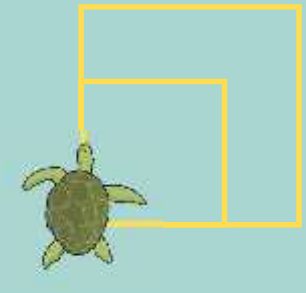
fd 4 rt 90

Repeating Squares

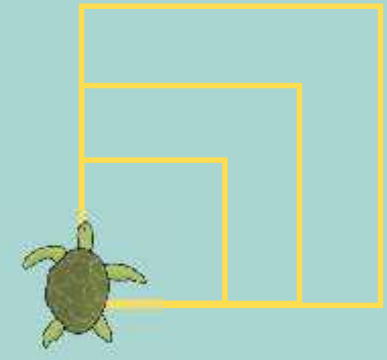
repeat 4[fd 100 rt 90]



repeat 4[fd 150 rt 90]



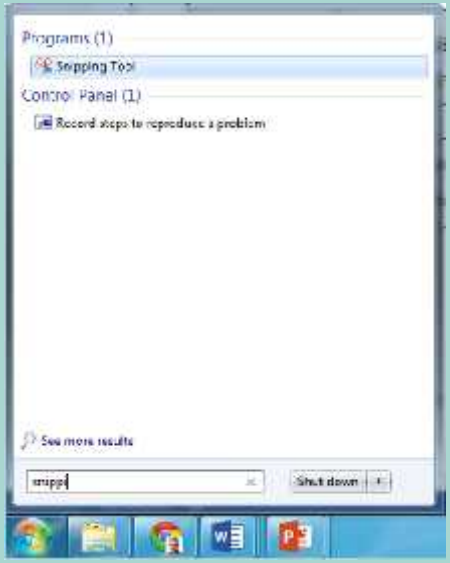
repeat 4[fd 200 rt 90]



Snipping Tool

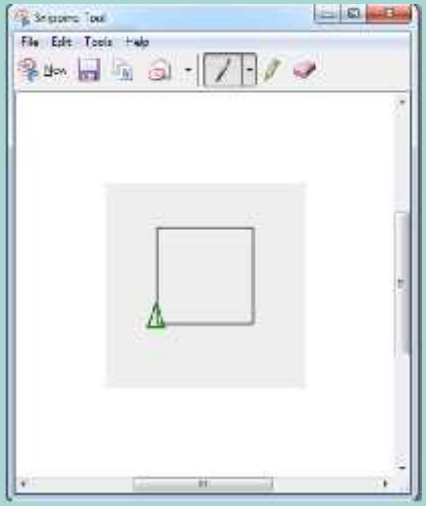
Use the snipping tool to save your pictures and patterns.

1. Go to the start menu



2. Type snip into the search bar and select the snipping tool.

3. Select the area to snip (or click new to do so).

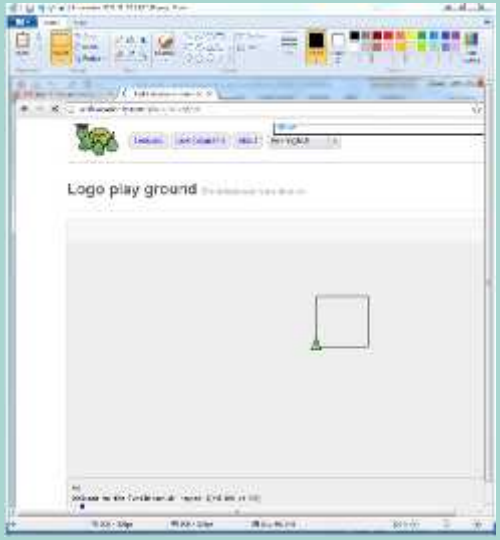


4. Save your snip.

Screenshot

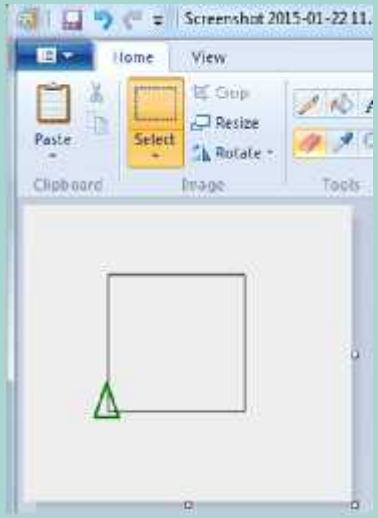
Alternatively you can take a screenshot to save your pictures and patterns.

1. Press “Print Screen” to copy the screen.



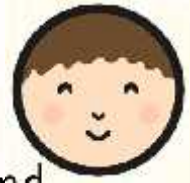
2. Paste the screenshot into Paint.

3. Select the picture with the select tool.

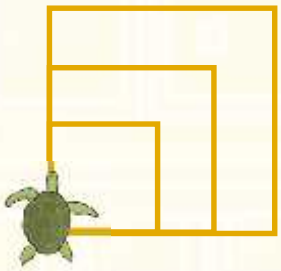


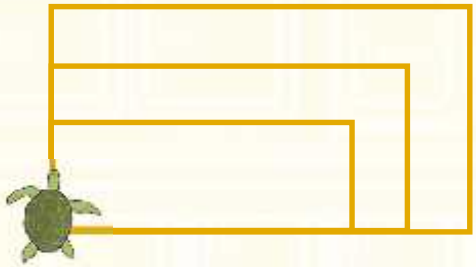
4. Crop and save the picture.

Growing Square and Rectangles



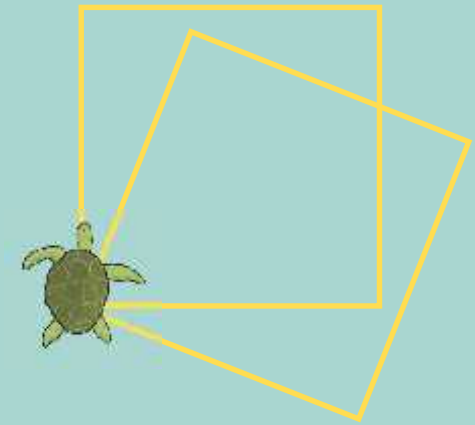
Draw some growing square and rectangles using the repeat command.





Can you use the backwards command to draw the squares?

Can you create an algorithm for this?



Share

Share your pictures, patterns and algorithms.

What difficulties did you have?

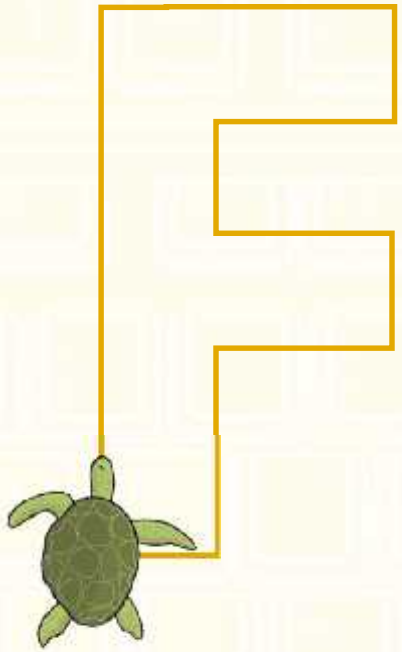
How did you overcome them?

What has your partner done well?

Which Letter?

Which letter is drawn by this algorithm?

fd 100 rt 90 fd 50 rt 90
fd 20 rt 90 fd 30 lt 90
fd 20 lt 90 fd 20 rt 90
fd 20 rt 90 fd 20 lt 90
fd 40 rt 90 fd 20 rt 90



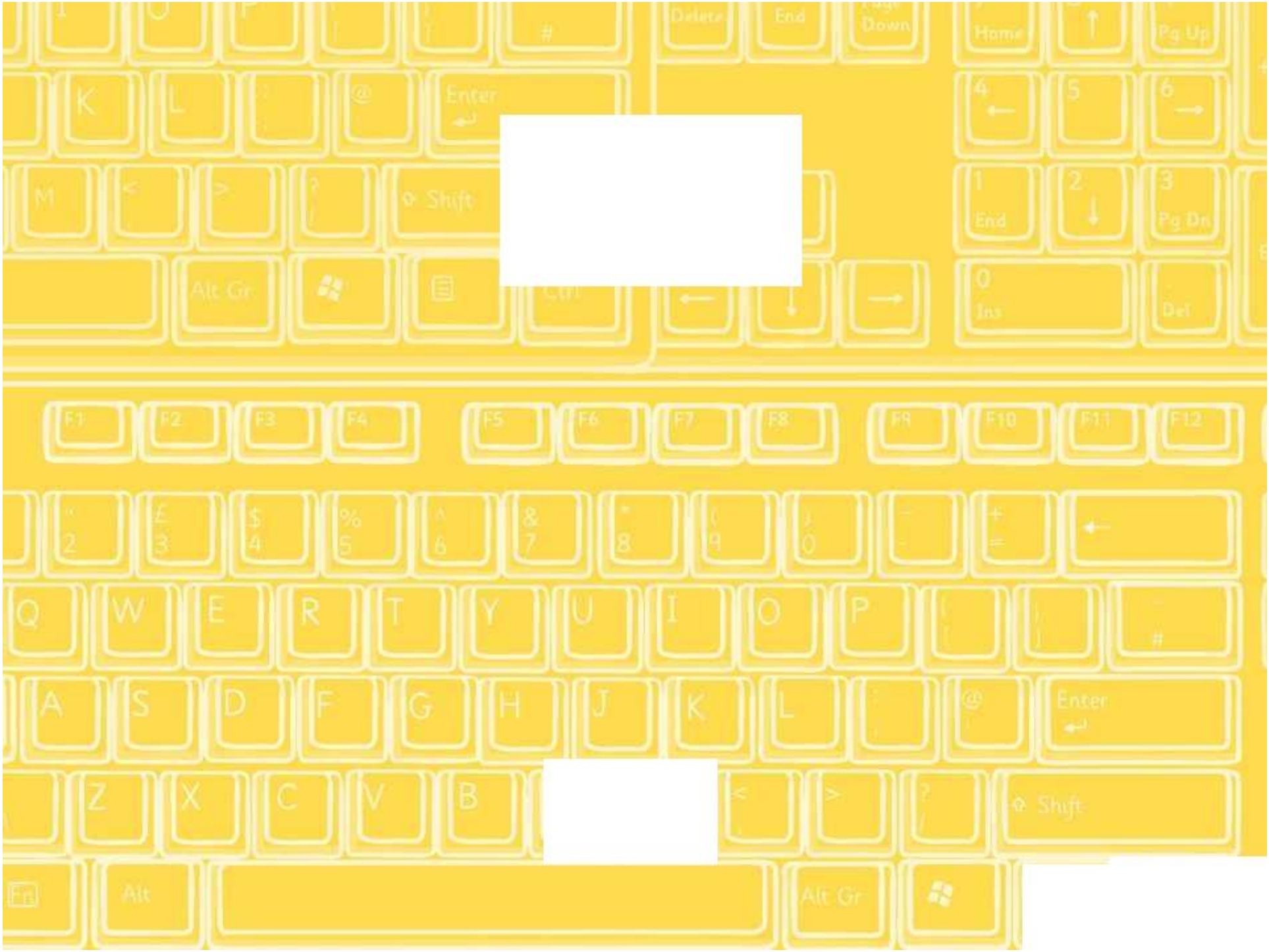
How would you draw 2 squares that are not touching?

Aim

- I can create and debug an algorithm using the move, rotate and repeat commands.

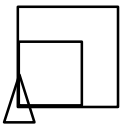
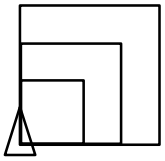
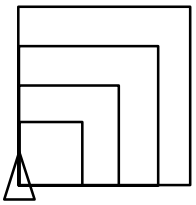
Success Criteria

- I can write commands in the correct order.
- I can write a variable value where required.
- I can correct any mistakes.
- I can use the commands fd, bk, lt, rt to move or rotate the turtle.
- I can use cs to clear the screen.
- I can use the repeat command.

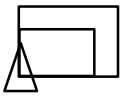
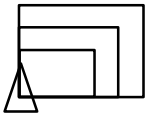
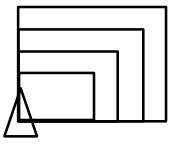


Backwards

Draw the following algorithms in Turtle Logo.
Remember to snip or take a screen shot of your work to save your pictures, patterns and algorithms.

1.	Draw a square with a size of 100 using the algorithm, repeat 4[fd 100 rt 90].	
2.	Draw a square of side 150 using the algorithm, repeat 4[fd 150 rt 90].	
3.	Draw a square of side 200.	
4.	Add a smaller square to complete thepattern.	

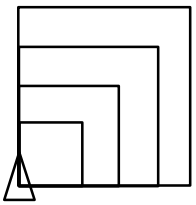
Create a set of rectangles that are all different sizes.

1.	Draw a rectangle with a side of 50 and 100 using the algorithm, repeat 2[fd 50 rt 90 fd 100 rt 90].	
2.	Draw a rectangle of side 75 and 150 using the algorithm, repeat 2[fd 75 rt 90 fd 150 rt90].	
3.	Draw a rectangle of side 100 and200.	
4.	Add a smaller rectangle to complete thepattern.	

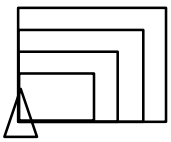
Now try drawing the square and rectangular patterns again but this time use the backwards command, repeat 4[bk 100 rt 90].

Backwards

Draw the following algorithms in Turtle Logo.
Remember to snip or take a screen shot of your work to save your pictures, patterns and algorithms.

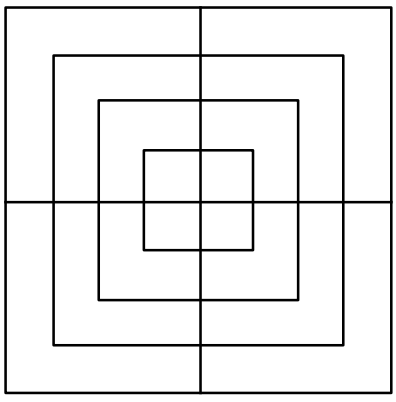
1.	Draw a square with a size of 100 using the algorithm, repeat 4[fd 100 rt 90].	You should end up with a pattern like this: 
2.	Draw a square of side 150.	
3.	Draw a square of side 200.	
4.	Add a smaller square to complete thepattern.	

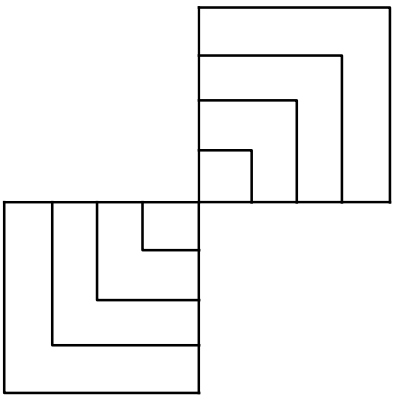
Create a set of rectangles that are all different sizes.

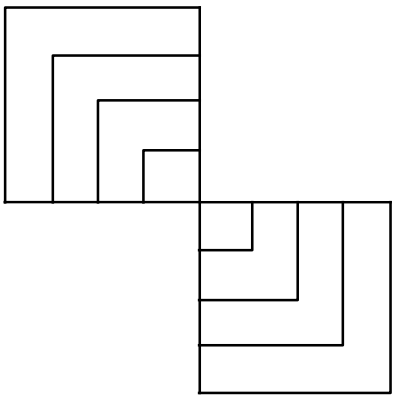
1.	Draw a rectangle with a side of 50 and 100 using the algorithm, repeat 2[fd 50 rt 90 fd 100 rt 90].	You should end up with a pattern like this: 
2.	Draw a rectangle of side 75and 150.	
3.	Draw a rectangle of side 100 and200.	
4.	Add a smaller rectangle to complete thepattern.	

Now try drawing the square and rectangular patterns again but this time use the backwards command, repeat 4[bk 100 rt 90].

Draw the patterns below using the backwards command. Don't forget to snip or take a screen shot of your pattern when you have completed it.

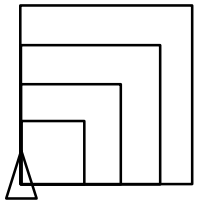




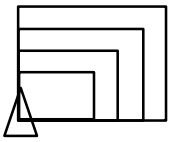


Backwards

Draw the following algorithms in Turtle Logo.
Remember to snip or take a screen shot of your work to save your pictures, patterns and algorithms.

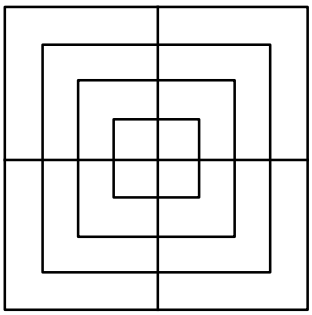
Use the repeat command to create a set of squares that are a different size. You should end up with a pattern like the one opposite.	

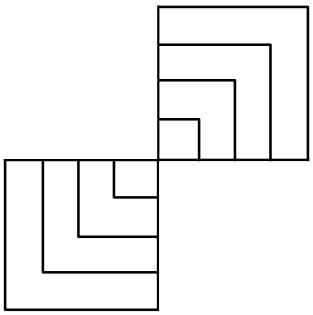
Create a set of rectangles that are all different sizes.

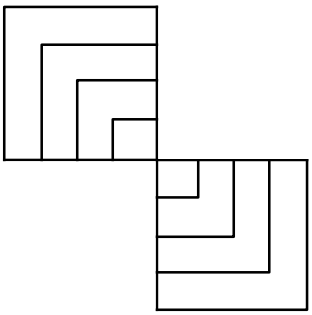
Use the repeat command to create a set of rectangles that are a different size. You should end up with a pattern like the one opposite.	

Now try drawing the square and rectangular patterns again but this time use the backwards command, repeat 4[bk 100 rt 90].

Draw the patterns below using the backwards command. Don't forget to snip or take a screen shot of your pattern when you have completed it.



