

E.a.R.L.2



Scheme of Work

Deliver exciting computer science lessons suitable from EYFS through to KS2

Includes:

- 9 lesson plans
- 20 independent challenges
- National Curriculum links
- Assessment grids to track progress
- E.a.R.L 2 user guide
- Printable resources
- Computer science glossary





Contents

Introduction	4
E.a.R.L 2 and the National Curriculum	5
Demystifying and achieving the primary computing objectives	6
E.a.R.L 2 and Early Years	7
How to use E.a.R.L 2	8
Customising the sounds on E.a.R.L 2	10
Additional E.a.R.L 2 resources to achieve curriculum objectives	11
Programming E.a.R.L 2 using Scratch	13
Computer science terminology	15
Unplugged - an introduction to programming	17
Lesson 1	18
Lesson 2	19
Lesson 3	21
Introducing E.a.R.L 2	22
Lesson 4 - Create a program for E.a.R.L 2	23
Integrating programming and robotics	24
Lesson 5 - E.a.R.L 2 Challenges	24
Challenge 1	25
Challenge 2	25
Challenge 3	25
Challenge 4	25
Challenge 5	26
Challenge 6	26
Challenge 7	26
Challenge 8	26
Lesson 6 E.a.R.L 2 Mat Challenges	27
Treasure Island Challenge 1	28
Challenge 2	28
Challenge 3	28
Challenge 4	28
Challenge 5	29



Street Scene	Challenge 1	30
	Challenge 2	30
	Challenge 3	30
	Challenge 4	30
Solar System	Challenge 1	31
	Challenge 2	31
	Challenge 3	31
	Challenge 4	31
Using E.a.R.L 2 with Scratch	Lesson 7	32
	Lesson 8	33
	Lesson 9	34
	Lesson 10	35
Assessment		36
Appendix		41





E.a.R.L 2 computing scheme of work for primary schools

The computing curriculum is important for preparing children for their future careers in a world where technology is rapidly advancing. Primary schools have a vital role to play in supporting their pupils to meet the demands of a future in a digital age.

Teachers should aim to give their pupils a broad, balanced and connected curriculum. The objectives set out in the national curriculum in computing aim to develop a number of key skills, but it also recognises the need for a cross-curricular approach to learning which provides links to a range of other subjects.

“A high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world.”

(National Curriculum in England, 2014)

This scheme of work contains essential subject terminology and knowledge, with a framework linked to the curriculum, guidance and plans for teaching inspiring computing lessons and assessing progress for all pupils to enable the successful delivery of the computer science objectives within the curriculum at Key Stage 1 and early key stage 2 .

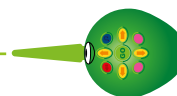
We begin by providing a number of ‘unplugged activities’ which introduces the concept of algorithms and the importance of logical thinking. The children then create their own algorithms to implement as programs on a digital device using E.a.R.L 2.

Computational thinking

A systematic approach to problem solving using logical and critical thinking skills that both humans and computers can understand

The computer science element of the key stage 1 curriculum places a strong emphasis on coding and **computational thinking**. It is vital that these skills are developed at a young age as we scaffold the learning through key stage 1 and into key stage 2. E.a.R.L 2 has been designed to help children achieve all of these objectives.

Pupils can use E.a.R.L 2 to create and debug simple programs that will be followed obediently. Their understanding that programs execute by following precise and unambiguous instructions is enriched through a range of practical and fun activities.



Paul Strickland is the CEO of Digital Wizards Ltd. He has many years' experience working as a primary school teacher, CAS master teacher and computing hub leader.



E.a.R.L 2 and the National Curriculum

E.a.R.L 2 is designed to address several areas of the National Curriculum for the Early Learning Goals and Computing specific objectives in Key Stage 1 and Key Stage 2.

Please find underneath the areas of the curriculum that E.a.R.L 2 will support your pupils.

ELG: Numerical Patterns

Children at the expected level of development will:

- Verbally count beyond 20, recognising the pattern of the counting system.

ELG: People, Culture and Communities

Children at the expected level of development will:

- Describe their immediate environment using knowledge from observation, discussion, stories, non-fiction texts and maps.

ELG: Creating with Materials

Children at the expected level of development will:

- Safely use and explore a variety of materials, tools and techniques, experimenting with colour, design, texture, form and function.

Key Stage 1

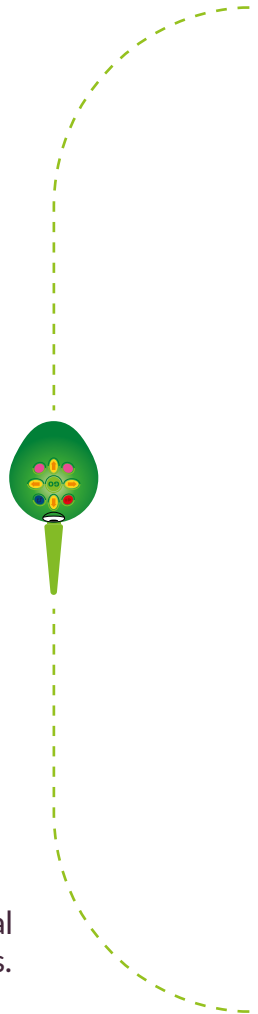
Pupils should be taught to:

- Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise unambiguous instructions.
- Create and debug simple programs.
- Use logical reasoning to predict the behaviour of simple programs.

Key Stage 2

Pupils should be taught to:

- 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.
- Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information.





Demystifying and achieving the primary computing objectives

The new national curriculum for computing provides schools with an exciting opportunity to reinvigorate teaching and learning important skills.

E.a.R.L 2 and the E.a.R.L 2 scheme of work will provide all the guidance and resources you will need to successfully deliver and help children achieve the computer science objectives within the new **computing curriculum**.

Computing Curriculum

The computing programme of study is split into 3 strands of learning; computer science, information technology and digital literacy.



E.a.R.L 2 is a floor robot, packed full of features to enhance the fun of learning from Early Years Foundation Stage, key stage 1 through to early key stage 2. His transparent shell allows pupils to see exactly how he works. With lights and sounds (including the option to record your own), children will love using E.a.R.L 2 to create and debug simple programs as they take their first steps on a coding and programming learning journey.

E.a.R.L 2 brings the curriculum to life, is fun to use and engages children. It provides a clear link between understanding what algorithms are and how they are implemented as programs on digital devices and its multi-sensory approach provides additional learning benefits for all pupils, particularly children with additional learning needs.

E.a.R.L 2 is the perfect resource to lay the foundations and develop children's computational thinking skills from a young age.

Achieving curriculum objectives in other subjects

There are many deep links between computing and other subject areas, particularly in mathematics, English, science and design and technology. Using E.a.R.L 2 as part of a cross-curricular approach to learning curates a mastery approach to learning and allows pupils to apply skills to different areas of the curriculum.

Cross-curricular opportunities have been outlined in this scheme of work. Speaking and listening skills are developed as children learn to communicate more effectively with their peers, developing language both verbally and in writing. It is also worth noting that the life skills gained from lessons using E.a.R.L 2 are invaluable. This includes perseverance, collaboration, creativity, problem solving and critical thinking.



E.a.R.L 2 and Early Years

E.a.R.L 2 allows children to have fun using their creativity and developing skills through play. Exploring E.a.R.L 2 through guided play or undirected play will support a number of areas, these include:

Technology: Children recognise that a range of technology is used in places such as homes and schools. They select and use technology for particular purposes.

Exploring and using media and materials: Children safely use and explore a variety of materials, tools and techniques, experimenting with colour, design, texture, form and function.

Being imaginative: Children use what they have learnt about media and materials in original ways, thinking about uses and purposes. They represent their own ideas, thoughts and feelings through design and technology, art, music, dance, role-play and stories.

Numbers: Children can count reliably with numbers from 1 to 20.

Space, Shape and Measure: Children use everyday language to talk about position. Children use everyday language to talk about distance.





How to use E.a.R.L 2

Getting started:

Before using E.a.R.L 2 for the first time, please fully charge by connecting E.a.R.L 2 to a USB port or USB charger, using the lead provided.

Charging:

Using the USB lead provided, connect E.a.R.L 2 to any USB port or USB charger. E.a.R.L 2's eye will light red to show it is charging. When E.a.R.L 2 is fully charged, the eye will light green. E.a.R.L 2 will be completely charged in less than 3 hours. Please note, always fully charge E.a.R.L 2 before storing for long periods.

Changing the battery:

In the unlikely event that the battery becomes defective, the battery may be accessed by removing the battery hatch using a screwdriver. The battery can then be unplugged from E.a.R.L 2 and an exact replacement fitted. Refit the battery hatch and secure using the screw.

Switching on:

There are two slide switches located next to the battery hatch. The first switch is for sound and the second for power. Normally both switches will be in the "on" position when in use, but if a number of E.a.R.L 2s are being used at the same time, the sound can be turned off by moving the sound switch to "speaker off" position. When E.a.R.L 2 is switched on, the LEDs will light for 3 seconds to indicate the remaining battery charge and E.a.R.L 2 will play a sound.

60 – 100% Green

30 – 60% Amber

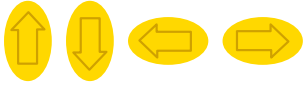





10 – 30% Red

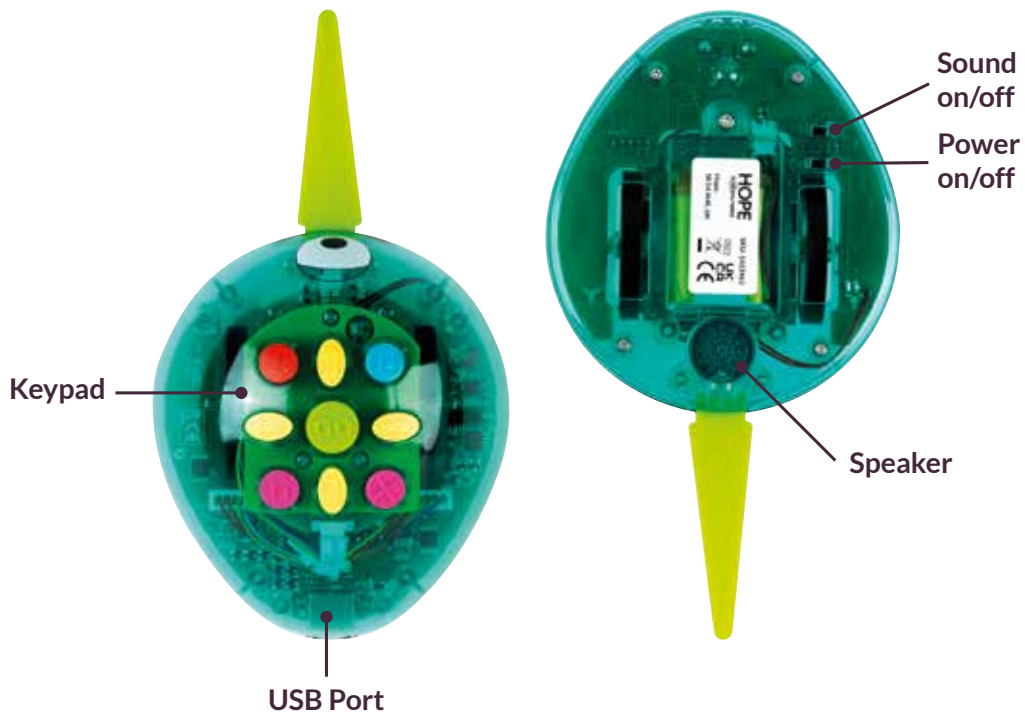
If E.a.R.L 2's battery charge is less than 10%, the eye will change to red to indicate that recharging is required.