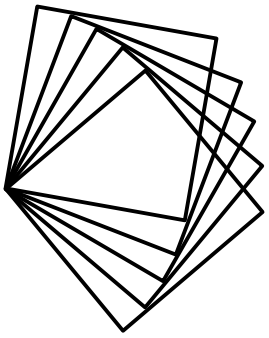


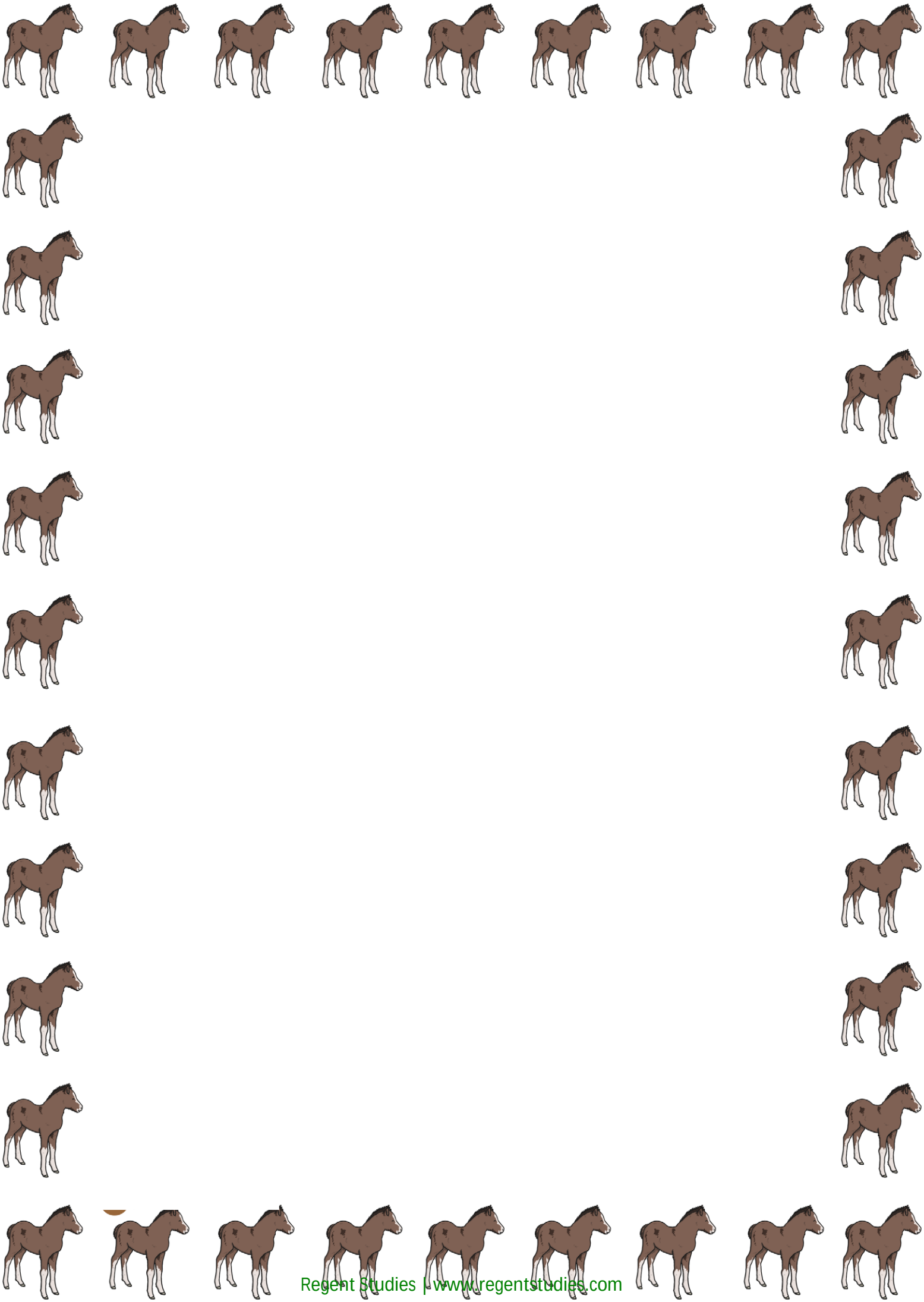


Challenge

Draw a square, rotate the turtle 10° and then repeat these commands to create a pattern. Now try g to include 2 repeat commands.

Draw the pattern again, this time with a rectangle.

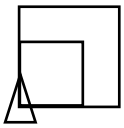
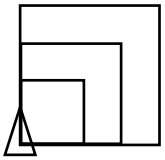
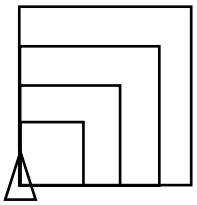




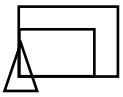
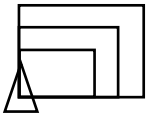
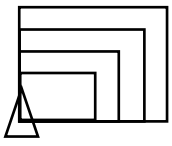
Backwards

Draw the following algorithms in Turtle Logo.
Remember to snip or take a screen shot of your work to save your pictures, patterns and algorithms.

Create a set of squares that are all different sizes.

1.	Draw a square with a size of 100 using the algorithm, repeat 4[fd 100 rt 90].	
2.	Draw a square of side 150 using the algorithm, repeat 4[fd 150 rt 90].	
3.	Draw a square of side 200.	
4.	Add a smaller square to complete the pattern.	

Create a set of rectangles that are all different sizes.

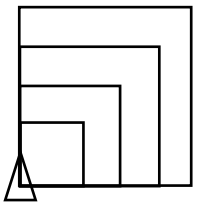
1.	Draw a rectangle with a side of 50 and 100 using the algorithm, repeat 2[fd 75 rt 90 fd 150 rt 90].	
2.	Draw a rectangle of side 75 and 150 using the algorithm, repeat 2[fd 150 rt 90 fd 75 rt 90].	
3.	Draw a rectangle of side 100 and 200.	
4.	Add a smaller rectangle to complete the pattern.	

Now try drawing the square and rectangular patterns again but this time use the backwards command, repeat 4[bk 100 rt 90].

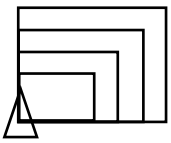
Backwards

Draw the following algorithms in Turtle Logo.
Remember to snip or take a screen shot of your work to save your pictures, patterns and algorithms.

Create a set of squares that are all different sizes.

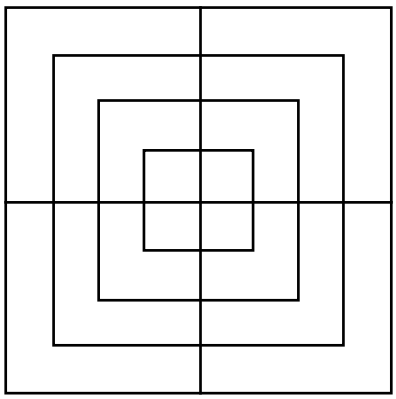
1.	Draw a square with a size of 100 using the algorithm, repeat 4[fd 100 rt 90].	You should end up with a pattern like this: 
2.	Draw a square of side 150.	
3.	Draw a square of side 200.	
4.	Add a smaller square to complete the pattern.	

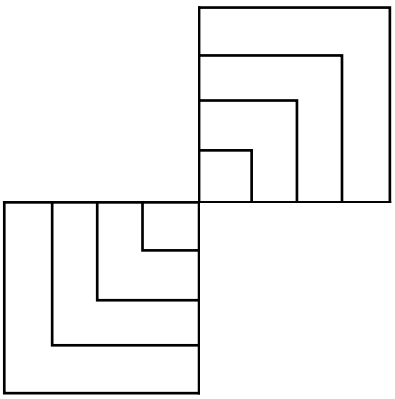
Create a set of rectangles that are all different sizes.

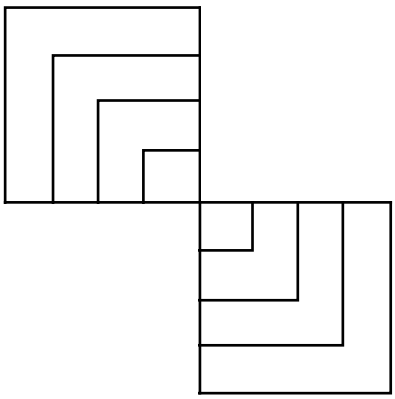
1.	Draw a rectangle with a side of 50 and 100 using the algorithm, repeat 2[fd 50 rt 90 fd 100 rt 90].	You should end up with a pattern like this: 
2.	Draw a rectangle of side 75 and 150.	
3.	Draw a rectangle of side 100 and 200.	
4.	Add a smaller rectangle to complete the pattern.	

Now try drawing the square and rectangular patterns again but this time use the backwards command, repeat 4[bk 100 rt 90].

Draw the patterns below using the backwards command. Don't forget to snip or take a screen shot of your pattern when you have completed it.



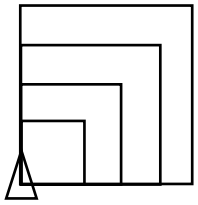




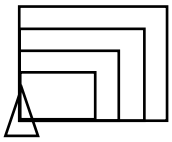
Backwards

Draw the following algorithms in Turtle Logo.
Remember to snip or take a screen shot of your work to save your pictures, patterns and algorithms.

Create a set of squares that are different sizes.

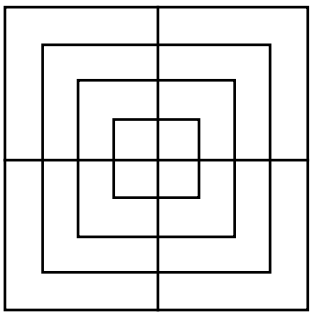
Use the repeat command to create a set of squares that are a different size. You should end up with a pattern like the one opposite.	

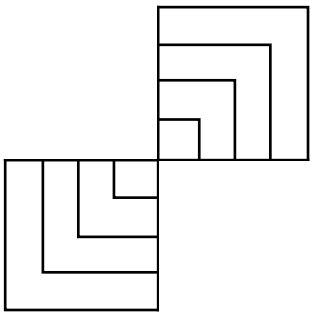
Create a set of rectangles that are all different sizes.

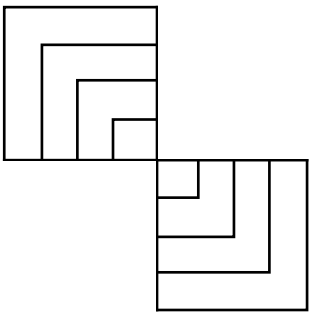
Use the repeat command to create a set of rectangles that are a different size. You should end up with a pattern like the one opposite.	

Now try drawing the square and rectangular patterns again but this time use the backwards command, repeat 4[bk 100 rt 90].

Draw the patterns below using the backwards command. Don't forget to snip or take a screen shot of your pattern when you have completed it.



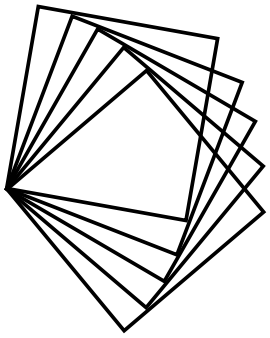




Challenge

Draw a square, rotate the turtle 10° and then repeat these commands to create a pattern. Now try writing an algorithm that will make this pattern in one go. Your algorithm will need to include 2 repeat commands.

Draw the pattern again, this time with a rectangle.



Avery Template: Name Badge Label, 8 per sheet | **Compatible Products**:15395, 25395, 42395, 45395, 48395, 5395, 8395, 88395, 85395.

Programming Turtle Logo and Scratch | Backwards

I can create and debug an algorithm using the move, rotate and repeat commands.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can use the commands fd, bk, lt, rt to move or rotate the turtle.		
I can use cs to clear the screen.		
I can use the repeat command.		

Programming Turtle Logo and Scratch | Backwards

I can create and debug an algorithm using the move, rotate and repeat commands.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can use the commands fd, bk, lt, rt to move or rotate the turtle.		
I can use cs to clear the screen.		
I can use the repeat command.		

Programming Turtle Logo and Scratch | Backwards

I can create and debug an algorithm using the move, rotate and repeat commands.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can use the commands fd, bk, lt, rt to move or rotate the turtle.		
I can use cs to clear the screen.		
I can use the repeat command.		

Programming Turtle Logo and Scratch | Backwards

I can create and debug an algorithm using the move, rotate and repeat commands.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can use the commands fd, bk, lt, rt to move or rotate the turtle.		
I can use cs to clear the screen.		
I can use the repeat command.		

Programming Turtle Logo and Scratch | Backwards

I can create and debug an algorithm using the move, rotate and repeat commands.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can use the commands fd, bk, lt, rt to move or rotate the turtle.		
I can use cs to clear the screen.		
I can use the repeat command.		

Programming Turtle Logo and Scratch | Backwards

I can create and debug an algorithm using the move, rotate and repeat commands.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can use the commands fd, bk, lt, rt to move or rotate the turtle.		
I can use cs to clear the screen.		
I can use the repeat command.		

Programming Turtle Logo and Scratch | Backwards

I can create and debug an algorithm using the move, rotate and repeat commands.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can use the commands fd, bk, lt, rt to move or rotate the turtle.		
I can use cs to clear the screen.		
I can use the repeat command.		

Programming Turtle Logo and Scratch | Backwards

I can create and debug an algorithm using the move, rotate and repeat commands.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can use the commands fd, bk, lt, rt to move or rotate the turtle.		
I can use cs to clear the screen.		
I can use the repeat command.		

Programming Turtle Logo and Scratch: Pen Up and Pen down

Aim:	**Success Criteria:**	**Resources:**
Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts. Use sequence, selection, and repetition in programs; work with variables and various forms of input and output. Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs. This unit continues the learning from the Year 2 Turtle Logo units and links well to shape and direction in Maths. I can create and debug algorithms using pen up and pen down.	I can write commands in the correct order. I can write a variable value where required. I can correct any mistakes. I can use the pen up and pen down command.	**Lesson Pack** Desktop computer /laptop Turtle Logo application (installed or online) Whiteboards and pens or books, pens and pencils for recording.
	Key/New Words: Algorithm, instructions, commands, forward (fd), left (lt), right (rt), move, turn, clear screen (cs), variable, pen up, pen down.	**Preparation:** None needed

Prior Learning: Children will have created an algorithm using the move, rotate and repeat commands in lesson 1.

Learning Sequence

	**How Many?** Give the children 2 minutes to draw as many squares as they can. They must start in the same corner and grow. What time saving commands can you use? *(Repeat and up arrow to copy previous algorithms).*	
	**Lifting the Pen:** Introduce children to penup and pendown commands. Ask the children what algorithm would make a dotted line? Remind the children to snip or screenshot their work as they did last lesson.	
	**Pen Up and Pen Down:** Children use the differentiated **Activity Sheets** and use a dotted line to draw concentric squares and rectangles.  Children are given step by step support to write the algorithms.  Children are given prompts to help them write the algorithms.  Children work independently.	
	**Share:** Children share the pictures and patterns they have created and the algorithms they have used to create the patterns and letters.	
	**Which Algorithm Will Draw This Shape?** Which algorithm will draw the shape on the slide? Children decide as a class which answer to select, ask them to explain their choice. *(If the incorrect answer is chosen you can click on the shape to return to the question to try again.)*	

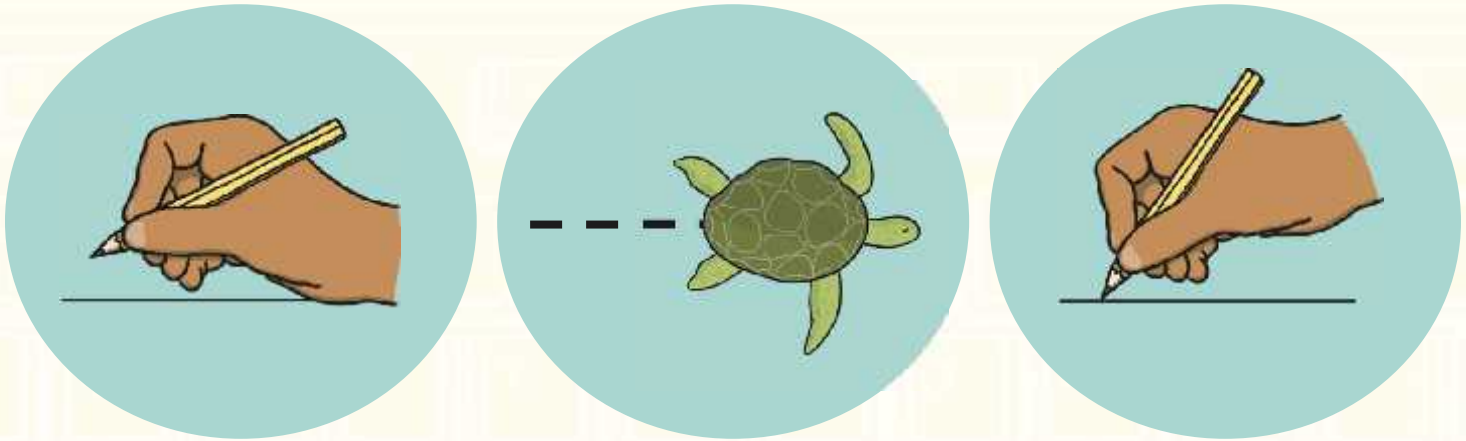
Task*it*

***Pattern*it:** Children make algorithms for repeating shapes with spaces between.

Computing

Programming Turtle Logo and Scratch

Pen Up and Pen Down



Aim

- I can create and debug algorithms using pen up and pen down.

Success Criteria

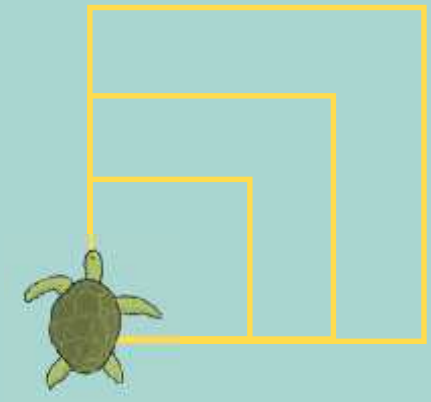
- I can write commands in the correct order.
- I can write a variable value where required.
- I can correct any mistakes.
- I can use the pen up and pen down commands.

How Many?

You have 2 minutes to draw as many squares as you can.
All of the squares need to start at the same corner.

Which commands will help?

- Repeat
- Up arrow to use previous algorithm



Click on the timer to start

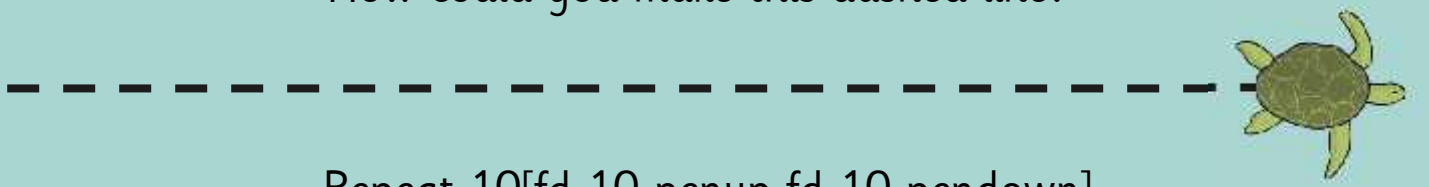
Lifting the Pen

Penup will lift the pen so the turtle will not draw as it moves.

Pendown places the pen back down so that it draws again.

These commands allow you to leave a space between objects.

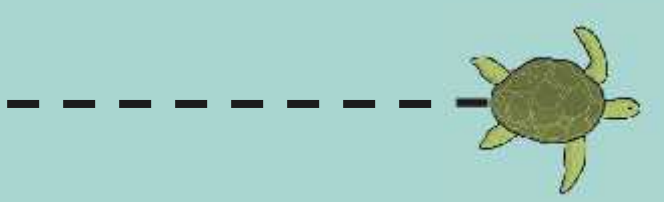
How could you make this dashed line?



Repeat 10[fd 10 penup fd 10 pendown]

Pen Up and Pen Down

1. Draw a dashed line using penup and pendown.



2. Draw a set of concentric circles and squares.

Remember to snip or take a screenshot of your pictures and algorithms.

Can you create algorithms for the capital letters T, O, P and A?

Can you create an algorithm for a 3 letter word?

Share

Share your pictures, patterns and algorithms.

What difficulties did you have?

How did you overcome them?

What has your partner done well?

Which Algorithm Will Draw This Shape?

A

```
repeat 4[fd 50 rt 90]
fd 100
repeat 4[fd 50 rt 90]
```

B

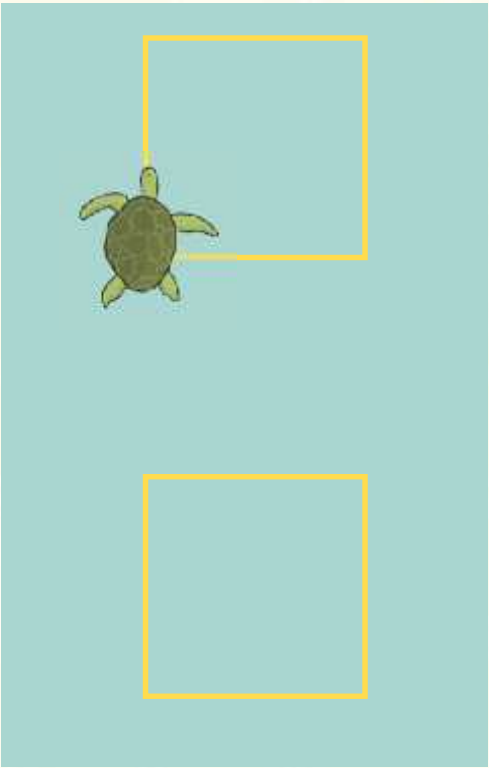
```
repeat 4[fd 50 rt 90]
penup fd 50
pendown
repeat 4[fd 50 rt 90]
```

C

```
repeat 4[fd 50 rt 90]
penup fd 150
pendown
repeat 4[fd 50 rt 90]
```

D

```
repeat 4[fd 50 rt 90]
penup fd 100
pendown
repeat 4[fd 50 rt 90]
```



Click on the algorithm that you think is correct

Which Algorithm Will Draw This Shape?

A

repeat 4[fd 50 rt 90]
fd 100
repeat 4[fd 50 rt 90]

B

repeat 4[fd 50 rt 90]
penup fd 50
pendown
repeat 4[fd 50 rt 90]

Incorrect:

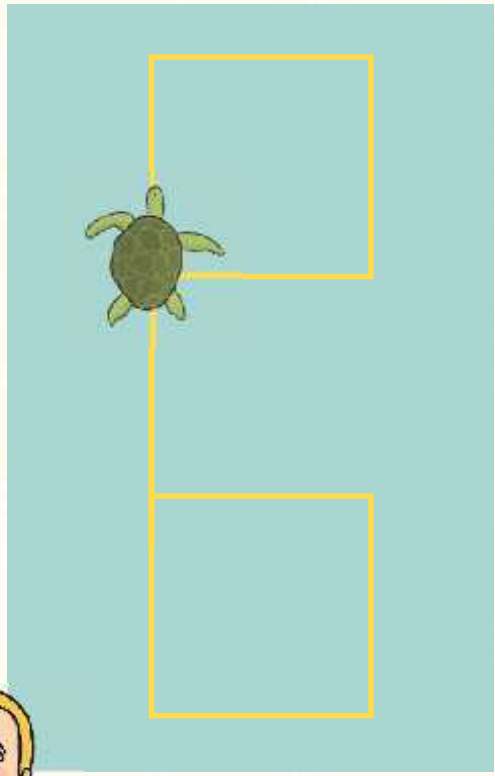
Pen not lifted so there would be a line joining squares.

C

repeat 4[fd 50 rt 90]
penup fd 150
pendown
repeat 4[fd 50 rt 90]

D

repeat 4[fd 50 rt 90]
penup fd 100
pendown
repeat 4[fd 50 rt 90]



Click on the shape to try again.

Which Algorithm Will Draw This Shape?

A

repeat 4[fd 50 rt 90]
fd 100
repeat 4[fd 50 rt 90]

B

repeat 4[fd 50 rt 90]
penup fd 50
pendown
repeat 4[fd 50 rt 90]

Incorrect:

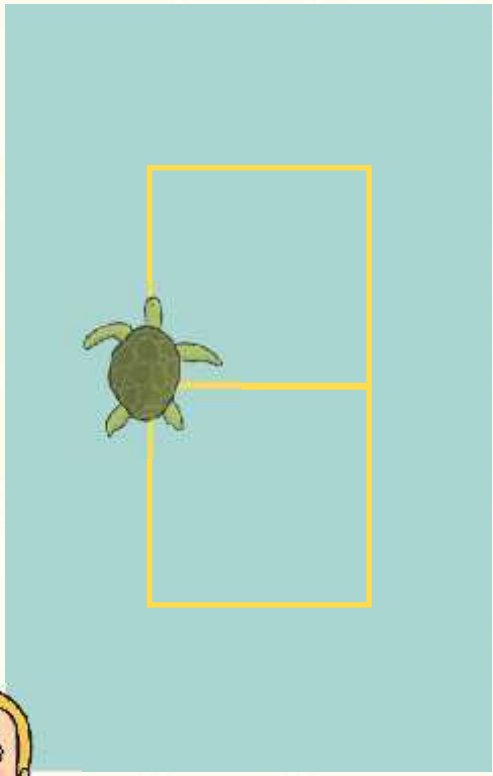
Squares would be touching, one above the other.

C

repeat 4[fd 50 rt 90]
penup fd 150
pendown
repeat 4[fd 50 rt 90]

D

repeat 4[fd 50 rt 90]
penup fd 100
pendown
repeat 4[fd 50 rt 90]



Click on the shape to try again.

Which Algorithm Will Draw This Shape?

A

```
repeat 4[fd 50 rt 90]
fd 100
repeat 4[fd 50 rt 90]
```

B

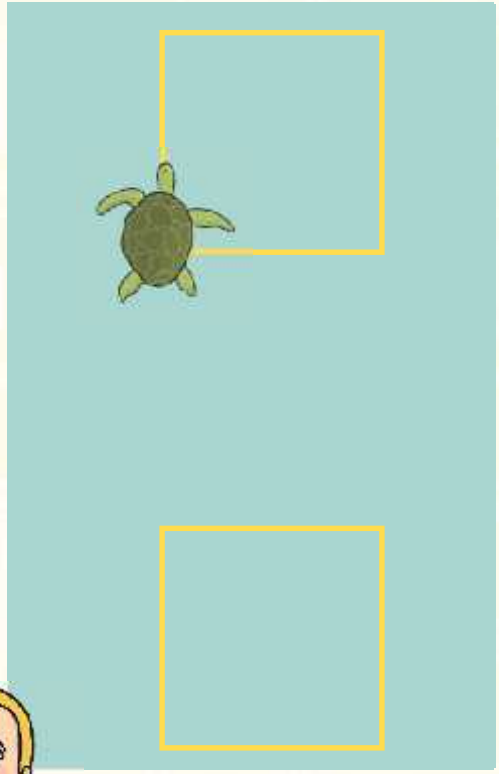
```
repeat 4[fd 50 rt 90]
penup fd 50
pendown
repeat 4[fd 50 rt 90]
```

C

```
repeat 4[fd 50 rt 90]
penup fd 150
pendown
repeat 4[fd 50 rt 90]
```

D

```
repeat 4[fd 50 rt 90]
penup fd 100
pendown
repeat 4[fd 50 rt 90]
```



Incorrect:

Space between squares too big (100).

Click on the shape to try again.

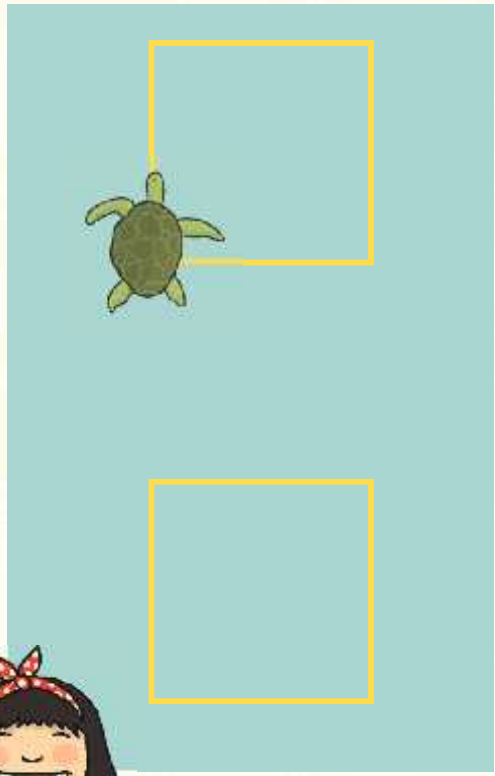
Which Algorithm Will Draw This Shape?

A

```
repeat 4[fd 50 rt 90]
fd 100
repeat 4[fd 50 rt 90]
```

B

```
repeat 4[fd 50 rt 90]
penup fd 50
pendown
repeat 4[fd 50 rt 90]
```



C

```
repeat 4[fd 50 rt 90]
penup fd 150
pendown
repeat 4[fd 50 rt 90]
```

D

```
repeat 4[fd 50 rt 90]
penup fd 100
pendown
repeat 4[fd 50 rt 90]
```

★ Correct ★

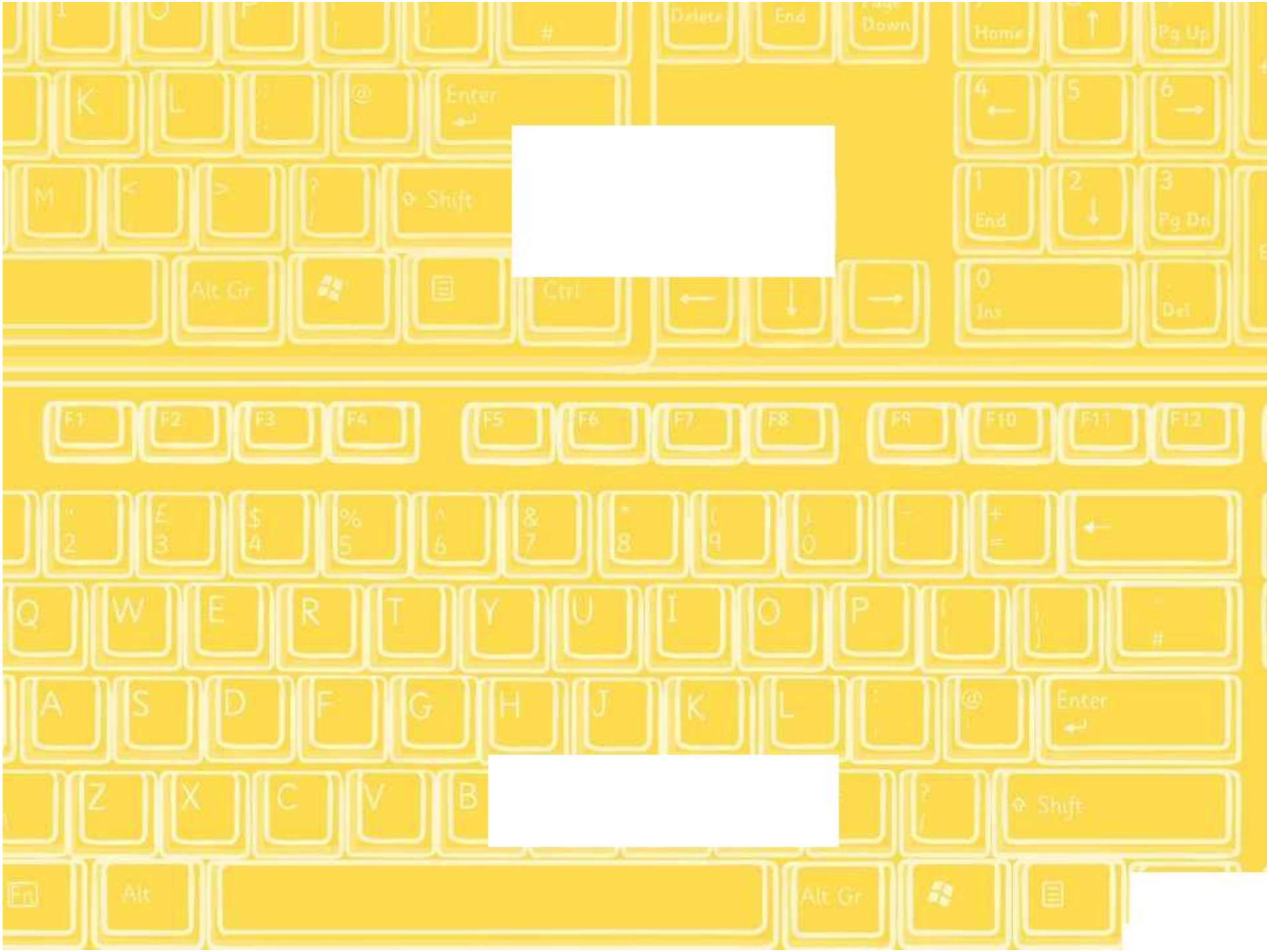
★ Well done! ★

Aim

- I can create and debug algorithms using pen up and pen down.

Success Criteria

- I can write commands in the correct order.
- I can write a variable value where required.
- I can correct any mistakes.
- I can use the pen up and pen down commands.



Pen Up and Pen Down

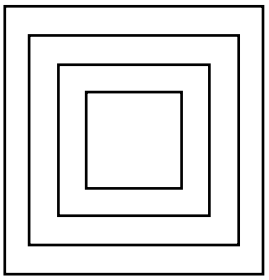
Draw the following algorithms in Turtle Logo.
Remember to snip or take a screen shot of your work to save your pictures, patterns and algorithms.

Concentric Squares

Draw a set of concentric squares (squares with the same centre).

1. Draw a square of side 50 using the algorithm, repeat 4[fd 50 rt 90].
2. Use the penup command.
3. Move to a new position using the algorithm, bk 25 lt 90 fd 25 rt 90.
4. Use the pendown command.
5. Draw a square of side 100.
6. Move the turtle again (remember penup and pendown).
7. Draw a square of side 150.
8. Move the turtle again (remember penup and pendown).
9. Draw a square of side 200.

You should end up with a pattern like this.

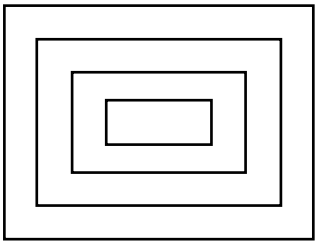


Concentric Rectangles

Draw a set of concentric rectangles (rectangles with the same centre).

1. Draw a rectangle of side 25 by 50 using the algorithm, repeat 2[fd 25 rt 90 fd 50 rt 90].
2. Use the penup command.
3. Move to a new position using the algorithm, bk 25 lt 90 fd 25 rt 90.
4. Use the pendown command.
5. Draw a rectangle of side 75 by 100 using the algorithm, repeat 2[fd 75 rt 90 fd 100 rt 90].
6. Move the turtle again (remember penup and pendown).
7. Draw a rectangle of side 125 by 150.
8. Move the turtle again (remember penup and pendown).
9. Draw a rectangle of side 175 by 200.

You should end up with a pattern like this.



Pen Up and Pen Down

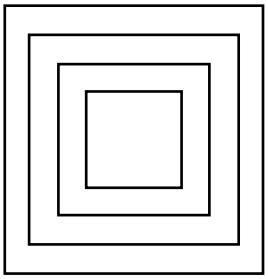
Draw the following algorithms in Turtle Logo.
Remember to snip or take a screen shot of your work to save your pictures, patterns and algorithms.

Concentric Squares

Draw a set of concentric squares (squares with the same centre).

1. Draw a square of side 50.
2. Lift the pen.
3. Move the turtle 25 to the left and 25 down.
4. Put the pen down again.
5. Draw a square of side 100.
6. Move the turtle again (remember penup and pendown).
7. Repeat this for squares of side 150 and 200.

You should end up with a pattern like this.

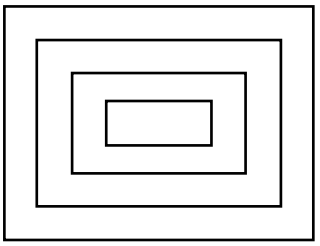


Concentric Rectangles

Draw a set of concentric rectangles (rectangles with the same centre).

1. Draw a rectangle of side 25 by 50 using the algorithm, repeat 2[fd 25 rt 90 fd 50 rt 90].
2. Lift the pen.
3. Move the turtle 25 to the left and 25 down.
4. Put the pen down again.
5. Draw a rectangle of side 75 by 100.
6. Move the turtle again.
7. Draw a rectangle of side 125 by 150.
8. Move the turtle again.
9. Draw a rectangle of side 175 by 200.

You should end up with a pattern like this.

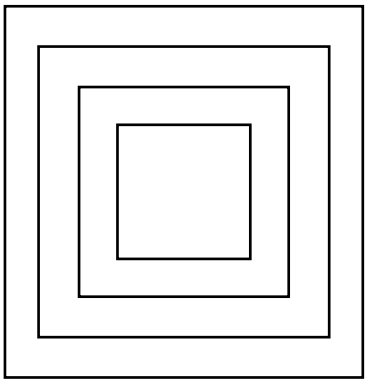


Now create algorithms for the letters T, O, P and A.

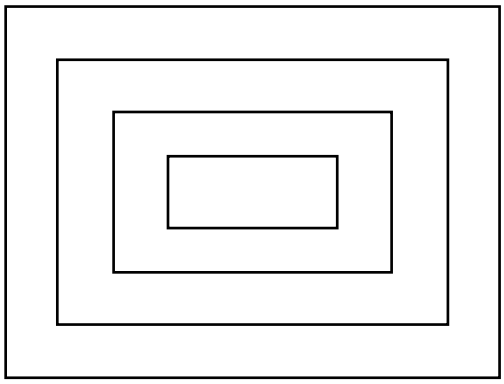
Pen Up and Pen Down

Draw the following algorithms in Turtle Logo.
Remember to snip or take a screen shot of your work to save your pictures, patterns and algorithms.

1. Draw a set of concentric squares (squares with the same centre). You should end up with a pattern like this.



2. Draw a set of concentric rectangles (rectangles with the same centre). You should end up with a pattern like this.



3. Now create algorithms for the letters T, O, P and A.

4. Create an algorithm to write a 3 letter word, you can use capital letters or lowercase.

Programming Turtle Logo and Scratch | Pen Up and Pen Down

I can create and debug algorithms using pen up and pen down.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can use the pen up and pen down command.		

Programming Turtle Logo and Scratch | Pen Up and Pen Down

I can create and debug algorithms using pen up and pen down.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can use the pen up and pen down command.		

Programming Turtle Logo and Scratch | Pen Up and Pen Down

I can create and debug algorithms using pen up and pen down.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can use the pen up and pen down command.		

Programming Turtle Logo and Scratch | Pen Up and Pen Down

I can create and debug algorithms using pen up and pen down.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can use the pen up and pen down command.		

Programming Turtle Logo and Scratch | Pen Up and Pen Down

I can create and debug algorithms using pen up and pen down.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can use the pen up and pen down command.		

Programming Turtle Logo and Scratch | Pen Up and Pen Down

I can create and debug algorithms using pen up and pen down.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can use the pen up and pen down command.		

Programming Turtle Logo and Scratch | Pen Up and Pen Down

I can create and debug algorithms using pen up and pen down.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can use the pen up and pen down command.		

Programming Turtle Logo and Scratch | Pen Up and Pen Down

I can create and debug algorithms using pen up and pen down.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can use the pen up and pen down command.		

Programming Turtle Logo and Scratch: Regular Polygons

Aim:	**Success Criteria:**	**Resources:**
Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts. Use sequence, selection, and repetition in programs; work with variables and various forms of input and output. Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs. This unit continues the learning from the Year 2 Turtle Logo units and links well to shape and direction in Maths. I can create and debug algorithms that draw regular polygons.	I can write commands in the correct order. I can write a variable value where required. I can correct any mistakes. I can rotate the turtle angles other than 90˚. I can use calculations as a variable.	**Lesson Pack** Desktop computer /laptop Turtle Logo application (installed or online) Whiteboards and pens or books, pens and pencils for recording.
	Key/New Words: Algorithm, instructions, commands, forward (fd), left (lt), right (rt), move, turn, clear screen (cs), variable, pen up, pen down, calculation.	**Preparation:** None needed

Prior Learning: Children will have created an algorithm using the pen up and pen down commands in lesson 2.