Sleep mode:

If no buttons are pressed for 5 minutes, E.a.R.L 2 will go into a very low power sleep mode in order to conserve the battery. Pressing any button will wake E.a.R.L 2. Any programmed steps are retained when E.a.R.L 2 wakes from sleep mode.



Button	Function
	Forwards, backwards, left, right
	Clear previous program
	Pause
	Make sound (customise E.a.R.L. 2's sounds via the USB port). See page 10
	Light up
	Active program





Customising the sounds on E.a.R.L 2.

There are several sounds which are pre-loaded on to the robot that can be customised easily.



When teaching this to the children, we can explain that sound is a type of output i.e. information or power leaving a digital device in response to an action (or input).

To locate the original sounds on E.a.R.L 2:

- Connect E.a.R.L 2 to a computer using the supplied USB lead
- Once the device is installed, go to File Explorer and click on E.a.R.L 2
- You can drag and drop files onto E.a.R.L 2
- When you first open Mp3 sound in File Explorer, you will see the following pre-loaded sounds:

06 SOUND	08 SLEEP	09 START
10 END	11 MAXPROG	12 CLICK

You will need to delete the current sounds if you want to customise them.



We recommend saving a copy of the original sounds on to your computer so that you can use them again in the future.

To customise the sounds on E.a.R.L 2:

- Locate a new sound on your computer and drag it onto E.a.R.L 2 in File Explorer
- In E.a.R.L 2, edit the file name and change it to one of the names above to match that particular action
- Disconnect E.a.R.L 2 and test out the new sounds using the buttons

You can also customise other sounds such as when a particular command button is pressed. Just add the sound file to E.a.R.L 2 as above and change the name to one of the following:

01 FWD	02 REV	03 LEFT
04 RIGHT	05 PAUSE	07 LIGHT



Customisable Sounds

You could record some of the children speaking the commands and upload the files to E.a.R.L 2 so they can hear their own commands when E.a.R.L 2 is in action.



Additional E.a.R.L 2 resources to achieve curriculum objectives

Command cards

It is important for children to understand that they must first create an algorithm before turning it into a program on a digital device. The command cards allow them to do this easily and they are a fantastic visual and kinaesthetic tool. For example, they can create an algorithm to get from one place to the other using the cards before they check it is correct using E.a.R.L 2.

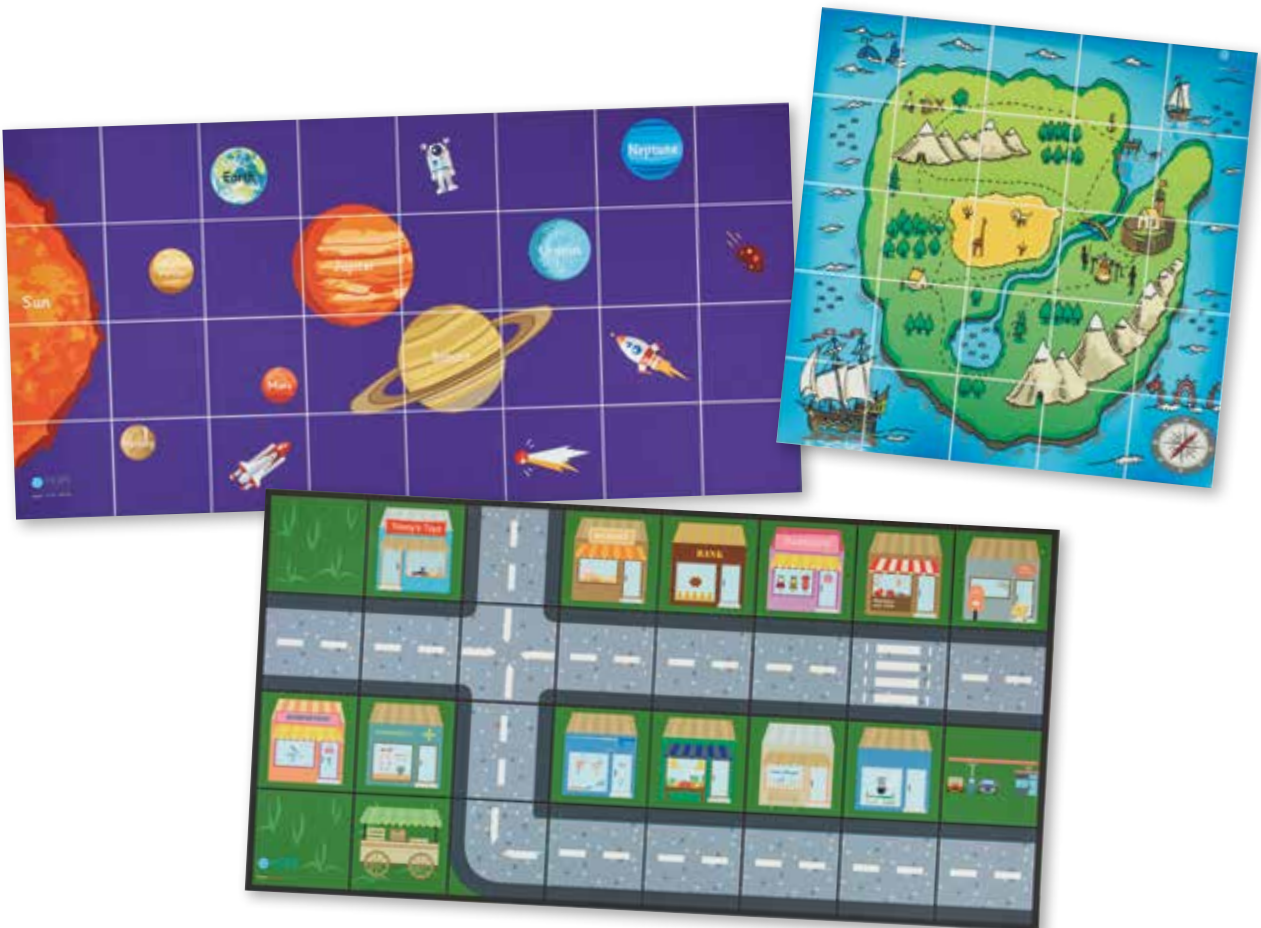


The children's logical reasoning skills could be developed further by simply placing the 'clear' card at the start, and the 'go' button at the end of their algorithm and creating a sequence of cards at random in between. They then need to predict the final destination before testing it using E.a.R.L 2.

Mats

There are a number of different mats available to enrich the cross-curricular learning children can have when using E.a.R.L 2.

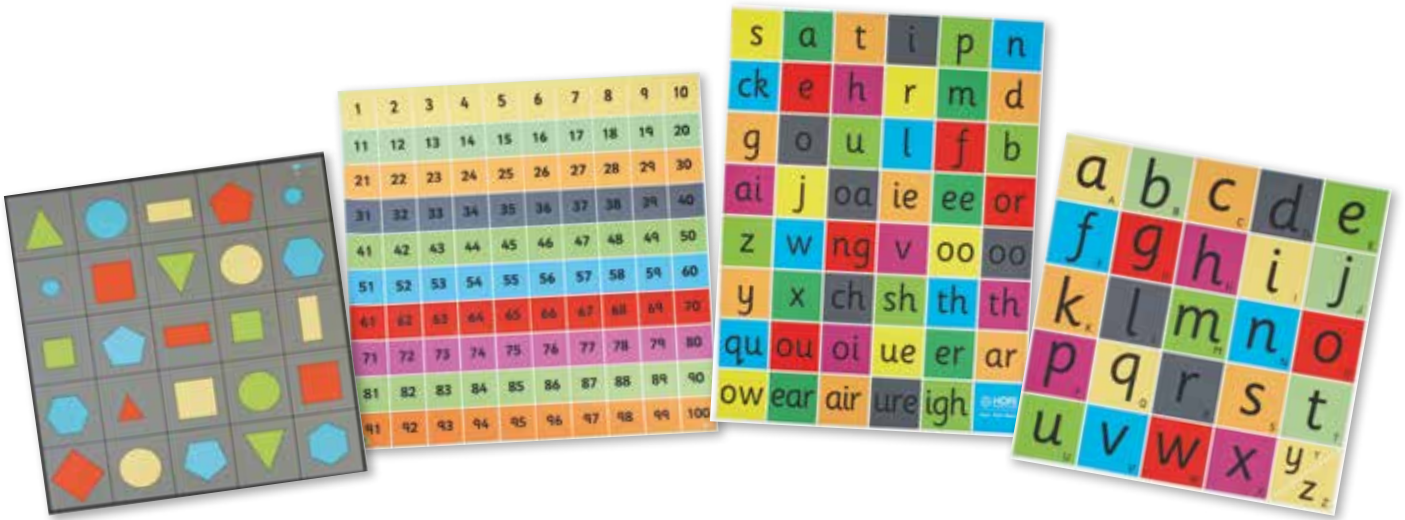
Children can learn programming and coding in a fun way whilst learning other subjects from the solar system to Treasure Island.





Challenge cards linked to the mats are available on page 28-31. These provide opportunities for a range of cross-curricular activities and allow the children to take ownership of their learning.

E.a.R.L 2's English and maths mats are a great resource to encourage active learning of phonics, numbers and shapes.



E.a.R.L 2 game mats inspire children to build their own coding journey.