Learning Sequence

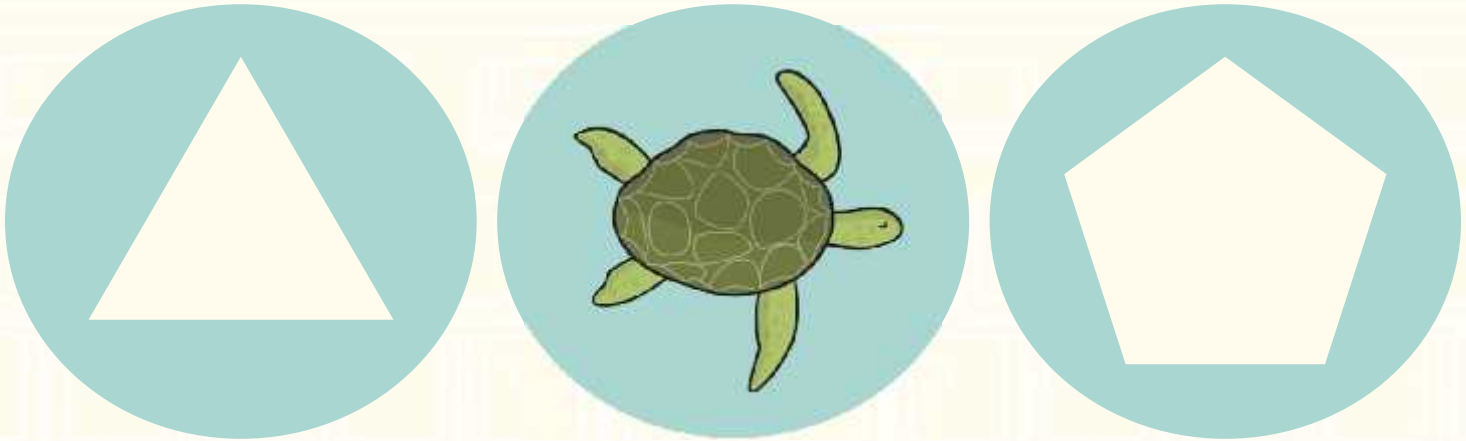
	**Write the Algorithm:** Ask the children to write algorithms for different shapes and test each one. • Write an algorithm for a square of side 120 and a square of 60 inside. • Write an algorithm for 4 rectangles of sides 30 and 50 with a space of 20 between each. • Write an algorithm for a rectilinear number 8.	
	**Turning Angles Other Than 90˚/What Angle to Turn?** Introduce the different rotating angles and demonstrate each one. Explain how to work out what angle you would use in your command and work through the example on the **Lesson Presentation.**	
	**Drawing Different Polygons:** Children use the differentiated **Activity Sheets** to draw different polygons, then either take a snip or screenshot to save their patterns and algorithms. Do you come across any problems with any of the shapes? What happens as you draw regular polygons with more and more sides? What is the most number of sides you can draw a regular polygon with?  Children are given step by step support to write the algorithms.  Children are given prompts to help them write the algorithms.  Children create algorithms to draw regular polygons and answer the questions on the **Activity Sheet.**	
	**Let Turtle Logo Work It Out!** When sufficient children have drawn a number of polygons and are confident with the algorithm, show how to let Turtle Logo calculate the required angle of turn. *(rt 360/6 is the same as rt 60 because 360/6=60.)*	
	**Write Your Own:** Children continue the activities above, but may use the option to have Turtle Logo calculate the turn.	
	**17 Sides?** Can you draw a regular polygon with 17 sides? Ask children to share their answers and explain how they worked out the algorithm.	
	**Which Shape Will Be Drawn?** Ask children to look at the algorithm, fd 120 rt 120 fd 120 lt 60 fd 120 rt 120 repeat 3[fd 120 rt 60]. Which shape do you think this algorithm will draw? Give children time to think about which solution is correct and then as a class decide on which answer to select.	

Task*it*
***Polygon*it:** Children make algorithms for repeating shapes with spaces between.

Computing

Programming Turtle Logo and Scratch

Regular Polygons



Aim

- I can create and debug algorithms that draw regular polygons.

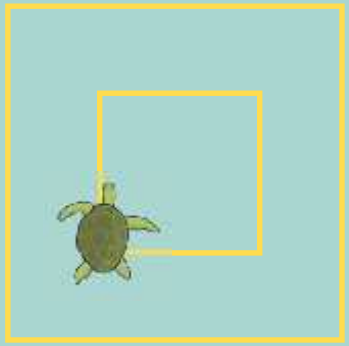
Success Criteria

- I can write commands in the correct order.
- I can write a variable value where required.
- I can correct any mistakes.
- I can rotate the turtle angles other than 90°.
- I can use calculations as a variable.

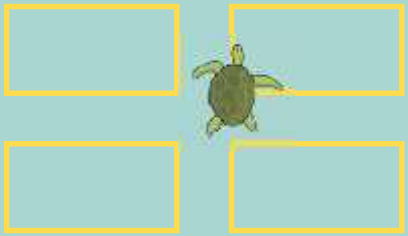
Write the Algorithm

Write algorithms for these different shapes and test them:

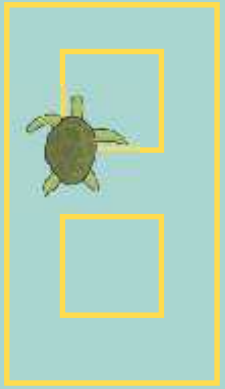
A square of 120 sides with a square of 60 sides inside.



4 rectangles of 30 and 50 sides with a space of 20 between each.



A rectilinear number 8.

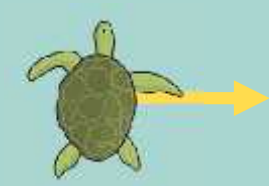


Turning Angles Other Than 90°

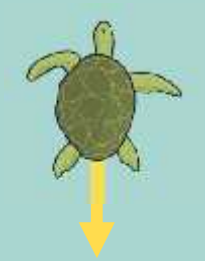
The turtle can be commanded to turn any angle. The angle of the turn is measured in degrees.

rt will turn the turtle an angle of 45° to the right.

90° is a quarter turn.



180° is a half turn.



360° is a full turn.

What Angle To Turn?

To work out the angle to turn for any polygon you need to divide 360 by the number of sides.

In a square there are 4 turns, so the turn is $360 \div 4 = 90$.

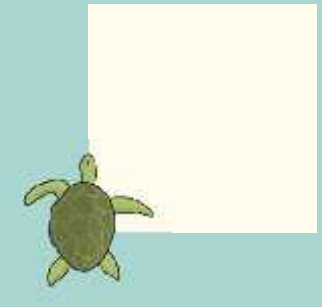
So, for a hexagon, it is $360 \div 6 = 60$.

Drawing Different Polygons

Using a Turtle Logo programme on a computer or tablet, draw different regular polygons.

How will you work out what angle you need to turn?

Remember a whole turn is 360°, and with a square you turned a quarter turn, which is 90°.



Let Turtle Logo Work it Out!

You can write an algorithm that includes a calculation.

So for a square you can make the turn 360/4 (representing 360 ÷ 4)

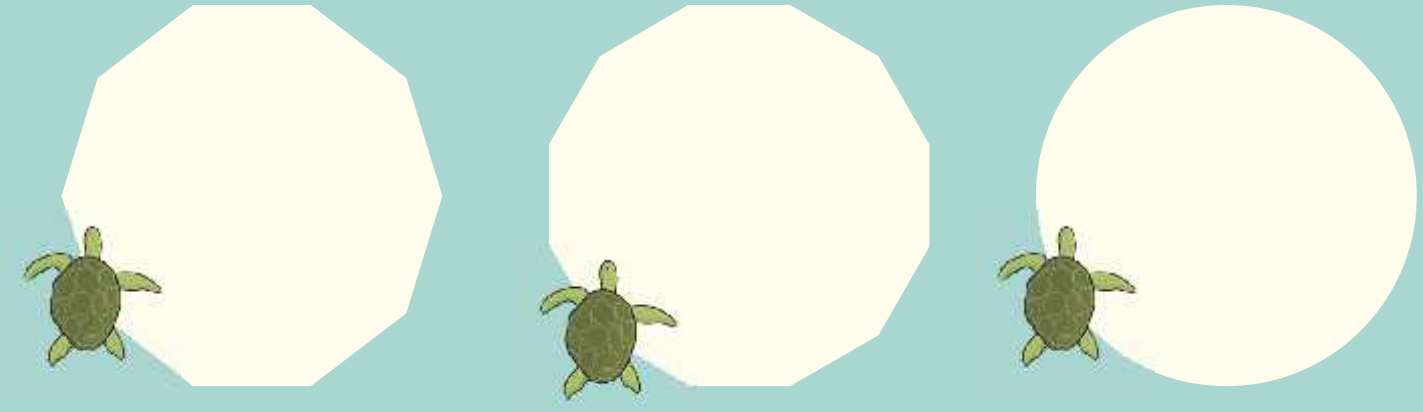
The command will be repeated 4[fd 100 rt 360/4]

So, for a heptagon, it is repeat 7[fd 80 rt 360/7]

What happens when you draw a polygon with many sides?
Try writing an algorithm for a shape with 360 side. What happens?

Write Your Own

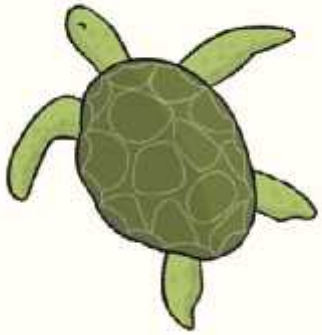
Write your own algorithms that include calculations to draw regular polygons that have a different numbers of sides.



17 Sides?

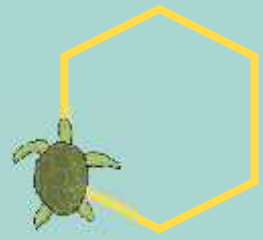
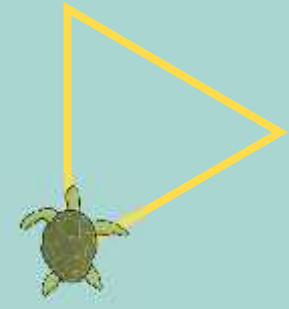
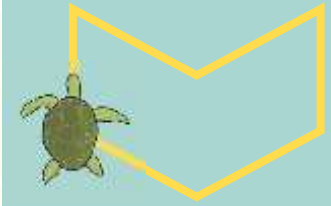
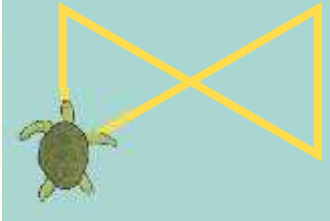
Can you write
the algorithm for a
regular 17 sided shape?

repeat 17[fd 40 rt 360/17]



Which Shape Will Be Drawn?

fd 120 rt 120 fd 120 lt 60 fd 120 rt 120
repeat 3[fd 120 rt 60]

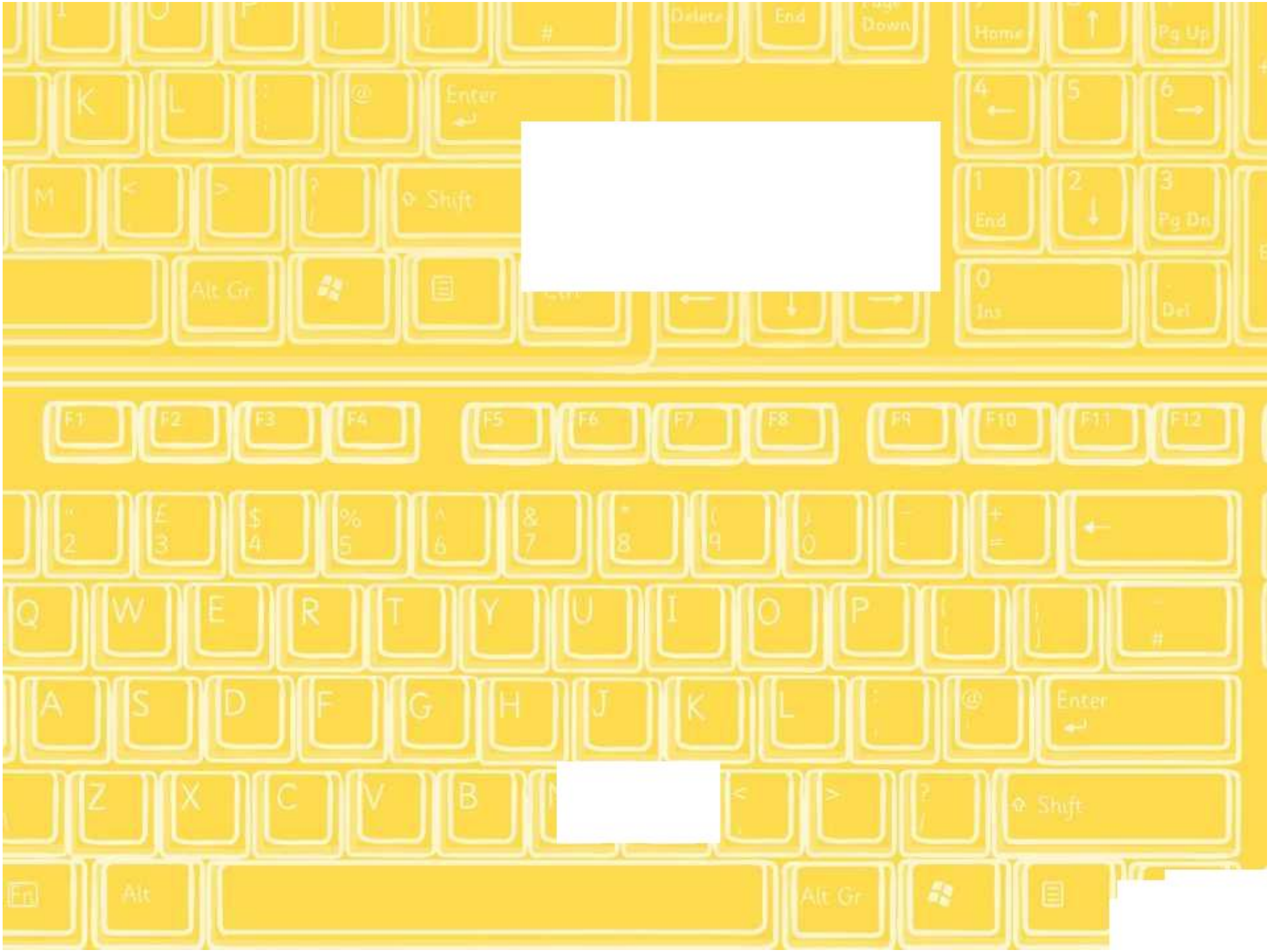
A	B	C	D
			
Incorrect: All turns would be rt 60 or lt 60.	**Incorrect:** All turns would be rt 60 or lt 60.	Click on the shape that you think is correct Correct Well done!	**Incorrect:** Second line would repeat the first line.

Aim

- I can create and debug algorithms that draw regular polygons.

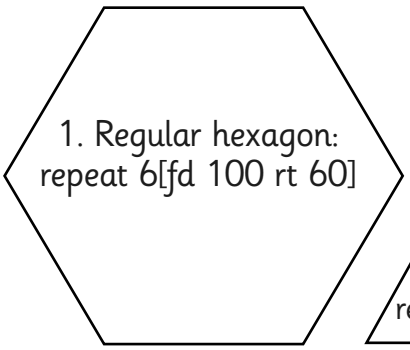
Success Criteria

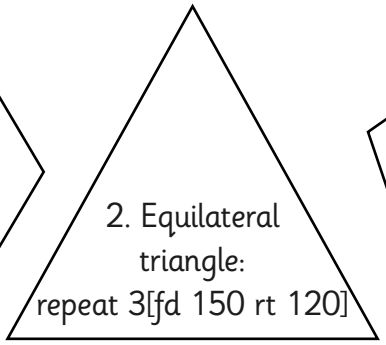
- I can write commands in the correct order.
- I can write a variable value where required.
- I can correct any mistakes.
- I can rotate the turtle angles other than 90°.
- I can use calculations as a variable.

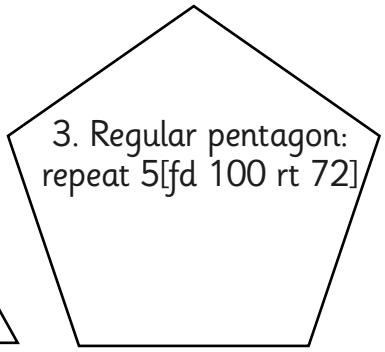


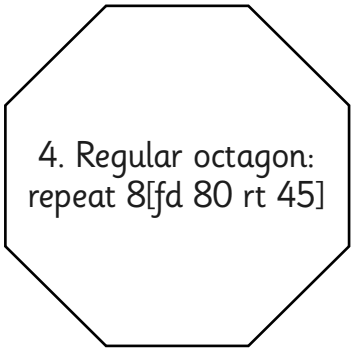
Regular Polygons

Draw the following algorithms in Turtle Logo.
Remember to snip or take a screen shot of your work to save your pictures, patterns and algorithms.



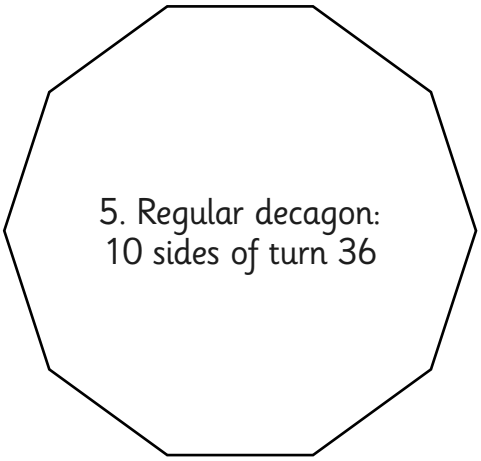


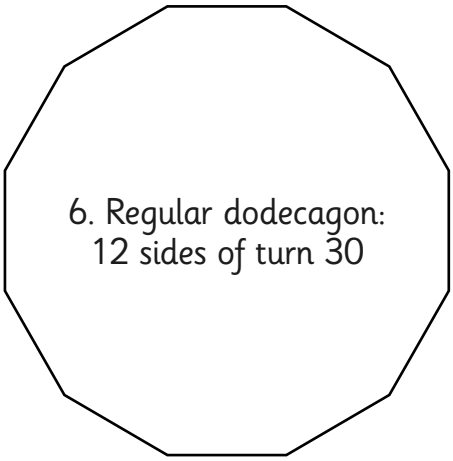


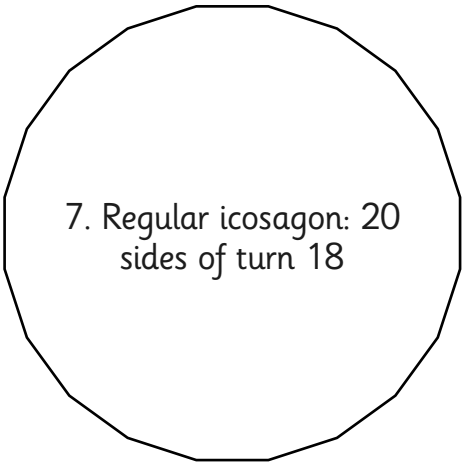


Now try drawing the following algorithms.

Don't forget to change the number of amount of times the algorithm is reapeated and the amount you want the turtle to turn. Take care not to make the sides too long!





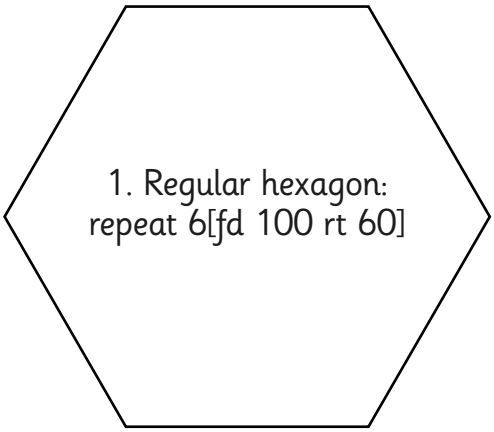


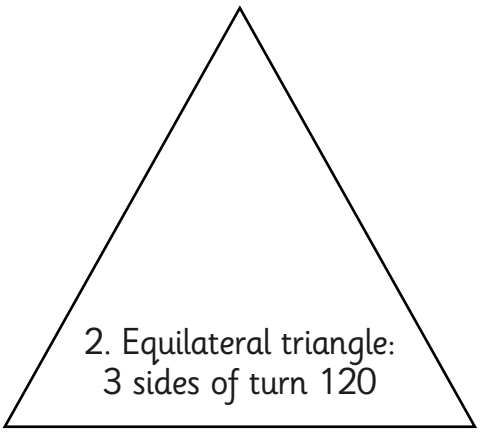
Now try drawing some other regular polygons.

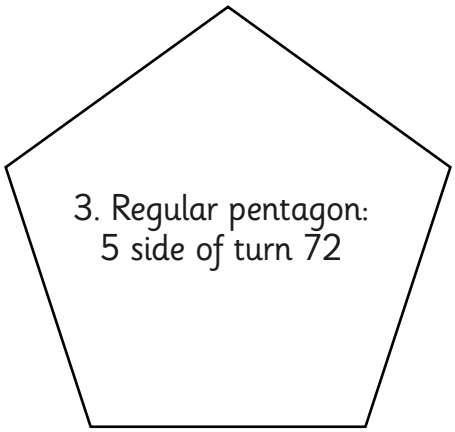
Have you come across any problems with any of the shapes?

Regular Polygons

Draw the following algorithms in Turtle Logo.
Remember to snip or take a screen shot of your work to save your pictures, patterns and algorithms.

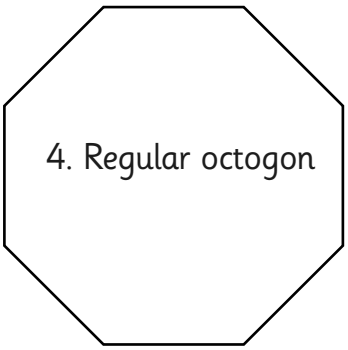


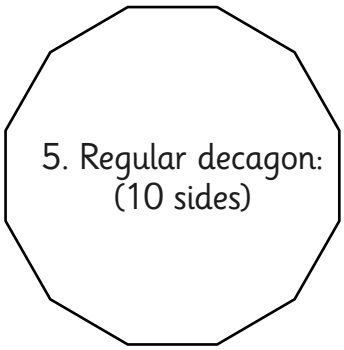


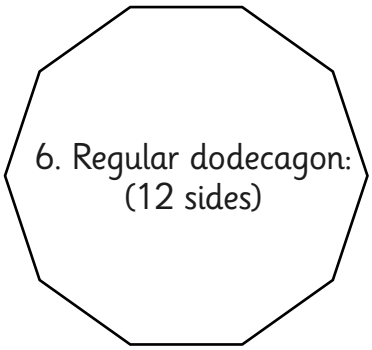


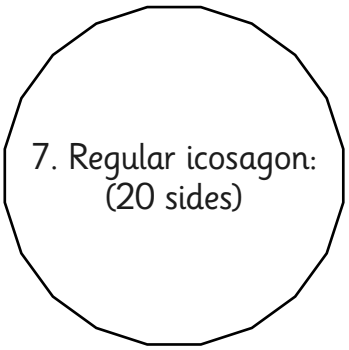
Now try drawing the following algorithms.

Calculate the turn by dividing 360 by the amount of sides the shape has. Take care not to make the sides too long!







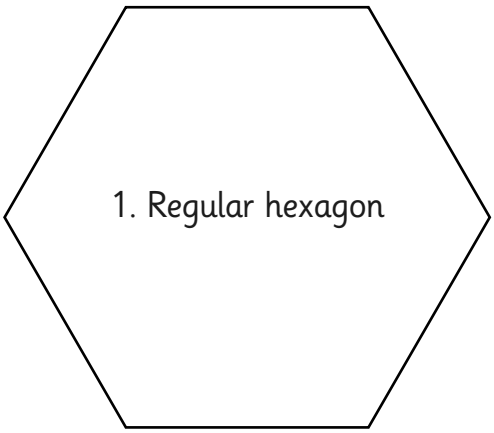


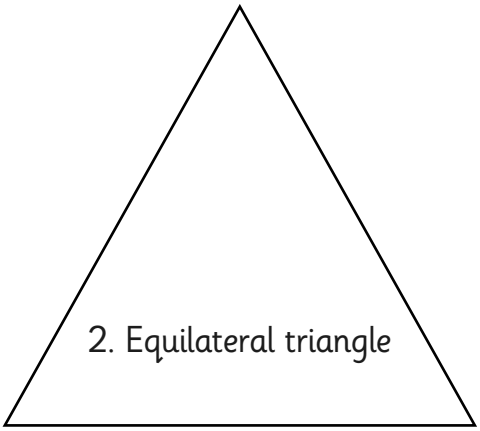
Now try drawing some other regular polygons.

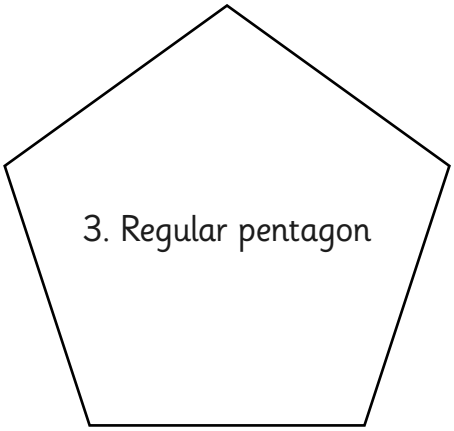
Have you come across any problems with any of the shapes?

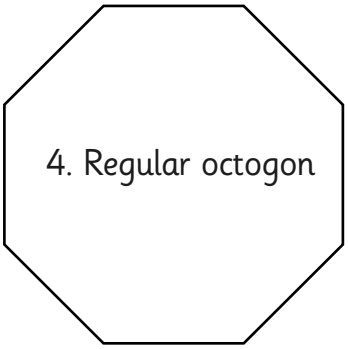
Regular Polygons

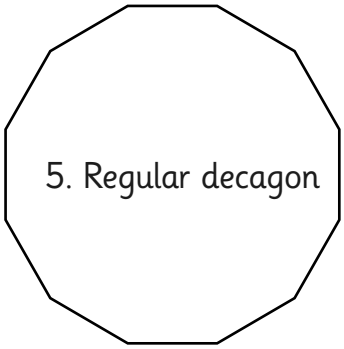
Draw the following algorithms in Turtle Logo. Remember to snip or take a screen shot of your work to save your pictures, patterns and algorithms. Calculate the turn by dividing 360 by the amount of sides the shape has. Take care not to make the sides too long!

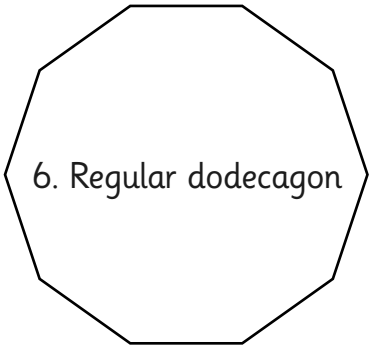


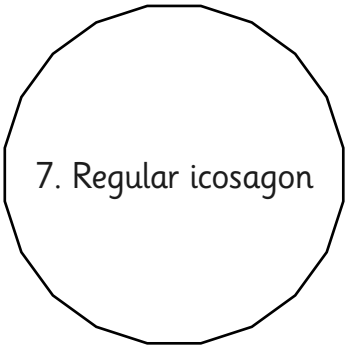












Now try drawing some other regular polygons and answer the questions below:

Have you come across any problems with any of the shapes?

What happens as you draw regular polygons with more and more sides?

What is the maximum number of sides that you can draw a regular polygon with?

Avery Template: Name Badge Label, 8 per sheet | **Compatible Products**:15395, 25395, 42395, 45395, 48395, 5395, 8395, 88395, 85395.

Programming Turtle Logo and Scratch | Regular Polygons

I can create and debug algorithms that draw regular polygons.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can rotate the turtle angles other than 90°.		
I can use calculations as a variable.		

Programming Turtle Logo and Scratch | Regular Polygons

I can create and debug algorithms that draw regular polygons.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can rotate the turtle angles other than 90°.		
I can use calculations as a variable.		

Programming Turtle Logo and Scratch | Regular Polygons

I can create and debug algorithms that draw regular polygons.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can rotate the turtle angles other than 90°.		
I can use calculations as a variable.		

Programming Turtle Logo and Scratch | Regular Polygons

I can create and debug algorithms that draw regular polygons.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can rotate the turtle angles other than 90°.		
I can use calculations as a variable.		

Programming Turtle Logo and Scratch | Regular Polygons

I can create and debug algorithms that draw regular polygons.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can rotate the turtle angles other than 90°.		
I can use calculations as a variable.		

Programming Turtle Logo and Scratch | Regular Polygons

I can create and debug algorithms that draw regular polygons.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can rotate the turtle angles other than 90°.		
I can use calculations as a variable.		

Programming Turtle Logo and Scratch | Regular Polygons

I can create and debug algorithms that draw regular polygons.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can rotate the turtle angles other than 90°.		
I can use calculations as a variable.		

Programming Turtle Logo and Scratch | Regular Polygons

I can create and debug algorithms that draw regular polygons.		
I can write commands in the correct order.		
I can write a variable value where required.		
I can correct any mistakes.		
I can rotate the turtle angles other than 90°.		
I can use calculations as a variable.		

Programming Turtle Logo and Scratch: Drawing

Aim:	Success Criteria:	Resources:
Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts. Use sequence, selection, and repetition in programs; work with variables and various forms of input and output. Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs. This unit continues the learning from the Year 2 Turtle Logo units and links well to shape and direction in Maths. I can create and debug algorithms to draw shapes.	I can use commands in the correct order. I can use a variable value where required. I can correct any mistakes. I can use the pen block to draw.	**Lesson Pack.** Desktop computer /laptops Scratch v2 installed or use online application. Alternatively, use Pyonkee on iPads.
	Key/New Words: Sprite, block, command, background or backdrop, algorithm, move, turn, green flag, key press, pen.	**Preparation:** Ensure application is installed on the computers, or available online. It will help if teachers work through the unit prior to teaching the children to ensure familiarity.

Prior Learning: Children will have created an algorithm to draw a regular polygon in lesson 3.

Learning Sequence

	**Can You Remember?** Ask the children to create an algorithm that will do the following: • Move forward 50 and change colour • Move back 100 and say "Forward again!" • Move back to the start and make a sound Children repeat this algorithm 8 times and program the algorithm to start when the Green flag is pressed. They then change the backdrop, add a different sprite that follows their own algorithm and start when the space bar is pressed.	
	**Algorithms:** Look at the answers to the first two parts above. Click on the algorithms to see them run in v2 online. The third has many possibilities. *(These initial activities will demonstrate how well the children can remember the commands they have learnt. Make note of the skills children are less confident with.)*	
	**Remember the Blocks:** Demonstrate, or get a child to demonstrate, the skills that need reinforcing *(from your observations in the initial activity).* **Drawing:** Show the children how to use the pen block to draw as the sprite moves. **Turning:** Show how to turn the cat (as in Turtle Logo). **Duplicate:** (Copy) Demonstrate how to copy blocks using the stamp or right click, duplicate. **Clear Screen:** Show how to use the blocks to clear and centre the sprite. **Saving Projects:** Show children how to save their projects. *(If using the standalone applications children can save their projects into their folder. If using the online version, they will need to register or download to a computer each time they save.)*	
	**Drawing Shapes:** Children use the pen, forward and turn blocks to draw different size squares and rectangles. They can then add the key press block to each algorithm. Children save their projects if possible.  The algorithm is provided to copy.  The algorithm is given in part to complete.  The blocks needed are listed.	
	**Share:** Children compare their algorithms, test and debug, with a partner.	
	**What Will Happen?** Show the example algorithms on the **Lesson Presentation** and ask if they will work. Click to load the algorithms in Scratch to test.	

Taskit

***Scratch*it:** If the children have access to computers, laptops or tablets, the children could continue to work through the **Challenge Cards** or create their own algorithms.

Computing

Programming Turtle Logo and Scratch


Drawing

Aim

- I can create and debug algorithms that draw shapes.

Success Criteria

- I can use commands in the correct order.
- I can use a variable value where required.
- I can correct any mistakes.
- I can use the pen block to draw.

Can You Remember?

Create an algorithm that will achieve the following:

1. Move forward 50 and change colour.
2. Move back 100 and say “Forward again!”.
3. Move back to the start and make a sound.
4. Have the algorithm repeat 8 times.
5. Start when the green flag is clicked.

Save your work, then...

Finally:

- Add a backdrop and another sprite.
- Create your own algorithm for the new sprite which starts when you press the space bar.

Algorithms

Click on the algorithms to see them running in Scratch

Remember the Blocks

Do you remember the blocks?

Moves the sprite forwards or backwards the number in the box.

Plays the chosen drum sound.

Changes the colour of the sprite.

Shows the words in the box for the time shown.

Starts the algorithm when the green flag is clicked.

Starts the algorithm when the space bar is pressed.

Repeats the algorithm enclosed for the number of times shown.

Drawing

We can draw using the pen blocks.

pen up

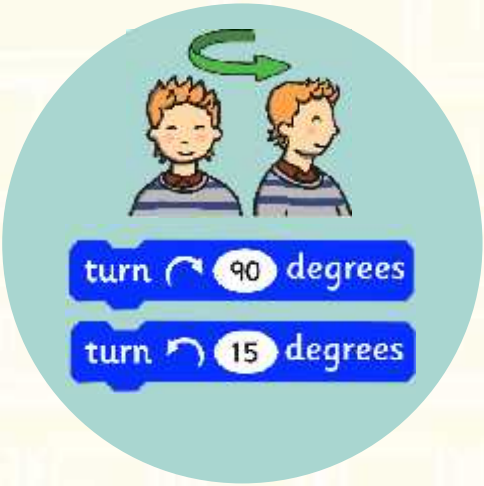
pen down

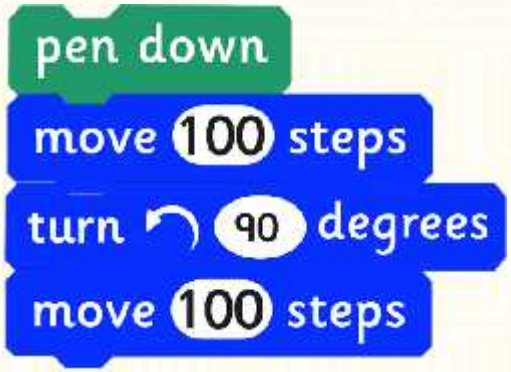
pen down

move 10 steps

Turning

We can draw using the Turn blocks.





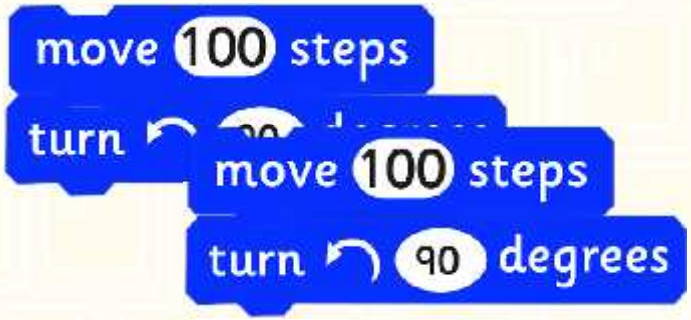
Duplicate

We can duplicate (copy) blocks.

Stamp tool

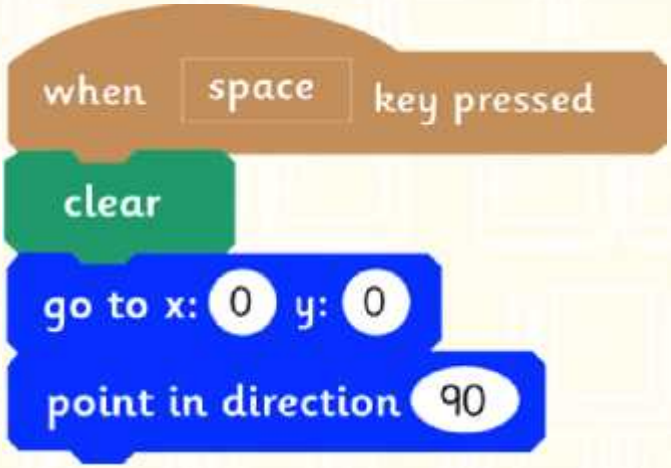
duplicate

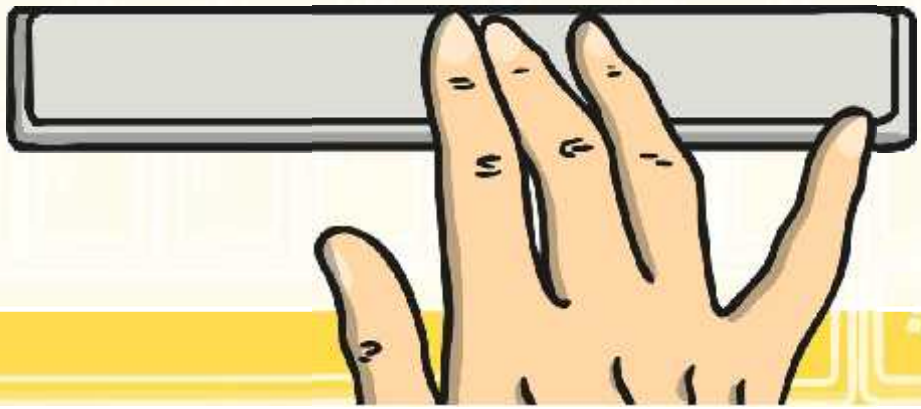
Right click, then select duplicate.



Clear Screen

Use the following algorithm to clear the screen.

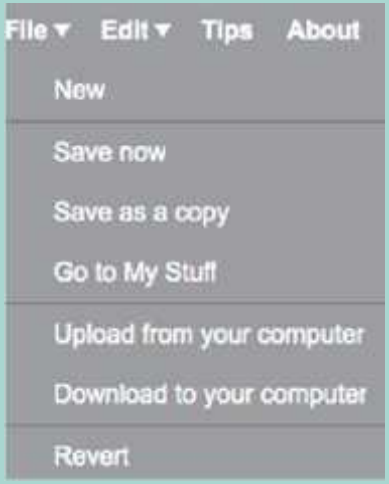




Saving Projects

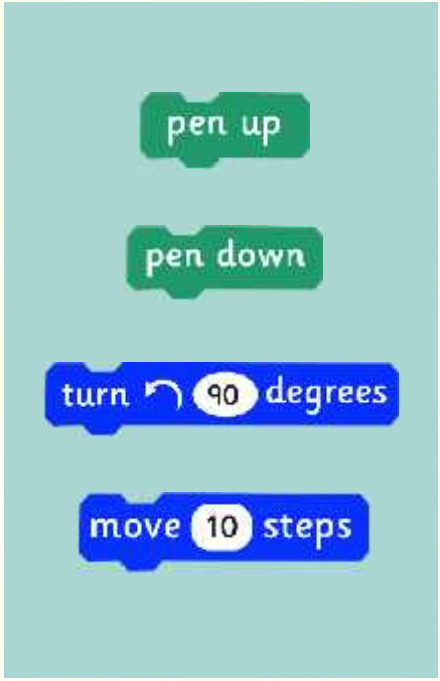
Select file, then save.

If you're working online, save if you have registered or download to your computer.

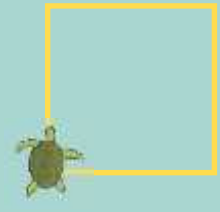


Drawing Shapes

Use the pen, forward and turn blocks to draw the shapes below.
Think about how you will start each of the shapes, use a different letter for each.



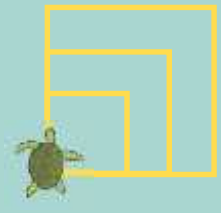
A square.



A rectangle.

A set of growing rectangles.

A set of growing squares.



Now try drawing the same shapes using repeat.