Programming E.a.R.L 2 using Scratch

Scratch is a project of the Lifelong Kindergarten Group at the MIT Media Lab. It is provided free of charge. To begin using Scratch, just follow these simple steps:

- 1) Download the free Scratch 2.0 offline editor from <https://scratch.mit.edu/download/scratch2> your PC or network.
- 2) Follow the instructions and install Scratch 2.0.
- 3) To download E.a.R.L 2 coding blocks, choose the relevant E.a.R.L 2 Scratch Helper and save it to your PC or network. These can be found on the Hope or Findel website.

UK: www.hope-education.co.uk/earl

International: www.findel-international.com/earl

- 4) Switch E.a.R.L 2 on, then using the USB lead provided, connect E.a.R.L 2 to a USB port on your PC. Press and hold the “GO” button for 5 seconds. The LEDs will flash blue, E.a.R.L 2 will be detected by your computer, and the necessary driver will be loaded automatically. Please note, it may be necessary to restart your PC.



All the necessary software is now installed on your computer.

- 5) Switch your E.a.R.L 2 on and connect to a USB port on your computer. Press and hold the “GO” button for 5 seconds until E.a.R.L 2’s LEDs flash blue, signalling E.a.R.L 2 is now in Scratch mode.
- 6) Double click the E.a.R.L 2 Scratch Helper icon to launch it. Check there is a tick under the image of E.a.R.L 2 to show that it is connected. If the tick isn’t present, check the USB connection.
- 7) Click on “Start Scratch” and the E.a.R.L 2 Scratch Helper will launch Scratch for you. Just click on “More Blocks” and you will find the custom blocks needed to control E.a.R.L 2. You can use these blocks to build your program.



You must use an “Event” to start your program. When you have finished creating your program, simply trigger the event you have chosen. In this example, click on the green flag to send the code to E.a.R.L 2. E.a.R.L 2 will make a sound to confirm the program has transferred. You can now unplug E.a.R.L 2 from your PC and press “GO” to run the program.



Please note that the motors are disabled when E.a.R.L 2 is connected to a computer.



For more ideas, download the free learning resources and getting started guides at:

UK: www.hope-education.co.uk/earl

International: www.fndel-international.com/earl





Computer Science Terminology

Term	Definition	Out of computing context examples
Abstraction	Using abstraction, we focus upon the details that matter whilst also removing any unnecessary detail.	If we were creating a dance sequence with the children, we could agree on several steps and then repeat the sequence for the duration of the song. This simplifies the dance and removes unnecessary detail.
Algorithm	An algorithm is a set of rules or instructions to complete a task. It is a series of simple steps that anyone can follow to solve a problem. In computing, we turn an algorithm into a program on a digital device using a specific programming language which the computer can understand.	Children use algorithms every day to complete specific tasks. For example, when they get dressed or eat their breakfast in the morning they follow an algorithm to do it correctly.
Bug	A bug is an error in an algorithm which prevents it from working exactly as we would like.	When children learn to brush their teeth, they follow an algorithm. If the steps are not in the correct sequence they will not brush their teeth correctly. e.g. If they put toothpaste on after they have brushed their teeth.
Code	A set of instructions which is created in a language the computer understands to solve a problem.	n/a
Computational thinking	Computational thinking forms the basis for much of computer science and a key skill for children to be taught. It is a systematic approach to problem solving using logical and critical thinking skills that both humans and computers can understand. To develop 'computational thinking' in our pupils, we can use a range of approaches to address 4 key concepts: 1. Algorithms 2. Decomposition 3. Pattern recognition 4. Abstraction	Children are often presented with scenarios in a range of subjects that challenge their thinking. For example, in science they may be given a question which needs to be investigated and they will need to use computational thinking to solve it.
Data	Information that is stored in or used by a computer.	Children can create bar charts in maths/science to present data.
Debugging	Debugging is the process we need to follow to locate and fix a bug.	If children are not brushing their teeth correctly (see bug example) they may need to move the sequence of steps around so that they are in the correct order.
Decomposition	When we break down a problem into simple and easy to manage parts, this is known as decomposition . Decomposition helps to solve complex problems and manage large projects.	Pupils use decomposition in many subjects. For example, if they are going to produce a piece of writing, they would first look at different examples of the particular genre and pick out the main features. In breaking a piece of writing up into smaller parts and focusing on each feature individually, it makes it easier to produce a longer piece of writing.
Hardware	The physical components of a computer or digital device.	The children will have used a range of hardware including a keyboard on a desktop computer.



Computer Science Terminology (continued)

Term	Definition	Out of computing context examples
Logical reasoning	Applying rules to problem solving.	When children spell unfamiliar words they could use their knowledge of similar sounding words and patterns to spell it correctly.
Pattern recognition	Problems often come in different shapes and sizes. Among the many differences, there are often similarities. When we look for patterns, we often recognise what different problems have in common and can often use what has worked before to help us out again.	By identifying patterns, we can make predictions, create rules and solve more general problems. In English and maths, children often improve their spellings and develop a quicker recall of the time tables using this process.
Sequence	Presenting the steps in the correct order.	The lunch time routine in school follows a sequence. If it is not done in the correct order things will go wrong.
Software	A collection of data or instructions that tell a computer how to work.	When children use apps on an iPad they are using software that runs on a computer for a specific purpose.



Unplugged – an introduction to programming

We can introduce computer science concepts, such as algorithms and debugging from a young age without giving access to a computer, robots or any type of technology.

The use of **unplugged activities** are a great way to introduce the basic concepts of programming and how we need to test to find any bugs and then follow the process of debugging to fix them.

Following the three introductory lessons, pupils will have experienced some key **computational thinking** concepts and approaches including algorithms, decomposition, debugging, perseverance and collaboration.

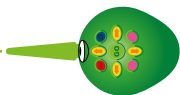
Unplugged Activities

Unplugged activities are used to introduce pupils to a range of computer science concepts, challenging them to solve problems logically without using computers, robots or technology.

Computational thinking

Facing a problem so that it can be modelled and solved by a computer. Made up of four key strategies:

- Decomposition
- Pattern matching
- Abstraction
- Algorithms





Lesson 1

Unplugged: an introduction to programming

Lesson objective: I can follow an algorithm accurately

Computing curriculum objectives:

Understand what algorithms are

Cross-curricular objectives:

Maths, Year 1:

- Recognise and name common 2-D shapes, including squares, circles and triangles

Resources:

- A3 paper
- A4 paper
- Pencils
- Simple drawing steps (Appendix 1)
- (Optional) plastic 2D shapes

Preparation:

- Using the activity instructions (Appendix 1) draw the picture on A3 paper - do not show this to the class until they have followed the instructions and drawn their own



You could make your own simple picture maybe something linked to the class topic or text and make your own list of simple steps. Remember to be vague, avoiding prepositions or details such as size and shape

- Give out A4 paper and a pencil to each child

Activities:

- Ask the children if they know what the word 'algorithm' means
- Tell the children that you will be sharing an algorithm with them and once the task is complete, they should be able to explain what the word means
- Read out the list of steps needed to draw the picture, pausing in between each one. Try to use the word 'algorithm' wherever possible so the pupils become familiar with the term
- Emphasise to the children that they should do exactly what they are asked to do and they cannot ask any questions
- Once you have finished ask the children to compare their drawings with each other
- Ask the children if they to explain what an 'algorithm' is
- Introduce and define the term 'algorithm'. (Refer to Computing Curriculum Terminology page 15-16)
- Compare all drawings with the original drawing
 - Questioning:-
 - What do you notice? (They should all look different)
 - Why are they different? (Because the algorithm wasn't very good and needed more detail)
 - How could we improve our algorithm? (Add more detail such as size and where things should be positioned)
 - Introduce and explain the term 'bug' and 'debugging'. (Refer to Computing Curriculum Terminology page 15-16)

Timings: 40 minutes

Extension:

- Ask the children to draw their own character and describe it to their partner
- Can the other child give feedback on how to improve the algorithm?

SEN/additional support:

- Prior to the activity, show children the different shapes before they listen and follow the algorithm. Children could use plastic 2D shapes if required to create the picture

Plenary:

- Watch BBC Bitesize video 'What is an algorithm?'
<https://www.bbc.com/bitesize/articles/z3whpv4>
- Key questions:
 - Can I describe to my partner what an algorithm is?
 - What do we need to consider when creating an algorithm?



Lesson 2

Unplugged: an introduction to programming

Lesson objective: I can take on the role of a robot and follow an algorithm accurately

Computing curriculum objectives:

KS1

- Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
- Create and debug simple programs

Cross-curricular objectives:

Maths, Year 1:

- Describe position, direction and movement, including whole, half, quarter and three quarter turns

Maths, Year 2

- Use mathematical vocabulary to describe position, direction and movement, including movement in a straight line and distinguishing between rotation as a turn and in terms of right angles for quarter, half and three-quarter turns (clockwise and anticlockwise)

Resources:

- Playground chalk
- Metre sticks or large ruler
- Whiteboard and pens
- Masking tape (optional)

Preparation:

- Outside, using chalk and a large ruler/metre stick draw several simple mazes on the playground for the children to be able to walk through
- Begin with a simple challenge, such as making one turn and increase the difficulty e.g. 2 turns etc

Tip: If it is wet outside, you can follow the same procedure using paper tissue or masking tape in school.