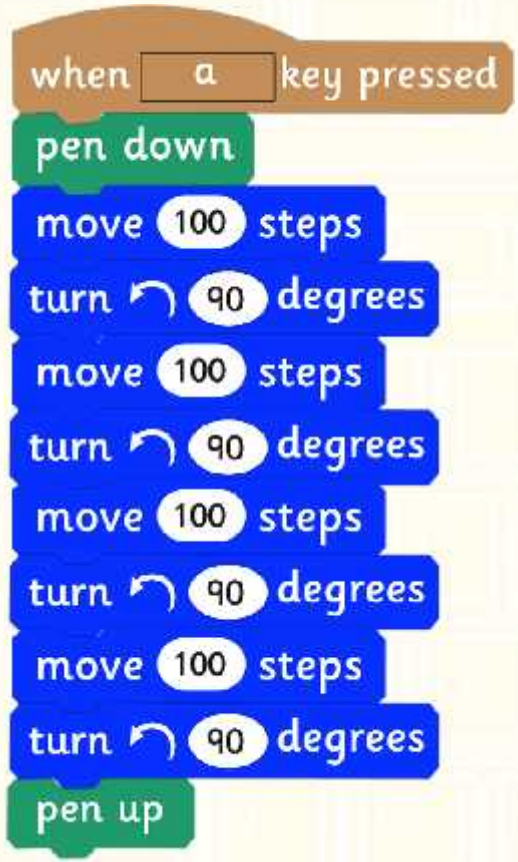
Share

Share your algorithm with your partner.

Test and debug it if you need to.

What Will Happen?

What will happen when these algorithms run?

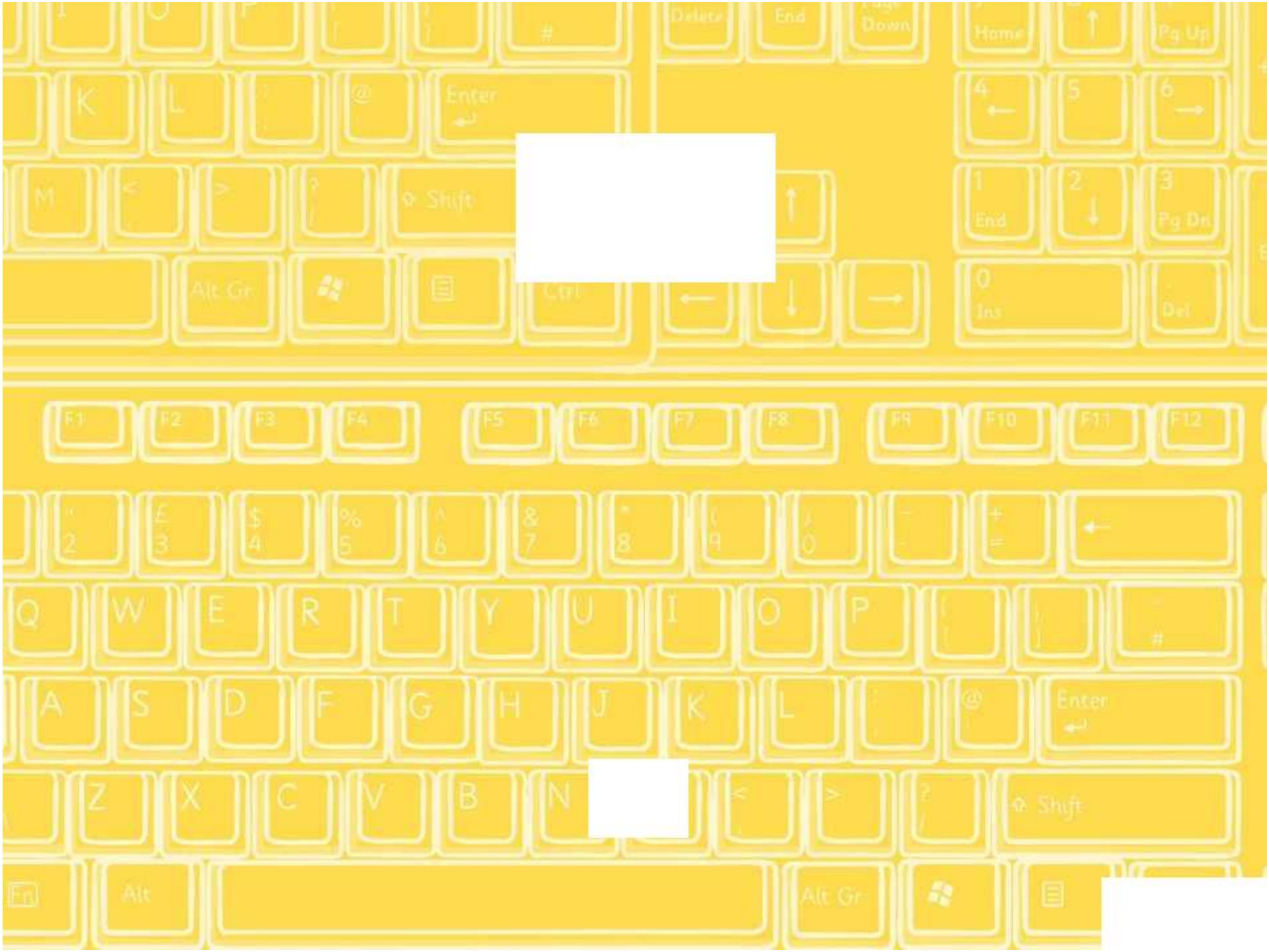


Aim

- I can create and debug algorithms that draw shapes.

Success Criteria

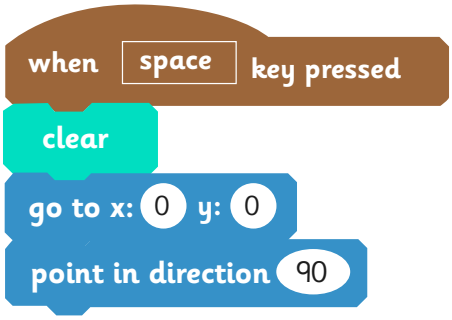
- I can use commands in the correct order.
- I can use a variable value where required.
- I can correct any mistakes.
- I can use the pen block to draw.



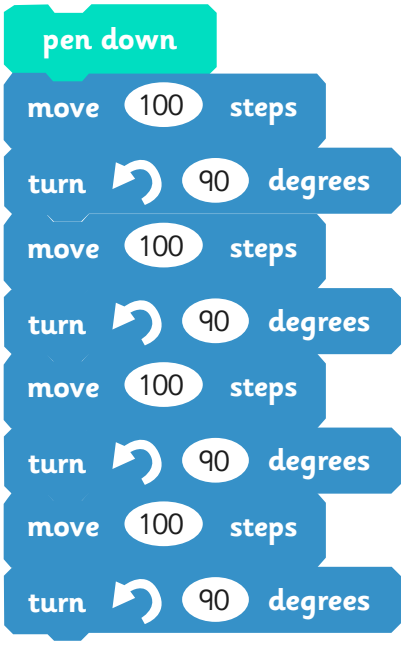
Drawing

Use Scratch to create the shapes below. Before you begin each algorithm remember to clear your screen using the algorithm opposite.

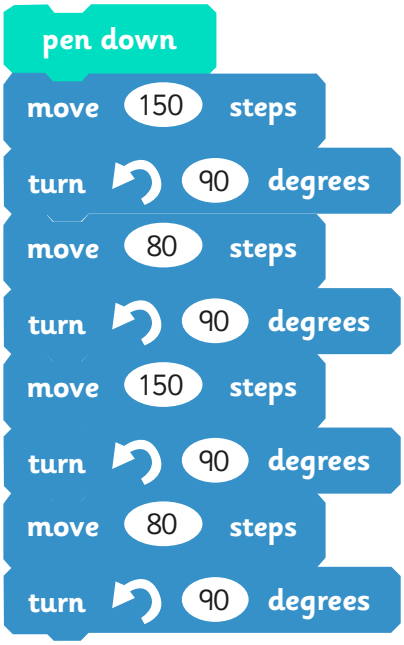
Remember to save your work after each project.



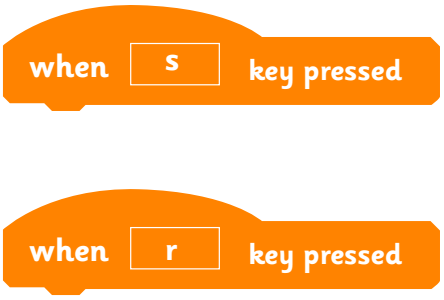
1. Draw a square using the following algorithm.



2. Draw a rectangle using the following algorithm.

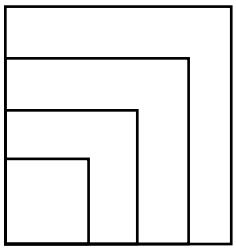


3. Add the key press block to the start of your algorithms.



Start the square with an 's' and the rectangle with an 'r'.

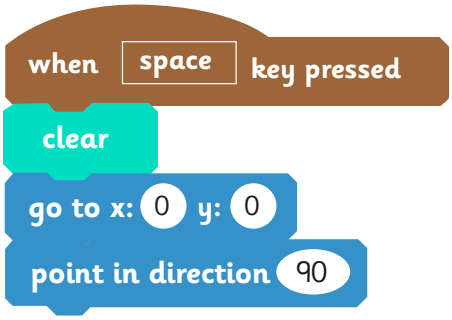
4. Draw a set of growing squares using the same blocks in algorithms above. You will need to change the variables (numbers) in all of the blocks.



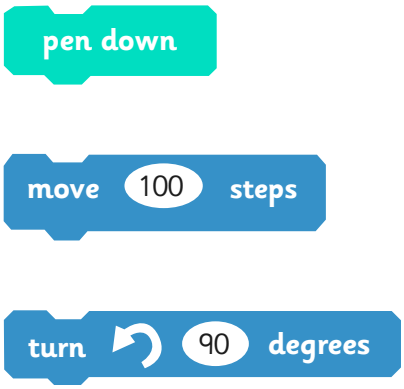
Drawing

Use Scratch to create the shapes below. Before you begin each algorithm remember to clear your screen using the algorithm opposite.

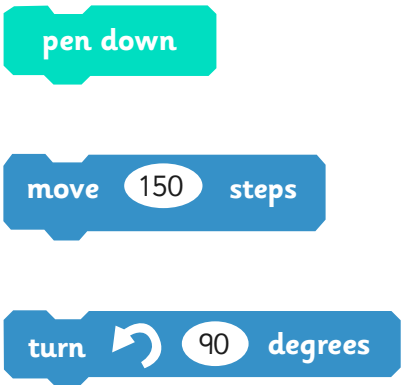
Remember to save your work after each project.



1. Draw a square using the following blocks.



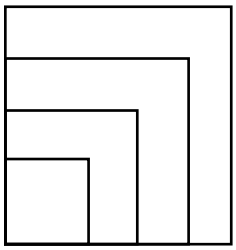
2. Draw a rectangle using the following blocks.

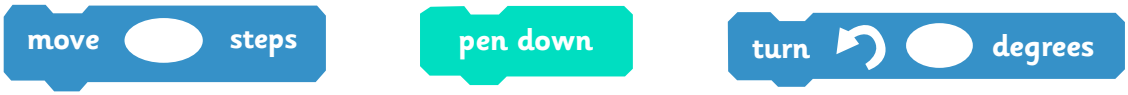


3. Add the key press block to the start of your algorithms.

Start the square with an 's' and the rectangle with an 'r'.

4. Draw a set of growing squares using the same blocks in algorithms above. You will need to change the variables (numbers) in all of the blocks.





5. Draw a set of growing rectangles.

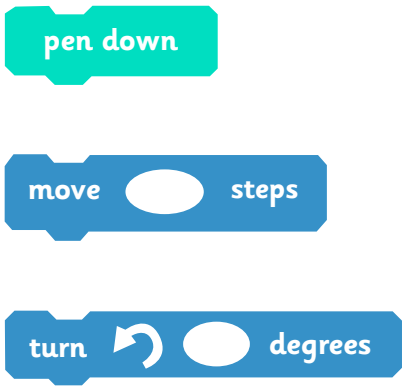
6. Draw the growing squares and rectangles again but this time use the repeat block.

Drawing

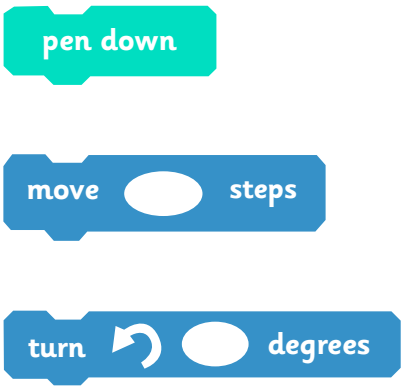
Use Scratch to create the shapes below. Before you begin each algorithm remember to clear your screen.

Remember to save your work after each project.

1. Draw a square using the following blocks.



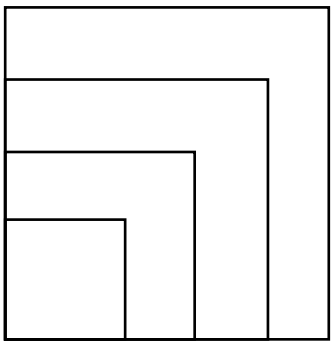
2. Draw a rectangle using the following blocks.



3. Add the key press block to the start of your algorithms.

Start the square with an 's' and the rectangle with an 'r'.

4. Draw a set of growing squares.



5. Draw a set of growing rectangles.

6. Draw the growing squares and rectangles again but this time use the repeat block.

Avery Template: Name Badge Label, 8 per sheet | **Compatible Products**:15395, 25395, 42395, 45395, 48395, 5395, 8395, 88395, 85395.

Programming Turtle Logo and Scratch | Drawing

I can create and debug algorithms to draw shapes.		
I can use commands in the correct order.		
I can use a variable value where required.		
I can correct any mistakes.		
I can use the pen block to draw.		

Programming Turtle Logo and Scratch | Drawing

I can create and debug algorithms to draw shapes.		
I can use commands in the correct order.		
I can use a variable value where required.		
I can correct any mistakes.		
I can use the pen block to draw.		

Programming Turtle Logo and Scratch | Drawing

I can create and debug algorithms to draw shapes.		
I can use commands in the correct order.		
I can use a variable value where required.		
I can correct any mistakes.		
I can use the pen block to draw.		

Programming Turtle Logo and Scratch | Drawing

I can create and debug algorithms to draw shapes.		
I can use commands in the correct order.		
I can use a variable value where required.		
I can correct any mistakes.		
I can use the pen block to draw.		

Programming Turtle Logo and Scratch | Drawing

I can create and debug algorithms to draw shapes.		
I can use commands in the correct order.		
I can use a variable value where required.		
I can correct any mistakes.		
I can use the pen block to draw.		

Programming Turtle Logo and Scratch | Drawing

I can create and debug algorithms to draw shapes.		
I can use commands in the correct order.		
I can use a variable value where required.		
I can correct any mistakes.		
I can use the pen block to draw.		

Programming Turtle Logo and Scratch | Drawing

I can create and debug algorithms to draw shapes.		
I can use commands in the correct order.		
I can use a variable value where required.		
I can correct any mistakes.		
I can use the pen block to draw.		

Programming Turtle Logo and Scratch | Drawing

I can create and debug algorithms to draw shapes.		
I can use commands in the correct order.		
I can use a variable value where required.		
I can correct any mistakes.		
I can use the pen block to draw.		

Programming Turtle Logo and Scratch: Regular Polygons in Scratch

Aim:	**Success Criteria:**	**Resources:**
Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts. Use sequence, selection, and repetition in programs; work with variables and various forms of input and output. Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs. This unit continues the learning from the Year 2 Turtle Logo units and links well to shape and direction in Maths. I can create and debug algorithms that draw regular polygons.	I can use commands in the correct order. I can use a variable value where required. I can correct any mistakes. I can create algorithms that draw regular polygons.	**Lesson Pack** Desktop computer /laptops Scratch v2 installed or use online application. Alternatively, use Pyonkee on iPads.
	Key/New Words: Sprite, block, command, background or backdrop, algorithm, move, turn, green flag, key press, pen, repeat.	**Preparation:** Ensure application is installed on the computers, or available online. It will help if teachers work through the unit prior to teaching the children to ensure familiarity.

Prior Learning: It would be helpful if children are able to write simple algorithms with blocks in Scratch and save files.

Learning Sequence

	**What Can You Do?** Children draw a regular hexagon using blocks, remembering how they did it with Turtle Logo and applying the same ideas to a different way of programming. Try other polygons and use the repeat command.	
	**Using Repeat:** Demonstrate how to use the repeat command to create algorithms for different regular polygons. **How Could You Start?** Show how to add the key press block to start the algorithm.	
	**Algorithms for Regular Polygons:** Children create algorithms for regular polygons, using the key press command to start each polygon with a different letter. Save the project.  Children are given the algorithms for a pentagon and hexagon, and given the sides and turns required for other regular polygons.  Children are given the blocks needed to write an algorithm for a pentagon and hexagon, and given the sides and turns required for other regular polygons.  Children need to work out the algorithm, including sides and angles, independently.	
	**Share:** Children share their algorithms, test and debug. They should be ready to share their ideas with the rest of the class.	
	**Have Another Go!** Children make any alterations or test ideas from their partner. Can you make the pattern shown on the **Lesson Presentation?**	
	**Which Algorithm?** Which algorithm will draw the pattern? Ask the children to explain why the other one will not. Click options for answers and to see the algorithms running online.	

Task*it*

***Polygon*it:** If the children have access to computers, laptops or tablets, the children could continue to create their own algorithms for regular polygons and patterns.

Computing

Programming Turtle Logo and Scratch

Regular Polygons in Scratch

Aim

- I can create and debug algorithms that draw regular polygons.

Success Criteria

- I can use commands in the correct order.
- I can use a variable value where required.
- I can correct any mistakes.
- I can create algorithms that draw regular polygons.

What Can You Do?

Create an algorithm that will draw a regular hexagon.

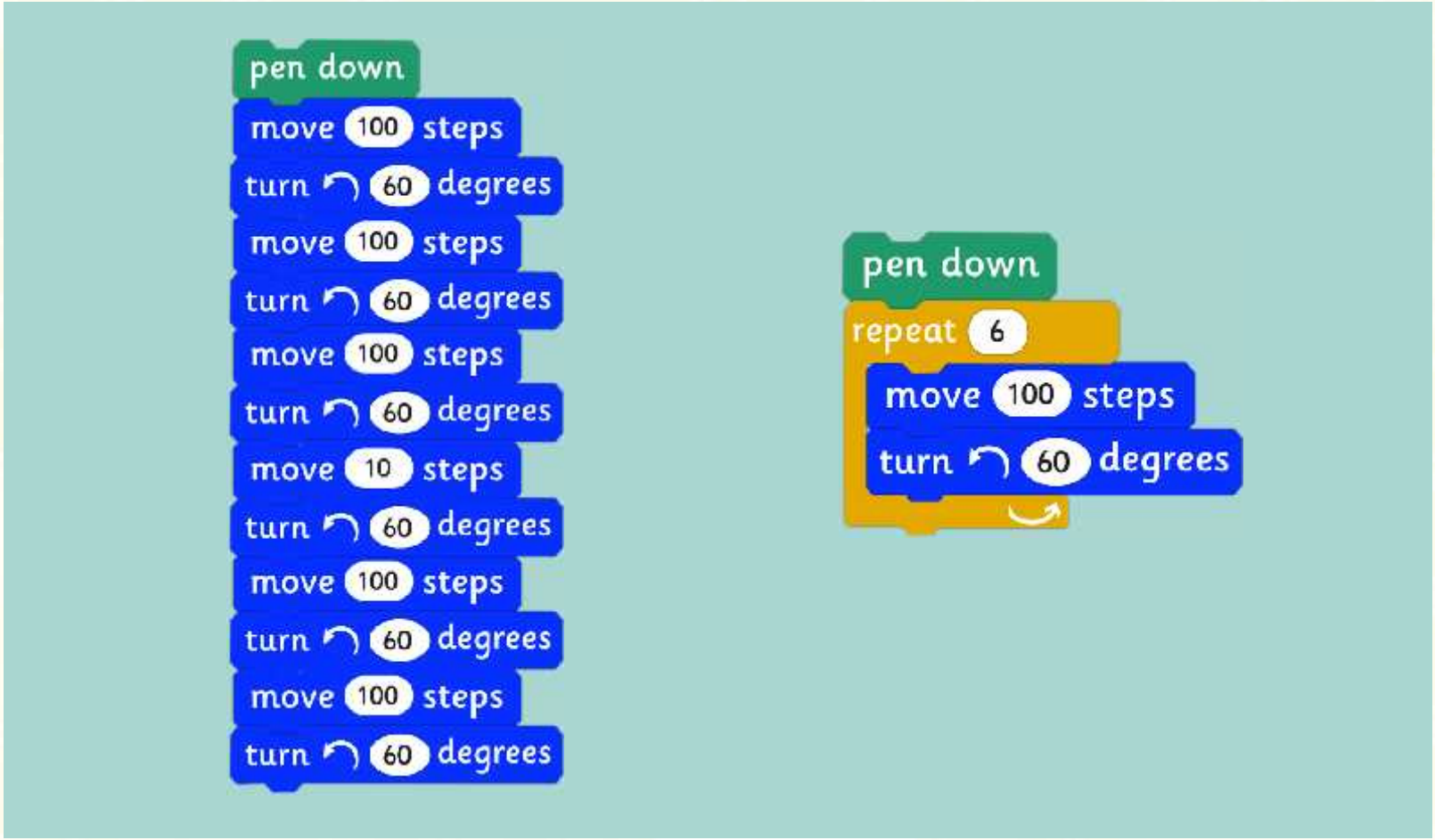
Think about the algorithm that you used in Turtle Logo.

What turn do you need?

Try other regular polygons using the repeat command.

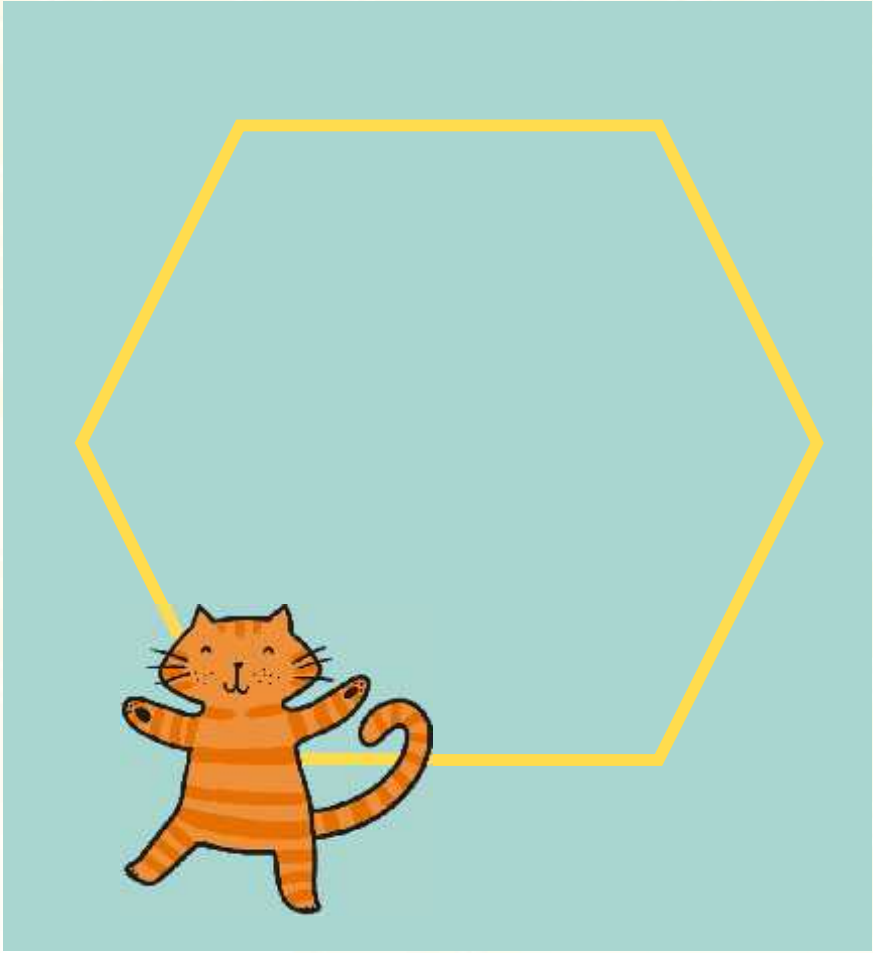
Possible Algorithms

Compare the 2 algorithms with and without repeat.



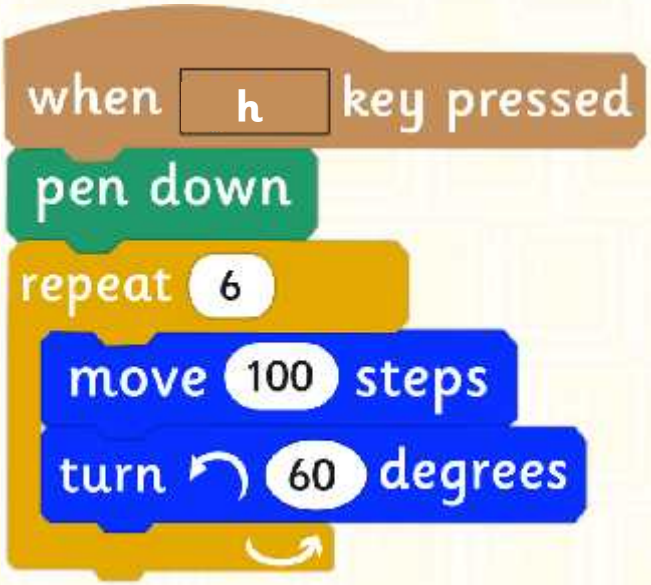
Using Repeat

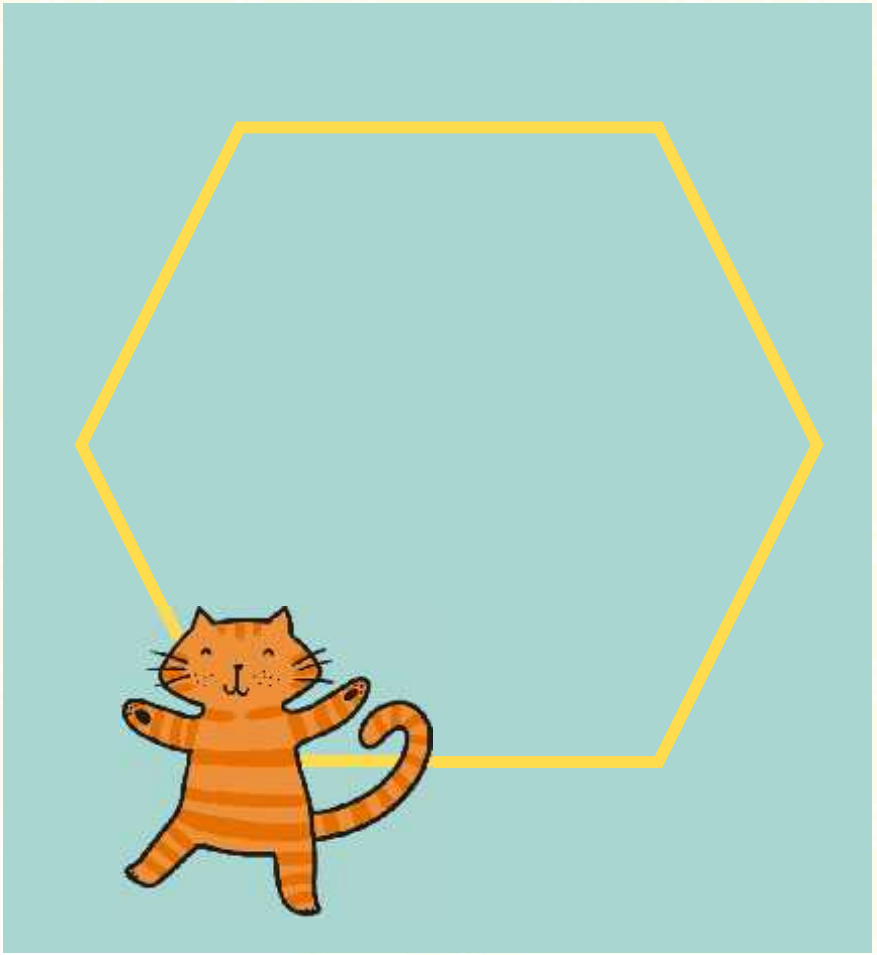
The repeat block can be used to draw regular polygons by wrapping the move and turn.



How Could You Start?

Add a key press block to start the algorithm. Choose which letter or number you want to start with.

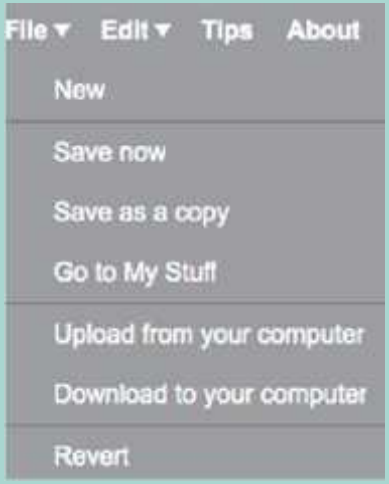




Saving Projects

Select file, then save.

If your working online, save if you have registered or download to your computer.

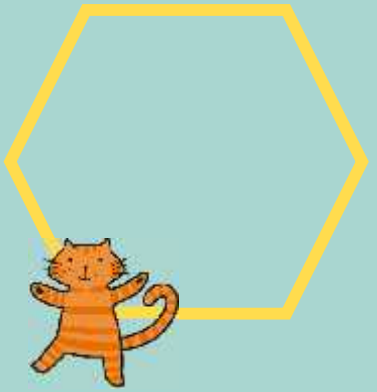


Algorithms for Regular Polygons

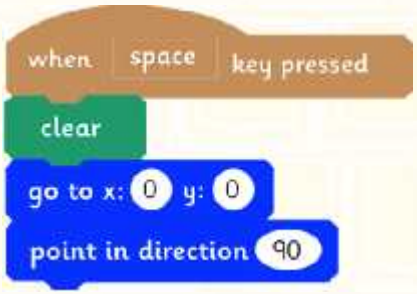
Create algorithms for regular polygons.

How many sides?

What angles will you turn?



Remember to use pen down.



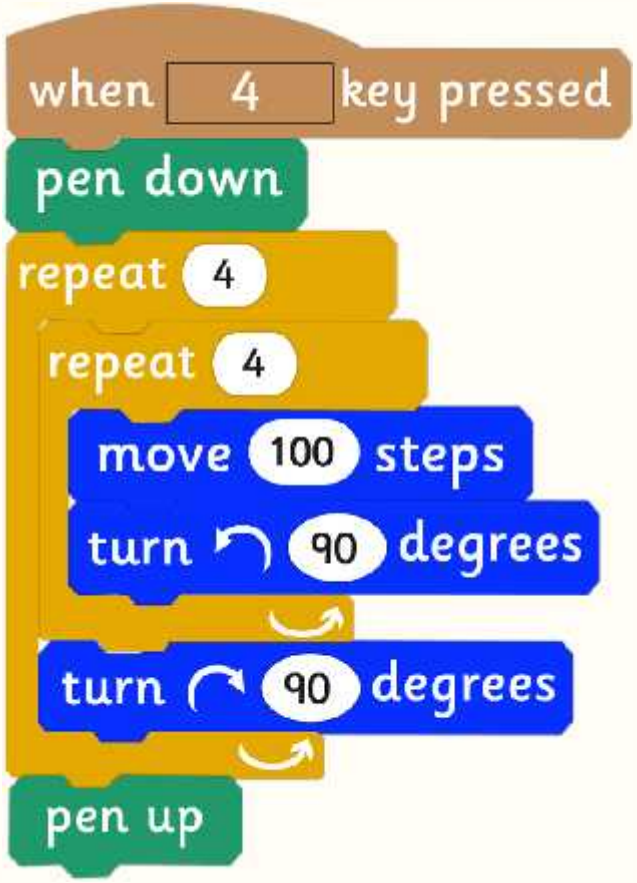
And clear screen.

Share

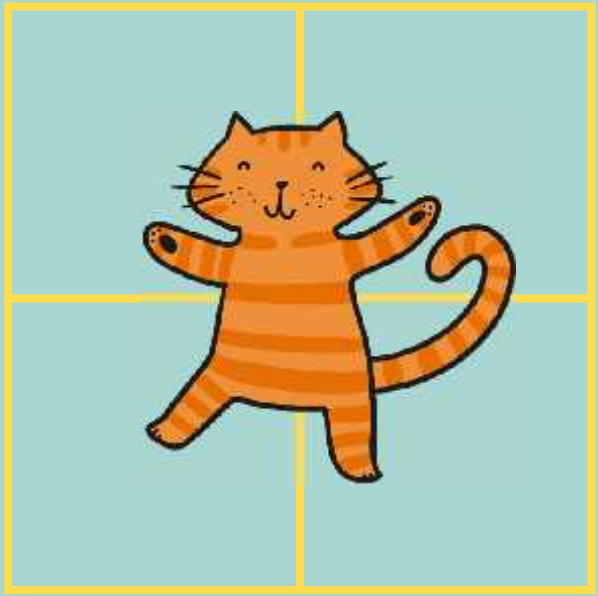
Share your algorithm with your partner.

Test and debug it if you need to.

Have Another Go!



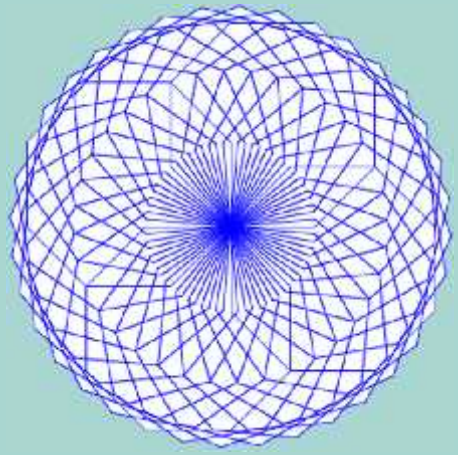
Can you make an algorithm for this pattern?



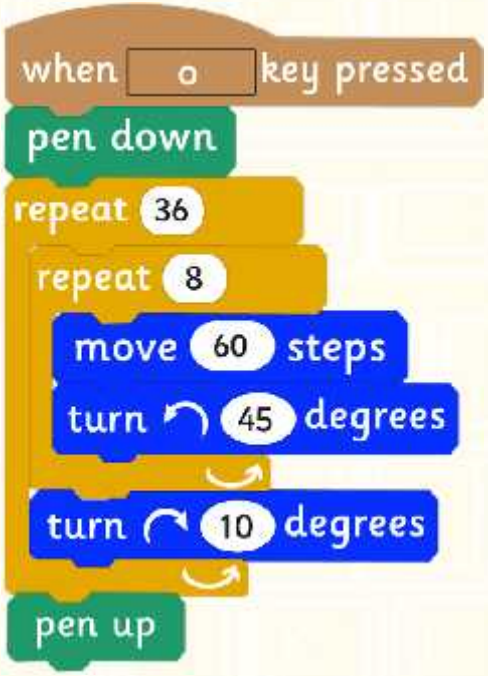
Click the cat to reveal the answer.

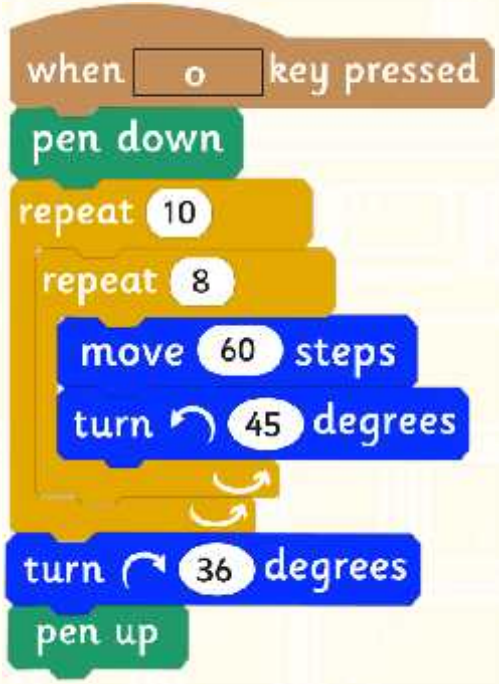
Which Algorithm?

Which algorithm will make this pattern?



Why will only one of them work?

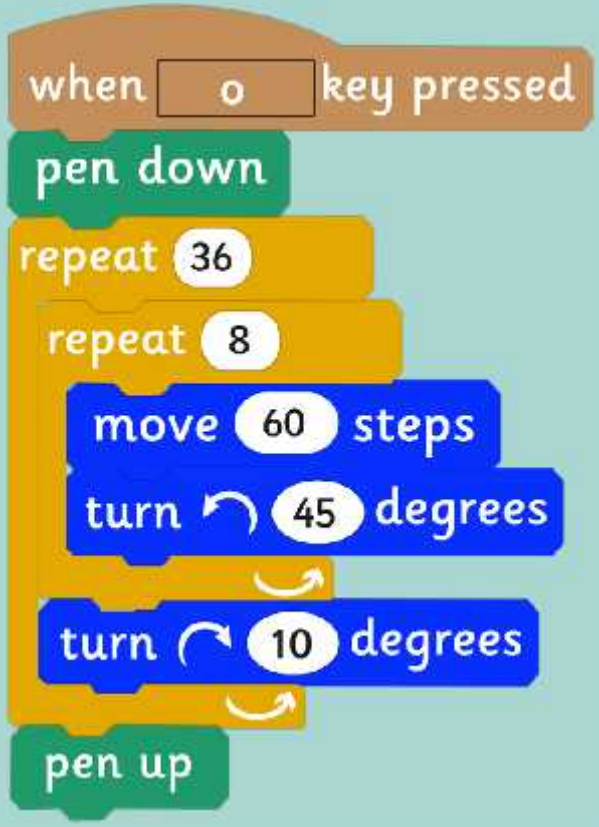




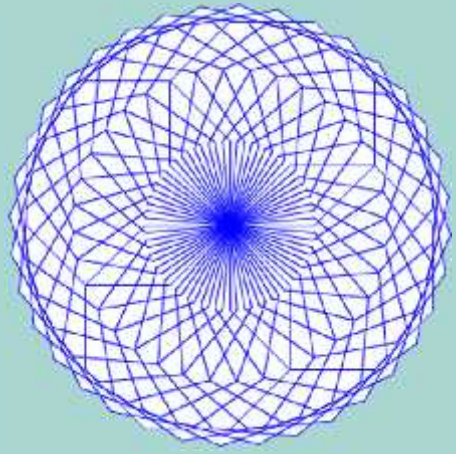
Click on each algorithm to reveal the answer.

Correct

Click on the algorithm to run online.