Activities:

- Remind the children of what the term 'algorithm' means. (Refer to Terminology page 15-16)
- Ask the children what happens if the algorithm does not include sufficient detail or is not accurate
- Watch the BBC Bitesize video 'What are computer bugs?'
<https://www.bbc.com/bitesize/articles/ztgjq6f>
- Discuss what 'bugs' are and revisit the term 'debugging'. (Refer to Terminology page 15-16)
- Tell the children that they will be turning into robots and will be given an algorithm to follow
- Emphasise that computers do not have brains, cannot think for themselves and will only do exactly what we tell them to
- Ask one of the children to be the robot. Then ask the children to shout out the steps they have written but not to correct any mistakes (bugs). Show the rest of the group/ class one of the mazes. Ask them to use their whiteboards to create an algorithm to navigate the robot through the maze
- Position the robot at the start of the maze. Choose one child to share their algorithm step by step for the robot to follow (this child should look away from the maze so they cannot correct any mistakes (bugs))
- Once the algorithm is complete, discuss what happened with the rest of the group/ class
 - Did you spot any bugs?
 - Could you debug them?
- Ask the children to navigate through the different mazes in pairs or small groups

Timings: 40 minutes

Extension:

- Can they create their own mazes and suitable algorithms?
- Can they create their own command cards to get them through the maze? e.g. 'forward one step', 'one right turn' etc
- Use Key Stage 2 vocabulary for turning such as rotation, right angles, clockwise and anti-clockwise

SEN/additional support:

- Provide a list of command cards that could be used in an algorithm e.g. move forward 1 step. This could be created with the children as they discuss the most common commands

Lesson 2 continued on next page

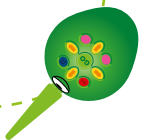
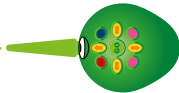


SEN/additional support:

- Tap the child on the right or left to help them turn the correct way

Plenary:

- Key questions:
 - What happens when the algorithm is not accurate?
 - How could we make sure it is accurate?
- Discuss testing the algorithm before we pass it to the robot. Would this help? What else may help us when we are testing it? Discuss breaking the algorithm down into smaller parts to make it easier. This introduces the concept of '**decomposition**'. (Refer to Terminology page 15-16)





Lesson 3

Unplugged: an introduction to programming

Lesson objective: I can create an accurate algorithm

Computing curriculum links:

Key stage 1

- Understand what algorithms are
- Programs execute by following precise and unambiguous instructions
- Create and debug simple programs

Key stage 2

- Solve problems by decomposing them into smaller parts

Cross-curricular objectives:

English

- Listen and respond appropriately to adults and their peers

Resources:

- Maze resource (Appendix 2 and 2.1)
- Pencils

Preparation:

- Print maze resource (Appendix 2 and 2.1) and hand out 1 per pair
- Ask children to sit in pairs

Activities:

- Remind the children of what the terms 'algorithm' and 'debugging' mean. (Refer to Terminology page 15-16)
- Tell the children to work in pairs to navigate around the paper-based maze
- Child 1 is to be the robot. They must close their eyes and hold the pencil on the paper ready to receive the algorithm
- Child 2 gives instructions to guide Child 1 through the maze
- Once complete, children swap roles and repeat

Timings: 40
minutes

Support:

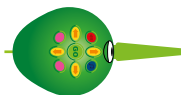
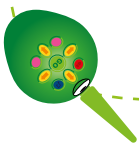
- Use simpler mazes.

Extension:

- Draw their own mazes to challenge other children
- Challenge the children to use less steps in their algorithm

Plenary:

- Discuss what happened with the rest of the group/ class
 - Did they successfully navigate through the maze?
 - What was challenging?
 - How could we have made it easier?
- Discuss testing the algorithm before it is passed to the robot. Would this help?
- What else may help us when we are testing it? Discuss breaking the algorithm down into smaller parts to make it easier. Could we have had checkpoints? This introduces the children to the concept of 'decomposition' which is taught in Key Stage 2





Introducing E.a.R.L 2

Integrating programming and robotics

Children will now have a basic understanding of some key computational thinking concepts. They will be aware of what an **algorithm** is, how they need to test it to find any **bugs** and then follow the process of **debugging** to fix them. They will also have some experience of using **decomposition** to break a problem down into smaller parts. The next step will be to turn an algorithm into a program on a digital device.

The opportunity to interact with E.a.R.L 2 allows children to develop their logical thinking and problem-solving skills further as they work through the following activities.





Lesson 4

Create a program for E.a.R.L 2

Lesson objective: I can give E.a.R.L 2 simple commands to follow

Computing curriculum links:

Key stage 1

- Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
- Create and debug simple programs

Cross-curricular objectives:

Maths, Year 1

- Recognise and name common 2-D including rectangles, squares, circles and triangles

Resources:

- E.a.R.L 2
- Large sheet of paper
- Pencils

Preparation:

- Fully charge E.a.R.L 2 prior to the activities
- Print 'How to use E.a.R.L 2' (Page 8)

Activities:

- Introduce the children to E.a.R.L 2
- Begin by looking at E.a.R.L 2's key features together – the clear outer shell, wheels, switches and buttons. Ask children to discuss and describe the features and what they might do
- Tell the children E.a.R.L 2 is a robot. Explain that a robot cannot think for itself and that it needs a human to create a program to make it move, sound or interact
- Explain that we can create a program by using the different coloured buttons
- With E.a.R.L 2 switched on, give the children time to explore the robot and its different functions
- Look for opportunities to discuss what each button does. Using the clear button to reset E.a.R.L 2 and make him start again
- Challenge the children to program E.a.R.L 2 to travel a short distance from one part of the room to the other
- Place start and end cones/objects on the floor
- Discuss whether the programming was successful

Timings: 40 minutes

Support:

- Ensure children understand how to make E.a.R.L 2 move in the four basic directions. (forwards, backwards, left and right). Place 4 objects on the floor that E.a.R.L 2 needs to travel to, from the same starting point

Extension:

- I can create a sequence of commands to make E.a.R.L 2 travel different distances in the classroom
- I can make E.a.R.L 2 take a pause and play a sound at different points on his journey

Plenary:

- Discuss E.a.R.L 2's features
- Discuss what might be helpful to do before we program E.a.R.L 2. (Create an algorithm)
- Discuss the importance of testing parts of the program (decomposition) and the whole program (debugging) (Refer to Terminology page 15-16)





Lesson 5

E.a.R.L 2 Challenges

Lesson objective: I can program E.a.R.L 2 to complete a series of challenges

Computing curriculum objectives:

Key stage 1

Pupils should be taught to:

- Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
- Create and debug simple programs
- Use logical reasoning to predict the behaviour of simple programs

Resources:

- Print E.a.R.L 2 challenge cards (Pages 25-26)
- Command cards
- Whiteboard and dry wipe pens (Extension option)

Preparation:

- Fully charge E.a.R.L 2 prior to the activities
- Refer to explanation for command cards (Using commands cards page 11)

Activities:

- Ask children to work in pairs or small groups. Ask the children to select a challenge card and use E.a.R.L 2 to complete the challenge
- When the children have completed the first challenge, introduce the command cards
- Explain how using the cards to create an algorithm can help to debug a program. (see page 13)

Approximately 20 minutes per card. This will vary depending on the children's ability

Support:

- An adult may be needed to support the children in reading the activities and to ensure they understand the task

Extension:

- I can create my own challenges for others to complete
- Children can use a whiteboard to create their algorithms instead of the command cards

Plenary:

- Can the children use logical reasoning to predict the behaviour of simple programs by predicting what E.a.R.L 2. will do based on other children's algorithms?
- Ask the children to share how they solved the challenges with others





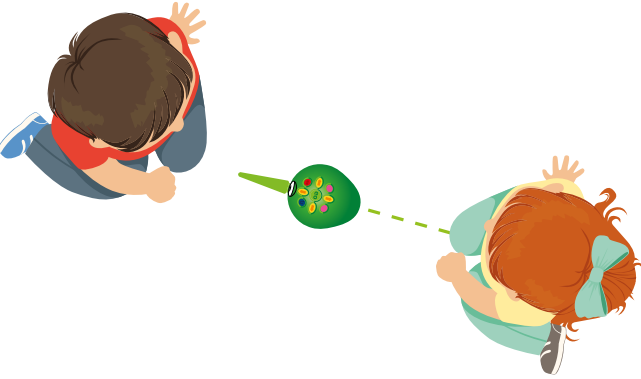
Challenge 1

Objective: I can make E.a.R.L. 2 travel to my partner

Task:

Sit facing your partner.

Take it in turns to program E.a.R.L. 2 to reach the other person.



Extra challenge:

I can make E.a.R.L. 2 visit my partner, make a sound, then turn around and come back to me.

I can connect E.a.R.L. 2 to Scratch to create a program using E.a.R.L. 2 coding blocks.



Challenge 2

Objective: I can measure the distance E.a.R.L. 2 travels when I press the forward button once

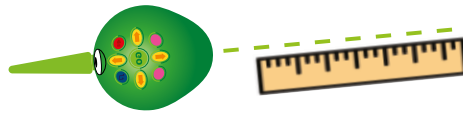
Task:

Discuss with your partner how you can measure the distance E.a.R.L. 2 travels when the forward button is pressed once.

With your partner talk about the materials you will need.

Work with your partner to collect and set up and complete the challenge.

Record your answer.



Find materials that will make your measurements as accurate as possible.

Extra challenge:

How far will E.a.R.L. 2 travel when you press the forward button 2 or 3 times?



Challenge 3

Objective: I can make E.a.R.L. 2 complete a short sequence of moves

Task:

Make E.a.R.L. 2 complete a dance routine using all the buttons at least once.

Create a sequence of moves and repeat it 3 times.



Using the command cards will help you to plan your sequence and make sure you do not miss any steps.

Extra challenge:

Could you ask another pair to predict what dance E.a.R.L. 2 will do and then program their E.a.R.L. 2 to do the same the dance at the same time?

Try making this dance in Scratch



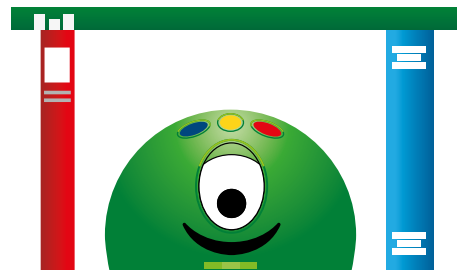
Challenge 4

Objective: I can make a new home for E.a.R.L. 2

Task:

Choose materials available in your classroom to make a good home for E.a.R.L. 2.

Build the home then program E.a.R.L. 2 to enter it from different areas in the classroom.



Your home will need to have an entrance wide enough for E.a.R.L. 2 to enter.

Extra challenge:

Can you program E.a.R.L. 2 to go into the home and then go back to the starting point?



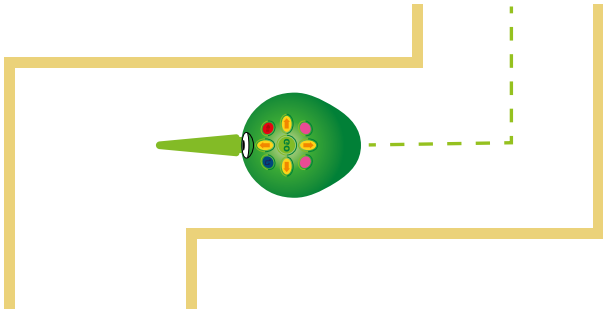
Challenge 5

Objective: I can make E.a.R.L. 2 follow a path through a maze

Task:

Use masking tape (inside) or playground chalk (outside) create a path/maze.

Program E.a.R.L. 2 to travel through the maze.



Don't forget to make your path wide enough for E.a.R.L. 2 to travel through.

Extra challenge:

Can you make E.a.R.L. 2 pause and play a sound at different parts of the maze?



Challenge 6

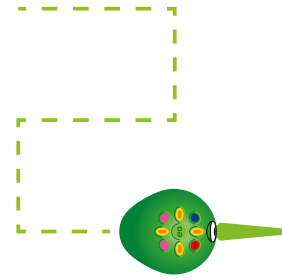
Objective: I can make E.a.R.L. 2 complete a number formation

Resource:

Print digital numbers (Appendix 3).

Task:

Program E.a.R.L. 2 to follow path along the shape of a digital number.



Don't forget to make your path wide enough for E.a.R.L. 2 to travel through.

Extra challenge:

Can you help E.a.R.L. 2 spell a simple 3 or 4 letter word, e.g. cat.



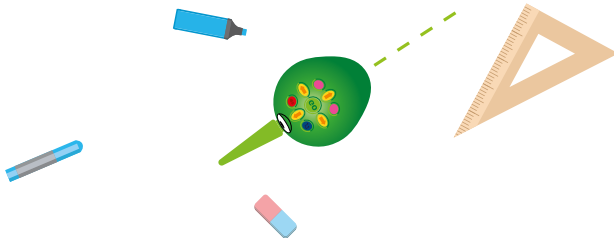
Challenge 7

Objective: I can program E.a.R.L. 2 to travel around different shaped objects

Task:

Choose some different shaped objects and place them in a clear space on the carpet.

Program E.a.R.L. 2 to travel around the sides of the objects.



Think about how far E.a.R.L. 2 will move when a yellow command button is pressed once.

Extra challenge:

Can you find examples of 2D shapes where commands are repeated to complete the shape?

Program E.a.R.L. 2 to turn around and repeat the journey the opposite way.



Challenge 8

Objective: Which surface does E.a.R.L. 2. travel the furthest on?

Task:

Test how far E.a.R.L. 2 will travel on different surfaces.

Can you predict which surface E.a.R.L. 2 will travel the most and the least on?

Think about how you will measure the distance travelled accurately.

Program E.a.R.L. 2 to move forward 5 times and measure the distance he has travelled along each surface using standard units.



Make sure you set up a fair test using different surfaces e.g. carpet, plastic, wood, thin wire mesh.

Extra challenge:

Can you make a list to order the distances that E.a.R.L. 2 travels on each surface?

Could you make a graph?

Explain why E.a.R.L. 2 doesn't travel the same distance on each surface.



Lesson 6

E.a.R.L 2 Mat Challenges

Lesson objective: I can program E.a.R.L to navigate around a mat to complete a set challenges

Computing curriculum links:

- Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
- Create and debug simple programs
- Use logical reasoning to predict the behaviour of simple programs

Cross curricular links:

The following challenges support a number of cross-curricular learning objectives

English:

- Ask relevant questions to extend their understanding and knowledge
- Articulate and justify answers, arguments and opinions participate in discussions

Treasure Island Mat Challenges:

Raises awareness of conservation issues which could lead on to wider reading and discussions in English and science.

Raises awareness of deforestation issues and losses of habitat which could lead on to wider reading and discussions in English and science.

Raises awareness of storm damage and supporting charities which could lead on to wider reading and discussions in English.

Street Scene Mat Challenges:

Raises awareness of supporting local businesses and buying fair trade products which could lead on to wider reading and discussions in English.

Solar system Mat Challenges:

In geography, KS1 pupils should be taught to:

- Use simple compass directions (North, South, East and West) and locational and directional language [for example, near and far; left and right], to describe the location of features and routes on a map

In geography, KS2 pupils should be taught to:

- Use grid references

In English, Year 1 pupils should be taught:

- The name the letters of the alphabet
- Naming the letters of the alphabet in order

Resources:

- E.a.R.L 2
- Treasure Island Mat/Street Scene Mat/Solar System
- Command cards
- Challenge cards
- Whiteboards and pens

Preparation:

- Place the mat flat down in an area
- Print and cut challenge cards

Activities:

- The children should now be comfortable creating algorithms using the command cards or on whiteboards before turning into a program on a digital device
- Give the children time to familiarise themselves with chosen mat
- Tell them they will be given a number of different activities and they have to program E.a.R.L 2 to complete the tasks

Timings:
Approximately
15 minutes
per challenge.
This could vary
depending on a
child's ability

Support:

- Provide command cards for the children to create their algorithm before programming E.a.R.L 2

Extension:

- Ask children to create their own challenges for others to complete
- Children can use their own symbols to create their algorithms on whiteboards instead of the command cards

Plenary:

- Can the children use logical reasoning to predict the behaviour of simple programs by predicting what E.a.R.L 2 will do based on other children's algorithms?
- Ask the children to share how they solved the challenges with others



Treasure Island: Challenge 1

Challenge: Help the sailors travel around the coast

The sailors have lost their map and need your help to sail all the way around the coast safely. Start where the ship is in the bottom left corner and lead it all of the way around the island and back to the starting position.



Extra challenge:

The sailors have had a long journey and are very hungry. Can you make E.a.R.L. 2 pause on different parts of the journey so that they can get something to eat and have a rest?



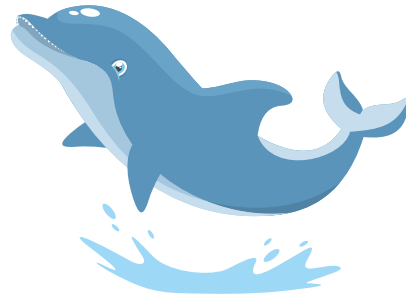
Treasure Island: Challenge 2

Challenge: Help the dolphin get back out to sea

A dolphin is lost and has managed to find its way into the lake. It needs to get back to the other dolphins in the sea as soon as possible.

Can E.a.R.L. 2 help the dolphin find her way out through the river?

Place E.a.R.L. 2 in the lake and create a program to lead the dolphin out through the river and back into the sea.



Extra challenge:

Can E.a.R.L. 2 travel from the sea into the river to collect the dolphin and then take it back to the sea to escape?



Treasure Island: Challenge 3

Challenge: Save the trees

A pirate has been chopping down lots of trees to build a new ship.

Choose a starting point then program E.a.R.L. 2 to visit all of the trees on the island to make sure they haven't been taken.



Extra challenge:

Try to visit all of them in the quickest possible time.



Treasure Island: Challenge 4

Challenge: Find the buried treasure

There has been a storm and the bridge has been damaged.

Starting at the wooden fort, can you program E.a.R.L. 2 to travel around the river and lake to find the buried treasure.



X marks the spot!

Extra challenge:

Collect the treasure then program E.a.R.L. 2 to take it back to the fort.

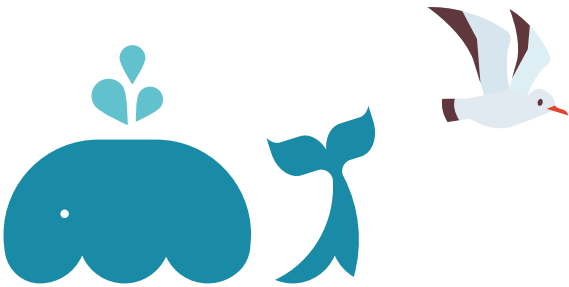


Treasure Island: Challenge 5

Challenge: Meet and greet

The pirates would like to build a home on the island but would like to know how many animals and people live there.

Program E.a.R.L.2 to visit all of the people and animals on the island and pause to say hello to each one.



Extra challenge:

Can you find the best starting point to find the quickest possible route?

Create a pictogram or bar chart displaying some of the features on the island e.g. mountains, ships, fish, people and trees.





Street Scene: Challenge 1

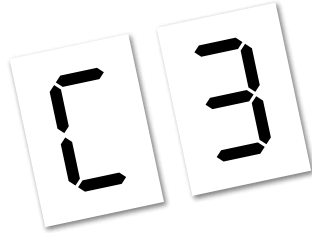
Challenge: I can use grid references accurately

E.a.R.L 2 has just arrived in a new city and doesn't know which way to go. Can you help E.a.R.L 2 get to the correct place?

To choose a location to go, select a card from each of the piles (1 x letter and 1 x number) to generate a grid reference.

Preparation:

Place grid reference labels around the outside of the mat (Appendix 4). Print out the letters A to D and numbers 1 to 8, cut them out and place them in 2 piles face down (Appendix 3).



The police are very strict in the new city and robots are only allowed to travel on the roads but they can drive into the shops when they get there.

Extra challenge:

Race another E.a.R.L 2 from different starting points to see who arrives at the destination first.



Street Scene: Challenge 2

Challenge: I can follow an algorithm to predict where E.a.R.L 2 will visit

Place the 'clear' card at the start and the 'go' card at the end in front of you, leave a space in between them.

Juggle the command cards and choose 3 cards at random.

Place them next to each other in the space between 'clear' and 'go'.

Can you predict where E.a.R.L 2 will travel to?

Program E.a.R.L 2 to see if you are correct.



Extra challenge:

Create an algorithm with 4, 5 and 6 cards and repeat the task.



Street Scene: Challenge 3

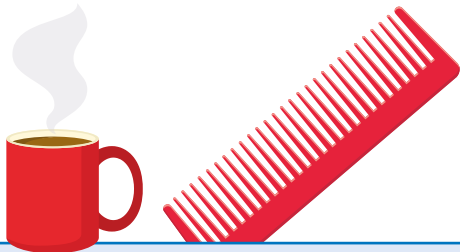
Challenge: Get E.a.R.L 2 ready for the party

E.a.R.L 2 has been invited to a party.

E.a.R.L 2 has just finished lunch at the café.

Then needs a haircut, some new clothes and wants to buy a present.

Program E.a.R.L 2 to visit all the places and get party ready.



Try to choose a good strategy to visit all of them in the quickest possible time.

Extra challenge:

Where else might E.a.R.L 2 need to go before the party? Add it to your journey.



Street Scene: Challenge 4

Challenge: Help E.a.R.L 2 collect the shopping

E.a.R.L 2's mum has given him a shopping list of items to collect.

Can you help E.a.R.L 2 to collect all the items on the list?



Extra challenge:

Program E.a.R.L 2 to collect all the items on the list in the quickest possible time.