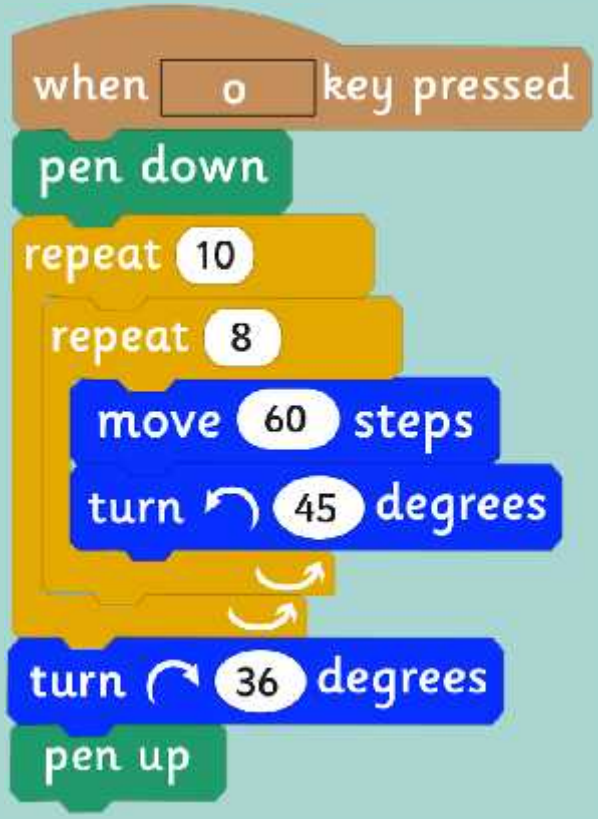


1. Start by pressing “o”
2. Pen down
3. Repeats the octagon 36 times
4. Turns 10° after each octagon
5. Pen up at the end

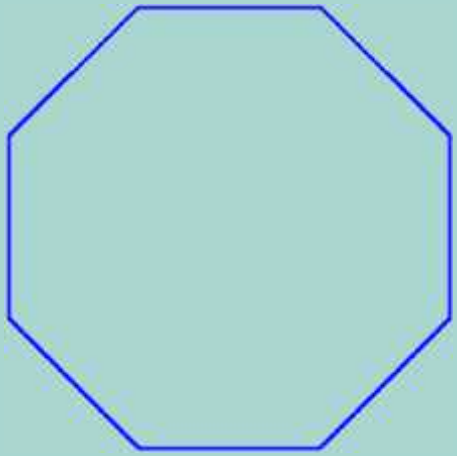


Incorrect

Click on the algorithm to run online.



1. Start by pressing “o”
2. Pen down
3. Repeats the octagon **10 times**
4. **Turns 36°** *after* **all of the octagons**
5. Pen up at the end

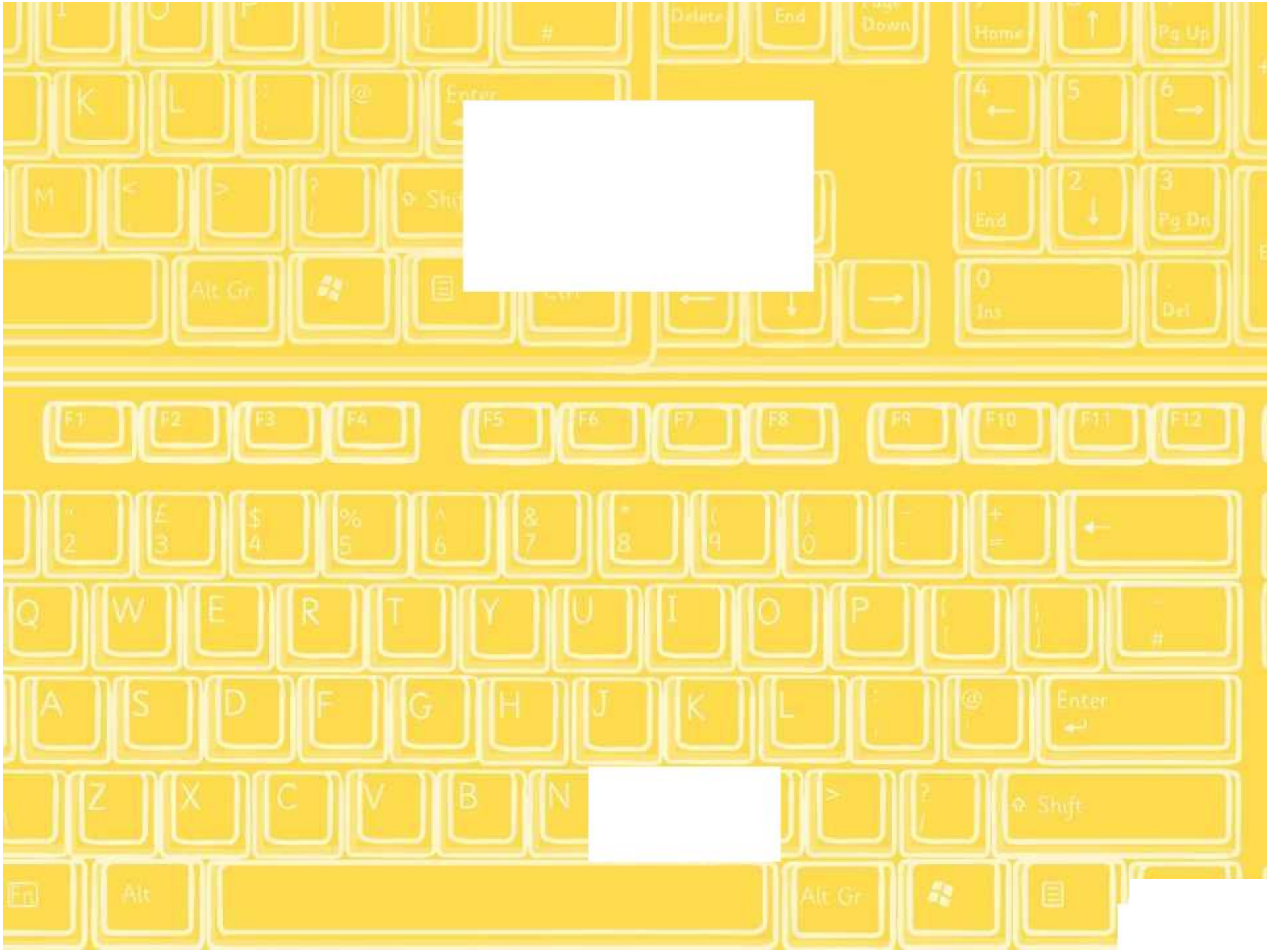


Aim

- I can create and debug algorithms that draw regular polygons

Success Criteria

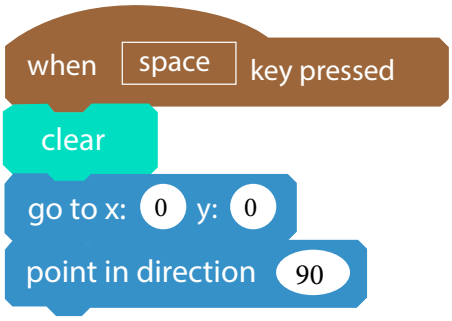
- I can use commands in the correct order.
- I can use a variable value where required.
- I can correct any mistakes.
- I can create algorithms that draw regular polygons.



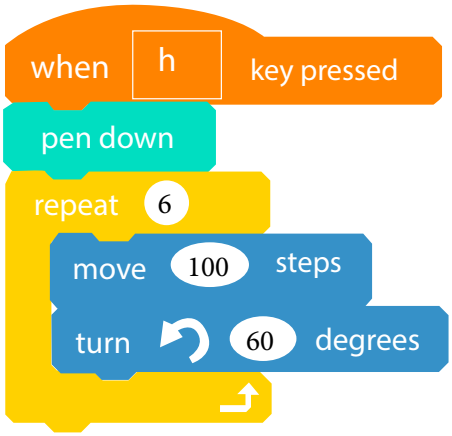
Regular Polygons in Scratch

Before you begin each algorithm remember to clear your screen.

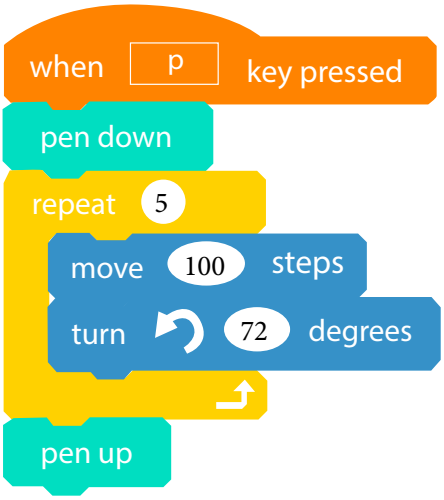
Create the following algorithms in Scratch. Start each shape with the key press block, using the first letter from the name of the shape.



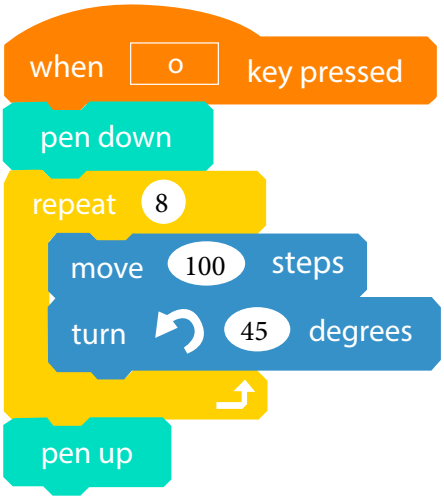
1. Draw a regular hexagon using the following algorithm.



2. Draw a regular pentagon using the following algorithm.



3. Draw a regular octogon using the following algorithm.



Don't forget to save your projects! Now draw the following shapes:

4. A regular decagon
Sides = 10
Turm = 36°

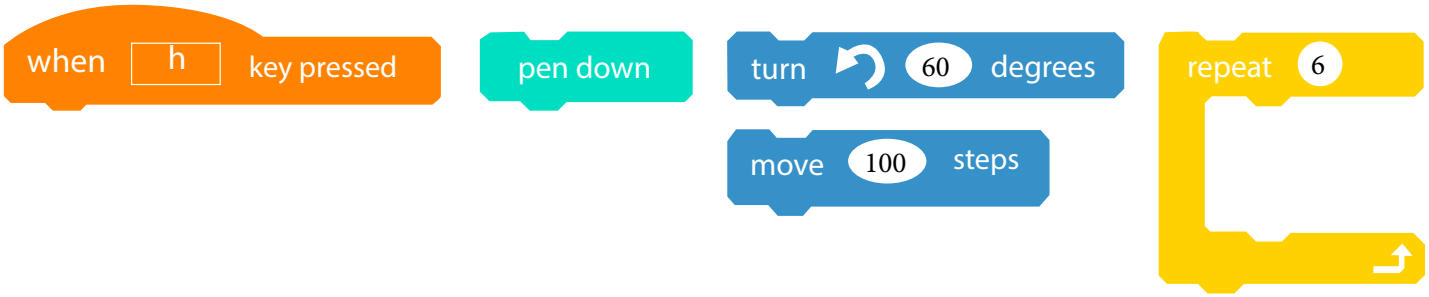
5. A regular dodeca-gon
Sides = 12
Turm = 30°

6. A regular icosagon
sides = 20
Turm = 18°

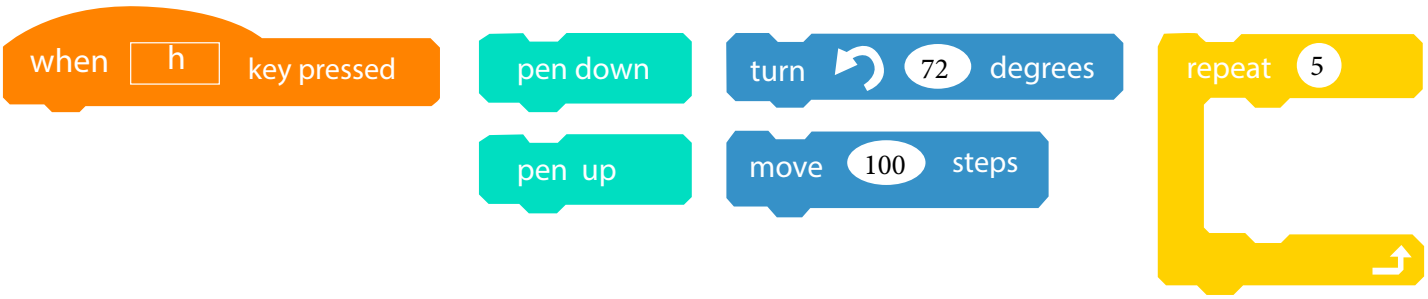
Regular Polygons in Scratch

Create the following algorithms in Scratch. Start each shape with the key press block, using the first letter from the name of the shape. Before you begin each algorithm remember to clear your screen.

1. Using the blocks below draw a regular hexagon. Save your project.



2. Using the blocks below draw a regular pentagon. Save your project.



3. Draw a regular octagon using the blocks above. The turn for the octagon is 45°, you will need to work out how many sides are needed. Don't forget to save your projects!

Now draw the following shapes:

4. A regular decagon
Sides = 10
Turn = 36°

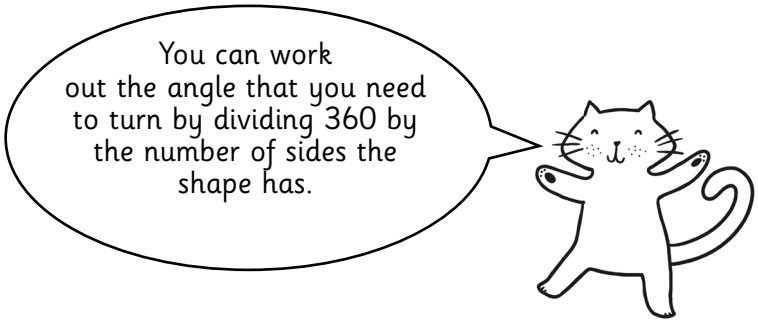
5. A regular dodecagon
Sides = 12
Turn = 30°

6. A regular icosagon
Sides = 20
Turn = 18°

Challenge
Try drawing different patterns by repeating regular polygons and turning after each one.

Regular Polygons in Scratch

Create the following algorithms in Scratch. Start each shape with the key press block, think about what the best letter to use for each shape would be. Before you begin each algorithm remember to clear your screen.



1. A regular hexagon.

2. A regular pentagon.

3. A regular octagon.

4. A regular decagon
Sides = 10

5. A regular dodecagon
Sides = 12

6. A regular icosagon
sides = 20

Challenge

Try drawing different patterns by repeating regular polygons and turning after each one. Now try creating the pattern using a single algorithm. Save your work.

Programming Turtle Logo and Scratch | Regular Polygons in Scratch

I can create and debug algorithms that draw regular polygons.		
I can use commands in the correct order.		
I can use a variable value where required.		
I can correct any mistakes.		
I can create algorithms that draw regular polygons.		

Programming Turtle Logo and Scratch | Regular Polygons in Scratch

I can create and debug algorithms that draw regular polygons.		
I can use commands in the correct order.		
I can use a variable value where required.		
I can correct any mistakes.		
I can create algorithms that draw regular polygons.		

Programming Turtle Logo and Scratch | Regular Polygons in Scratch

I can create and debug algorithms that draw regular polygons.		
I can use commands in the correct order.		
I can use a variable value where required.		
I can correct any mistakes.		
I can create algorithms that draw regular polygons.		

Programming Turtle Logo and Scratch | Regular Polygons in Scratch

I can create and debug algorithms that draw regular polygons.		
I can use commands in the correct order.		
I can use a variable value where required.		
I can correct any mistakes.		
I can create algorithms that draw regular polygons.		

Programming Turtle Logo and Scratch | Regular Polygons in Scratch

I can create and debug algorithms that draw regular polygons.		
I can use commands in the correct order.		
I can use a variable value where required.		
I can correct any mistakes.		
I can create algorithms that draw regular polygons.		

Programming Turtle Logo and Scratch | Regular Polygons in Scratch

I can create and debug algorithms that draw regular polygons.		
I can use commands in the correct order.		
I can use a variable value where required.		
I can correct any mistakes.		
I can create algorithms that draw regular polygons.		

Programming Turtle Logo and Scratch | Regular Polygons in Scratch

I can create and debug algorithms that draw regular polygons.		
I can use commands in the correct order.		
I can use a variable value where required.		
I can correct any mistakes.		
I can create algorithms that draw regular polygons.		

Programming Turtle Logo and Scratch | Regular Polygons in Scratch

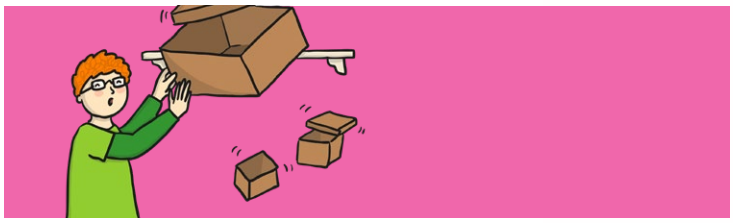
I can create and debug algorithms that draw regular polygons.		
I can use commands in the correct order.		
I can use a variable value where required.		
I can correct any mistakes.		
I can create algorithms that draw regular polygons.		

Programming Turtle Logo and Scratch

Computing | Year 3 | Unit Overview

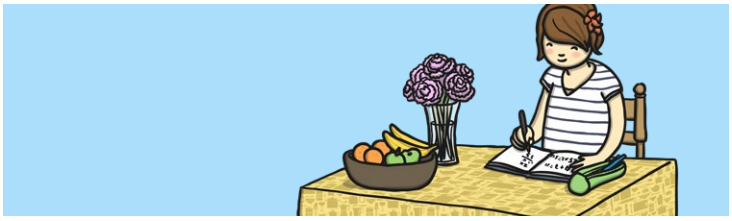
Introduction

This Programming Turtle Logo and Scratch unit will teach your class to create and debug algorithms. Following on from the earlier Year 2 unit on Preparing for Turtle Logo, the children use the basic commands in Logo to move and draw using the turtle on screen, and then further develop algorithms using the "repeat" command. These skills are then developed by teaching children to create algorithms in Scratch using a selection of blocks.



Health & Safety

Children should be encouraged to have good posture and sit up to the computer.



Home Learning

Task 1 Turtle Logo: Polygons and Patterns: Children use Turtle logo to create different algorithms for different regular polygons and then use them to create a pattern.

Task 2 Scratch: Polygons and Patterns: Children use Scratch to create different algorithms for different regular polygons and then use them to create a pattern.

Assessment Statements

By the end of this unit...

...all children should be able to:

- Create and debug algorithms to draw regular polygons using the repeat command/ block (Turtle Logo and Scratch)

...most children will be able to:

- Draw shapes with spaces between using penup and pendown (Turtle Logo)
- Change and alter the pen settings (Scratch)

...some children will be able to:

- Draw regular polygons using Logo to calculate the angle (Turtle Logo)
- Create and debug algorithms to draw patterns by repeating regular polygons (Scratch)

Lesson Breakdown

1. Backwards

Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.

Use sequence, selection, and repetition in programs; work with variables and various forms of input and output.

Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.

Using Turtle Logo.

- I can create and debug an algorithm using the move, rotate and repeat commands.

Resources

- Desktop Computer or Laptop
- Turtle Logo application (installed or online)
- Whiteboards and pens or books, pens and pencils for recording

2. Pen Up and Pen Down

See above.

Using Turtle Logo.

- I can create and debug algorithms using penup and pendown.

- Desktop Computer or Laptop
- Turtle Logo application (installed or online)
- Whiteboards and pens or books, pens and pencils for recording

3. Regular Polygons

See above.

Using Turtle Logo.

- I can create and debug algorithms that draw regular polygons.

- Desktop Computer or Laptop
- Turtle Logo application (installed or online)
- Whiteboards and pens or books, pens and pencils for recording

4. Drawing

See above.

Using Scratch.

- To create and debug algorithms that draw shapes.

- Desktop Computer or Laptop Scratch application (installed or online)
- Whiteboards and pens or books, pens and pencils for recording

5. Regular Polygons in Scratch

See above.

Using Scratch.

- To create and debug algorithms that draw regular polygons.

- Desktop Computer or Laptop Scratch application (installed or online)

6. Pens

See above.

Using Scratch

- To create and debug algorithms to draw patterns.

- Desktop Computer or Laptop Scratch application (installed or online)
- Whiteboards and pens or books, pens and pencils for recording