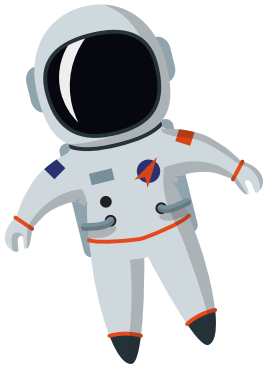


Solar System: Challenge 1

Challenge: Rescue the astronaut

An astronaut is lost in space.

Program E.a.R.L. 2 to rescue them and return to Earth.



Choose a rocket for E.a.R.L. 2 to start the rescue from.

Extra challenge:

The astronaut has left a rock sample on Mars. Take the astronaut to collect it before you travel to Earth.

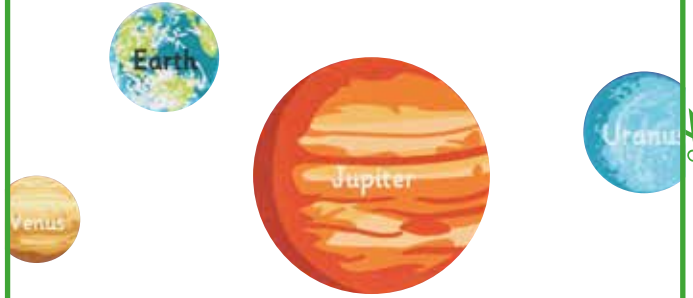


Solar System: Challenge 2

Challenge: Visit all the planets in the solar system

Starting at the Sun, E.a.R.L. 2 wants to visit all of the planets in the solar system.

Program E.a.R.L. 2 to take the journey.



Break the whole journey down into smaller parts and create separate algorithms to travel from one planet to the next.

Extra challenge:

Can you visit the planets in size order from smallest to largest.



Solar System: Challenge 3

Challenge: Visit all of the planets in alphabetical order

E.a.R.L. 2 has been learning the alphabet at robot school.

Starting from Earth can you program E.a.R.L. 2 to visit all of the planets in alphabetical order.



Work out the order of the planets before you create the algorithm.

Support:

Provide an alphabet strip.

Extra challenge:

Can you play a sound when E.a.R.L. 2 reaches each planet?



Solar System: Challenge 4

Preparation:

Print out several copies of the compass points North, South, East and West and place them face down next to the mat (Appendix 5).

Challenge: Use compass points accurately

Teach E.a.R.L. 2 how to travel in different directions using compass points.

Place E.a.R.L. 2 on the planet Saturn.

Pick up a compass point card.

Program E.a.R.L. 2 to travel in that direction and stop at the first thing he reaches.



Extra challenge:

Can you play a sound when E.a.R.L. 2 reaches a planet/object?



Lesson 7

Using E.a.R.L 2 with Scratch

Lesson objective: I can use Scratch to create a program to make E.a.R.L 2 travel in a right angle

Computing curriculum links:

Key Stage 2

Pupils should be taught to:

- 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

Cross-curricular links:

Maths, Year 3:

- Recognise angles as a property of shape or a description of a turn
- Identify right angles

Resources:

- E.a.R.L 2
- Playground chalk (Support option)
- Computer with Scratch software

Preparation:

- Pupils should have a good understanding of a right angle
- Pupils should have experience of dragging blocks in Scratch 2 to code E.a.R.L 2
- Print/read Scratch instructions page 13-14

Activities:

E.a.R.L 2's advanced features shows specific angle turns with a default set at 90 degrees, ask the children to test turns of various degrees, what do they notice? Are they still performing a right angle turn when the degree amount is greater or less than 90?

Timings:
45
minutes

Support:

- Using chalk in the playground, draw 2 straight lines that meet at a right angle and ask the children to walk along it. Can they create an algorithm for their journey? Use this to help them create their code in Scratch

Extension:

- I can create different sized right angles?

Plenary:

- What shapes are the children aware of that have right angles in them? e.g. square, rectangle
- Can they create an algorithm to move along that shape?



Lesson 8

Using E.a.R.L 2 with Scratch 2

Lesson objective: I can use Scratch 2 to create a program to make E.a.R.L 2 travel a quarter turn, half turn and whole turn

Computing curriculum links:

Key Stage 2

Pupils should be taught to:

- 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

Cross-curricular links:

Maths, Year 3:

- Recognise angles as a property of shape or a description of a turn
- Identify right angles, recognise that two right angles make a half-turn, three make three quarters of a turn and four a complete turn;

Resources:

- E.a.R.L 2
- Computer with Scratch software

Preparation:

- Pupils should have a good understanding of a right angle and that it represents a quarter turn
- Pupils should have experience of dragging blocks in Scratch 2 to code E.a.R.L 2
- Print/read Scratch instructions page 15-16

Activities:

- Discuss the properties of a right angle and that it represents a quarter turn



- Could we use Scratch to make E.a.R.L 2. move a quarter turn?



- If 90 degrees is a quarter turn, how many degrees would E.a.R.L 2 travel through to complete a half turn? Can we use Scratch to make E.a.R.L 2. move through a half turn?



- Repeat for a three quarter turn and full turn



Timings: 60 minutes

Support:

- Ask the children to physically move in quarter, half, three quarters and full turns prior to the task
- Support with right and left directions
- Ask your children if there is a more efficient way of turning? Do we need to repeat the command? Let the children know that the white box can be edited to input whatever degree of movement we want. If 90 degree is one right angle turn, how can we work out how many degrees 2 right angle turns are?



Extension:

- Could they make it turn 720 degrees? 540 degrees etc.?

Plenary:

- How many degrees are there in a quarter, half, three quarter and full turn?



Lesson 9

Using E.a.R.L 2 with Scratch

Lesson objective: I can use Scratch to create a program with repeat loops to make E.a.R.L 2 travel a quarter turn, half turn and whole turn

Computing curriculum links:

Key Stage 2

Pupils should be taught to:

- 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
- Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information.

Cross-curricular links:

Maths, Year 3:

- Recognise angles as a property of shape or a description of a turn
- Identify right angles, recognise that two right angles make a half-turn, three make three quarters of a turn and four a complete turn;

Resources:

- E.a.R.L 2
- Computer with Scratch software

Preparation:

- Pupils should have completed the previous activity using Scratch to make E.a.R.L 2 travel in quarter, half, three quarters and full turns

Activities:

- Discuss the code needed to create each turn. It should look like below:

Quarter turn:



Half turn:



3 quarter turn:



Whole turn:



- What do the children notice about the code? (It is repeating for each turn)
- How could we do 2 full turns? (Use the rotate block 8 times)
- Discuss how this could get repetitive and we can use a repeat block to make our code simpler



- Can the children use this to create 2 full turns?



- Is this more efficient than writing 180 degrees? When might a repeat be more useful? You may wish to ask the children to test this using the Earl Light Colour blocks.

Support:

- Ask the children to use the repeat blocks to do less turns

Extension:

- How many repeats are needed to do 3,4,5 turns? Can we use our times tables to help us?

Plenary:

- When we simplify our code and remove unnecessary detail we call this **abstraction**. Where else might we use **abstraction** in real life?

Timings: 60 minutes



Lesson 10

Using E.a.R.L 2 with Scratch

Lesson objective: I can use Scratch 2 to make E.a.R.L 2 make a square shaped journey

Computing curriculum links:

Key Stage 2

Pupils should be taught to:

- 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
- Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information.

Cross-curricular links:

Maths, Year 3:

- Recognise angles as a property of shape or a description of a turn
- Identify right angles, recognise that two right angles make a half-turn, three make three quarters of a turn and four a complete turn;

Resources:

- E.a.R.L 2
- Computer with Scratch software

Preparation:

- Pupils should understand the properties of a square (4 equal sides and 4 right angles)

Activities:

- Ask the children to create an algorithm to draw a square
- Challenge them to turn their algorithm into a program on a digital device using Scratch 2 and E.a.R.L 2
- Their code should look something like this



- Discuss how we used abstraction in the previous activity. Is there a way we could simplify the code above?
- Could we then add a sound at the end to say it is completed?

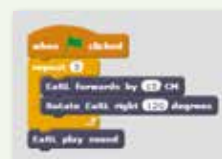


Support:

- Create the algorithm for a square together and let the children use it to create their code

Extension:

- Could we create a rectangle in this way using abstraction and repeat loops?
- Could we change the angles and turns to create other shapes? One example is an equilateral triangle (see image)



Plenary:

- Where might we see a repeat code occur in real life? Do you see any on your way to school where a machine is repeating a function? (Traffic lights)

Timings: 60 minutes



Assessment

There will be lots of opportunities to assess the children as they work collaboratively through the lessons, activities and challenges.

As well as observing them completing the tasks, it is also important to check the children's understanding and observe their approach to the 4 computational thinking concepts.

The following questions can be used to assess the children:

What is an algorithm?

An algorithm is a set of rules or instructions to complete a task. It is a series of simple steps that anyone can follow to solve a problem.

What is decomposition?

When we break down a problem into simple and easy to manage parts, this is known as decomposition.

What is pattern recognition?

Problems often come in different shapes and sizes. Among the many differences, there are often similarities. When we look for patterns, we often recognise what different problems have in common and can often use what has worked before to help us out again.

What is abstraction?

In using abstraction, we focus upon the details that matter whilst also removing any unnecessary detail.

What is the different between an algorithm and a program?

When we turn an algorithm into commands on a digital device we call this a program.

Computational thinking

Facing a problem so that it can be modelled and solved by a computer. Made up of four key strategies:

- Decomposition
- Pattern matching
- Abstraction
- Algorithms



EYFS assessment grid

E.a.R.L.2 allows children to have fun using their creativity and developing skills which hit a number of Early Learning Goal objectives. These include:

ELG: Numerical Patterns: Children at the expected level of development will:

- Verbally count beyond 20, recognising the pattern of the counting system.

ELG: People, Culture and Communities: Children at the expected level of development will:

- Describe their immediate environment using knowledge from observation, discussion, stories, non-fiction texts and maps.

ELG: Creating with Materials: Children at the expected level of development will:

- Safely use and explore a variety of materials, tools and techniques, experimenting with colour, design, texture, form and function.

Name: _____ Date _____



Activity	Success criteria	What level did the children work at in this activity?		
		WT	WA	GD
Lesson 4	<p>I can make E.a.R.L.2. move</p> <p>I can make E.a.R.L.2. light up</p> <p>I can make E.a.R.L.2. create a sound</p>			
Lesson 5 and E.a.R.L.2 challenge cards	<p>I can describe what will happen when I press the different buttons on E.a.R.L.2.</p> <p>I can make E.a.R.L.2. follow a sequence of commands to get to a chosen destination</p>			



Year 1 activities assessment grid

Pupils should be taught to:

- Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
- Create and debug simple programs
- Use logical reasoning to predict the behaviour of simple programs

Name: _____ Date _____

Activity	Success criteria	What level did the children work at in this activity?		
		WT	WA	GD
Lesson 1	I can use the word 'algorithm' accurately.			
Lesson 2	I can use the word 'debug' accurately.			
Lesson 3	I can follow and create algorithms. I can make E.a.R.L. 2. move.			
Lesson 4	I can describe what will happen when I press the different buttons on E.a.R.L. 2.			
Lesson 5	I can press the buttons in the correct sequence to I make E.a.R.L. 2 move where I want it to. I can use logical reasoning to predict what will happen when I look at an algorithm. I can debug.			





Year 2 activities assessment grid

Pupils should be taught to:

- Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
- Create and debug simple programs
- Use logical reasoning to predict the behaviour of simple programs

Name: _____ Date _____

Activity	Success criteria	What level did the children work at in this activity?		
		WT	WA	GD
Lesson 1 Lesson 2 Lesson 3	I can use the word 'algorithm' accurately? I can use the word 'debug' accurately? I can follow and create algorithms?			
Lesson 4	I can press the buttons in the correct sequence to make E.a.R.L 2 move where I want it to? I can use logical reasoning to predict what will happen when I look at an algorithm? I can debug accurately?			
Lesson 5 EaRL challenge cards	I can use logical reasoning to predict the behaviour of simple programs? I can create and debug simple programs?			
Lesson 6 EaRL mat challenge cards	I can program E.a.R.L to navigate around a mat to complete a set challenges. I can create accurate algorithms. I can use my algorithms to create programs on a digital device. I can create and debug simple programs. I can use logical reasoning to predict the behaviour of simple programs.			





Key Stage 2 activities assessment grid

In key stage 2, pupils should be taught to:

- 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
- Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information.

Name: _____ Date _____

Activity	Success criteria	What level did the children work at in this activity?		
		WT	WA	GD
Lesson 6	I can program E.a.R.L. to navigate around a mat to complete a set challenges. I can create accurate algorithms. I can use my algorithms to create programs on a digital device. I can create and debug simple programs. I can use logical reasoning to predict the behaviour of simple programs.			
Lesson 7 Lesson 8	Can I create different outputs using E.a.R.L. 2.?			
Lesson 9	Can I use block based coding to give E.a.R.L. 2. commands? Can I write and debug programs that accomplish specific goals? Can I control physical systems? Can I solve problems by decomposing them into smaller parts? Can I use sequence and repetition in programs? Can I use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs? Can I use abstraction to simplify my code?			
Lesson 10				





Appendix 1

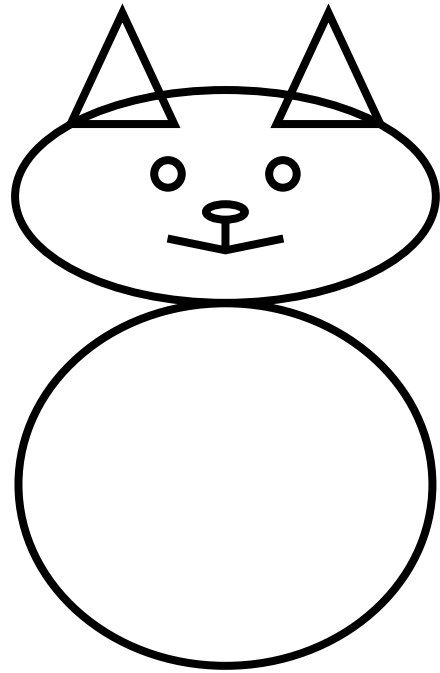
Lesson 1

Unplugged: an introduction to programming

Lesson objective: I can follow an algorithm accurately

Algorithm/drawing steps

- Draw one circle on top of the other.
- Put 2 triangles on each side of the top of the head.
- Draw two circles for the eyes.
- Draw a circle for the nose.
- Give it a smile





Appendix 2.1

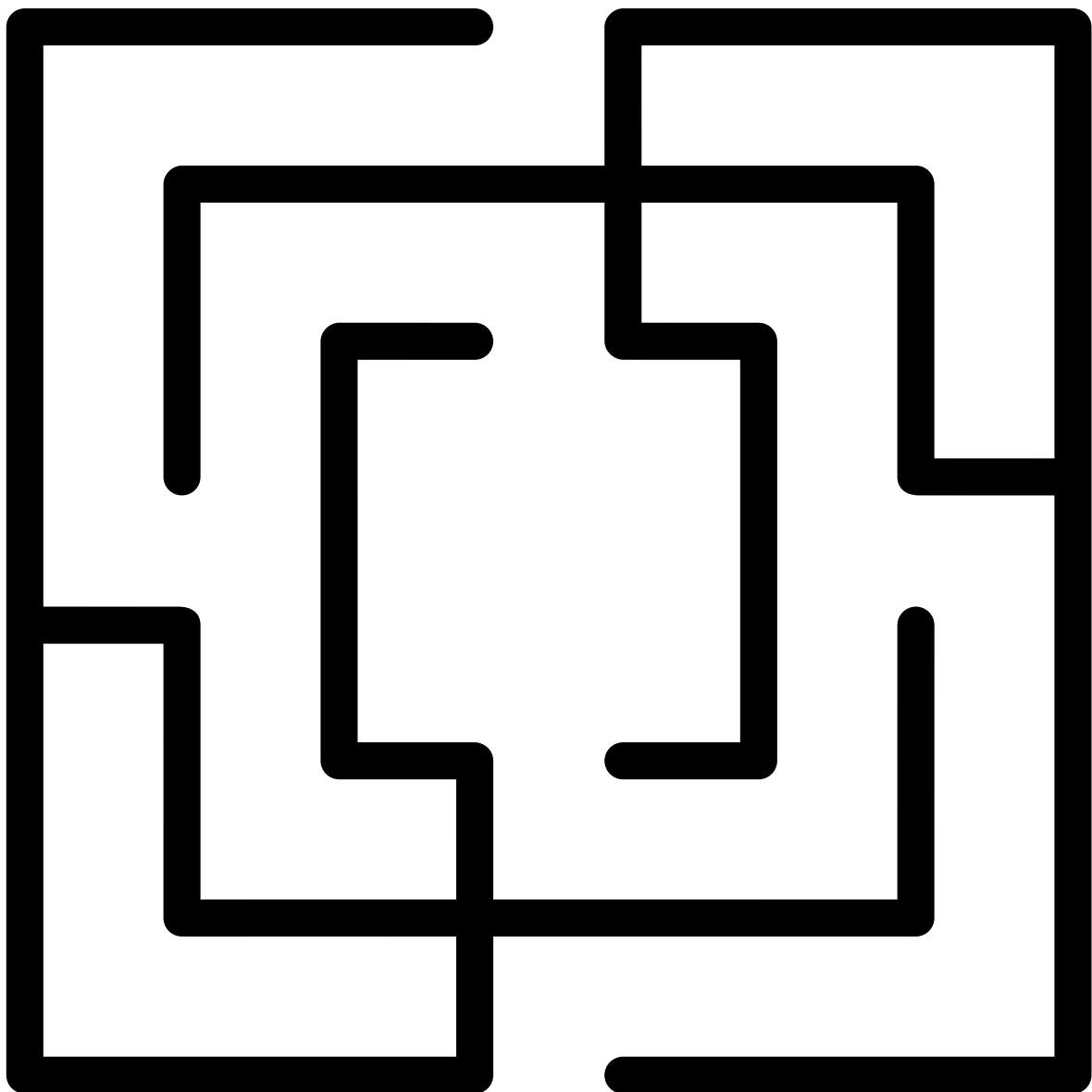
Lesson 2 - extension maze

Unplugged an introduction to programming

Lesson objective: I can take on the role of a robot and follow an algorithm accurately

Can I direct my partner out of the maze?

- Child A closes their eyes with a pencil in their hand
- Child B gives child A instruction on how to get through the maze





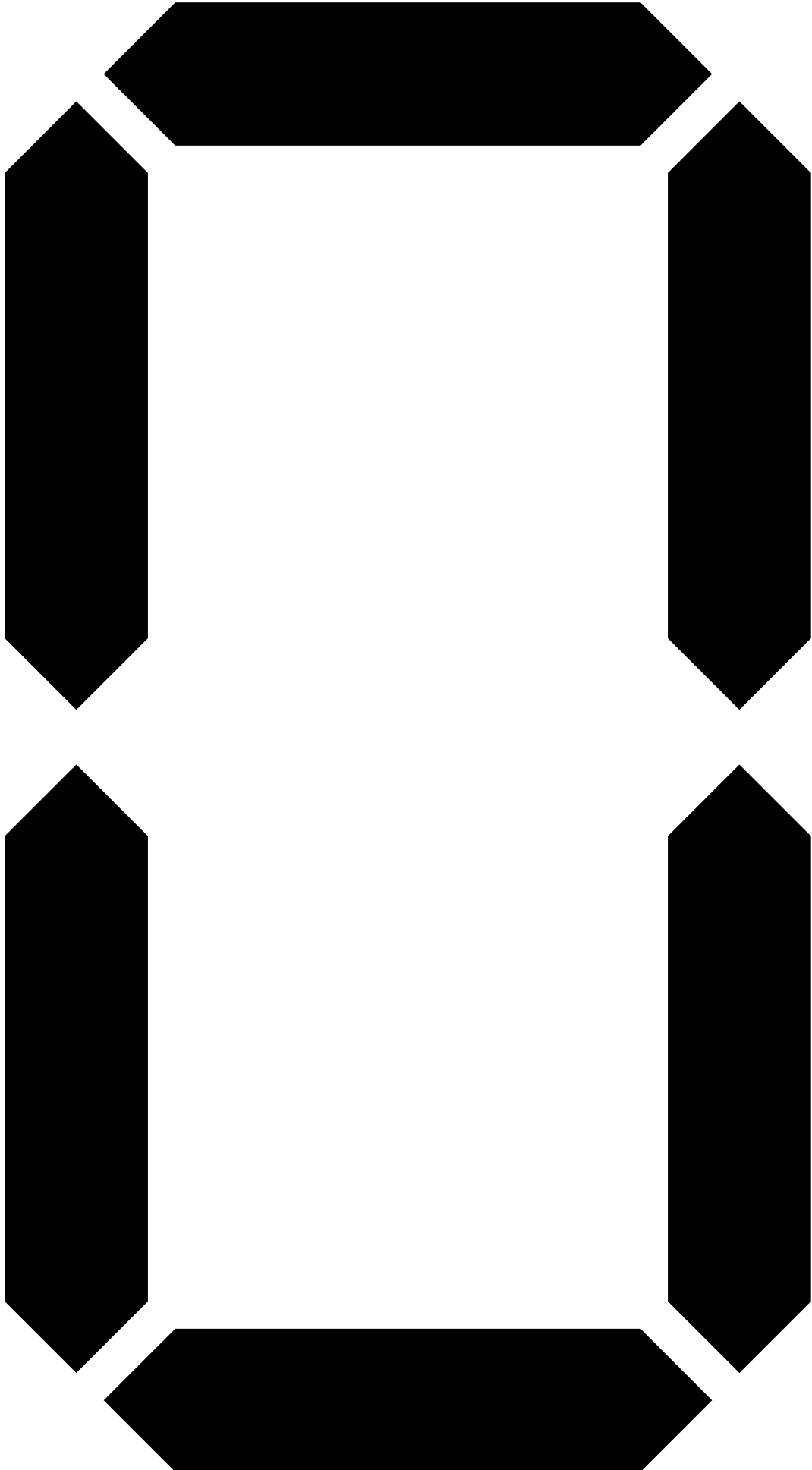
Appendix 3

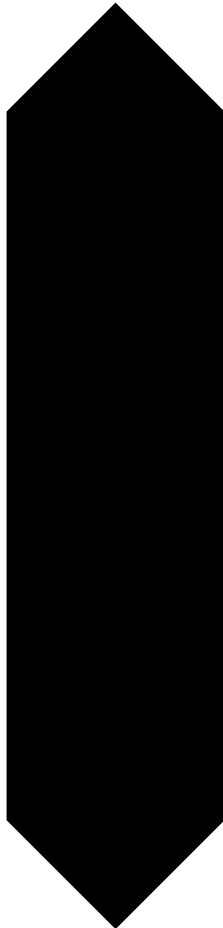
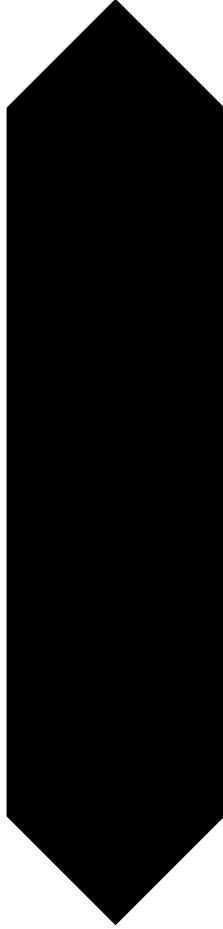
E.a.R.L 2 challenge card 6

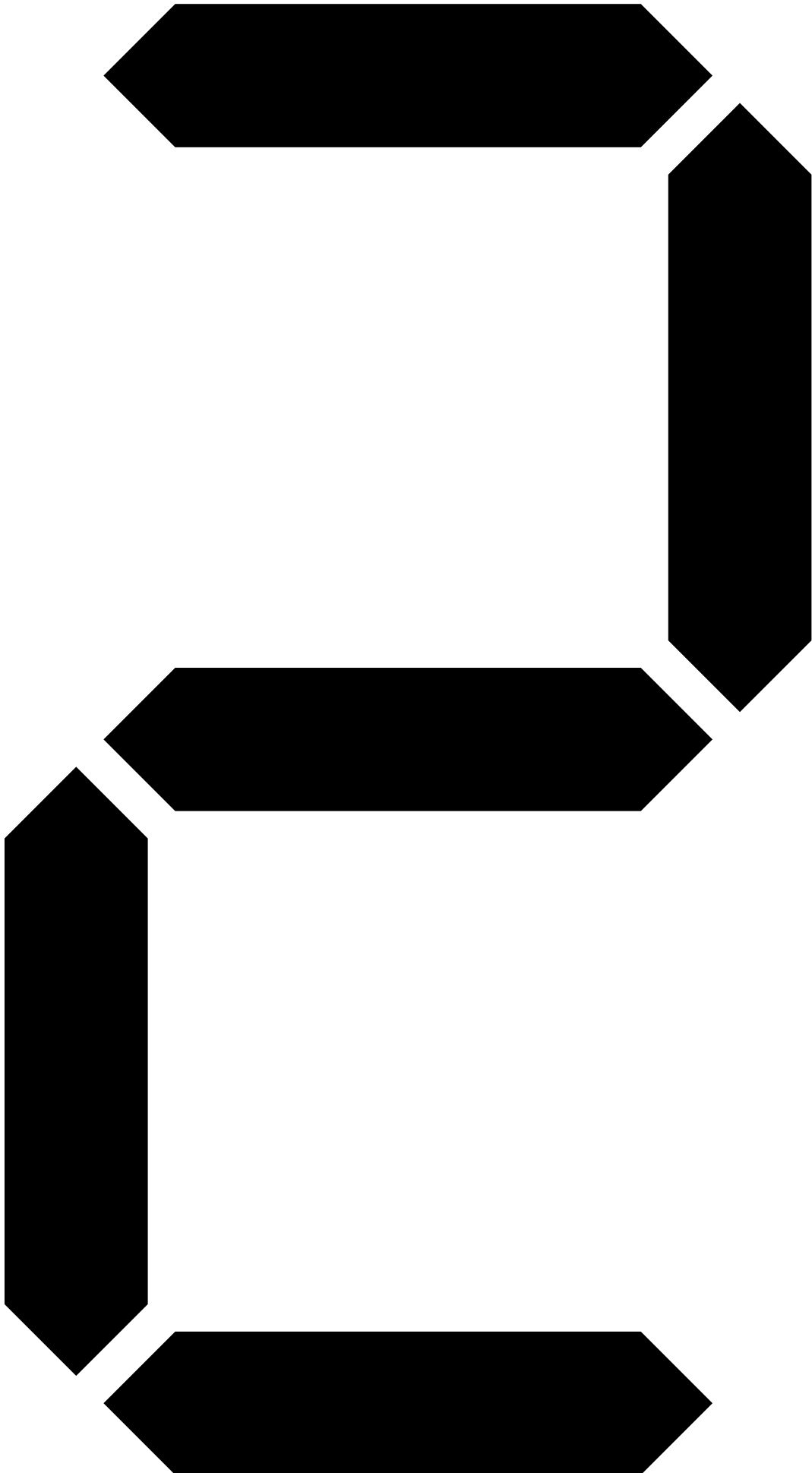
To follow are the digital letters and numbers for E.a.R.L 2 challenge card 6 (Page 26)

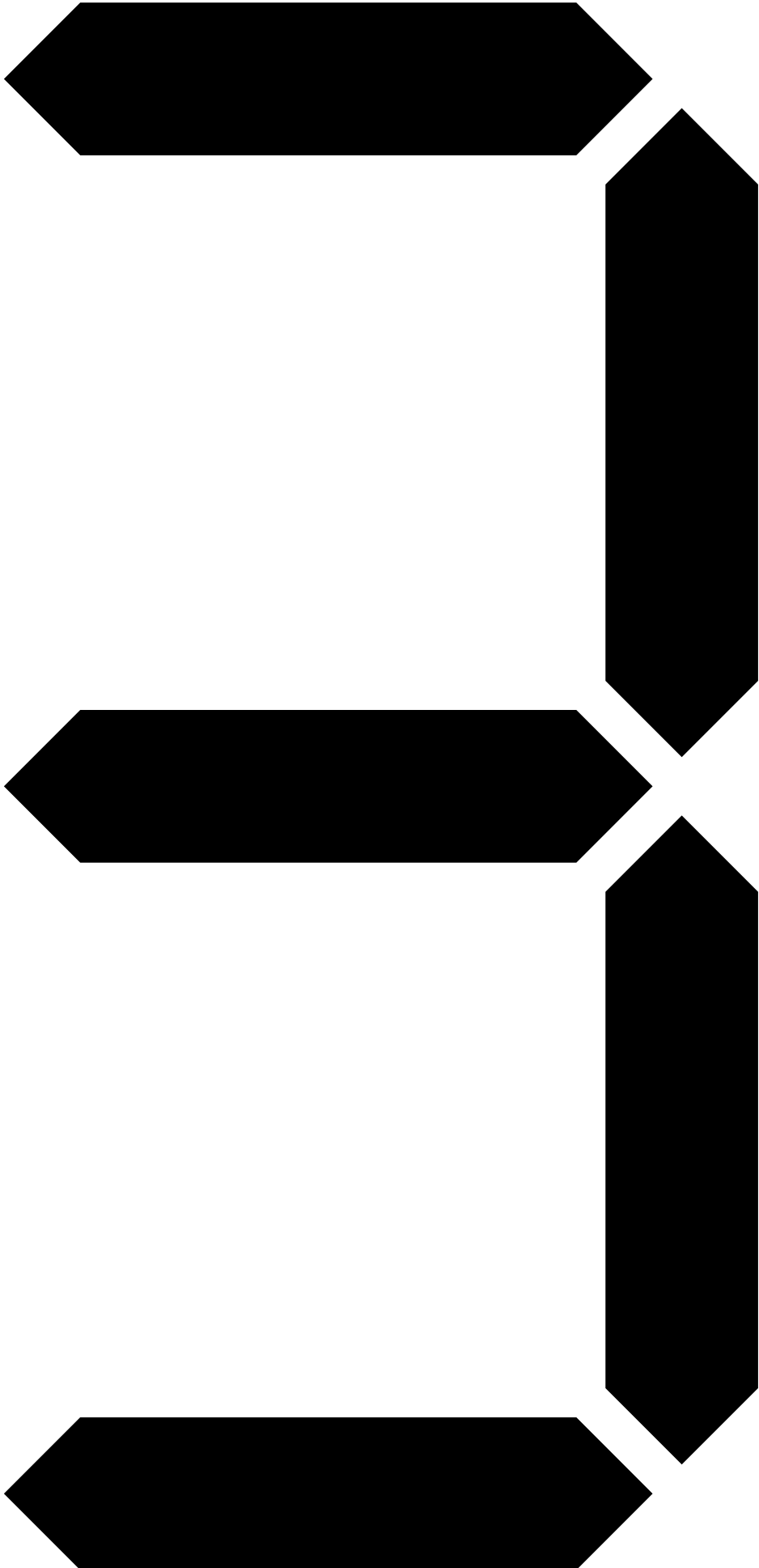


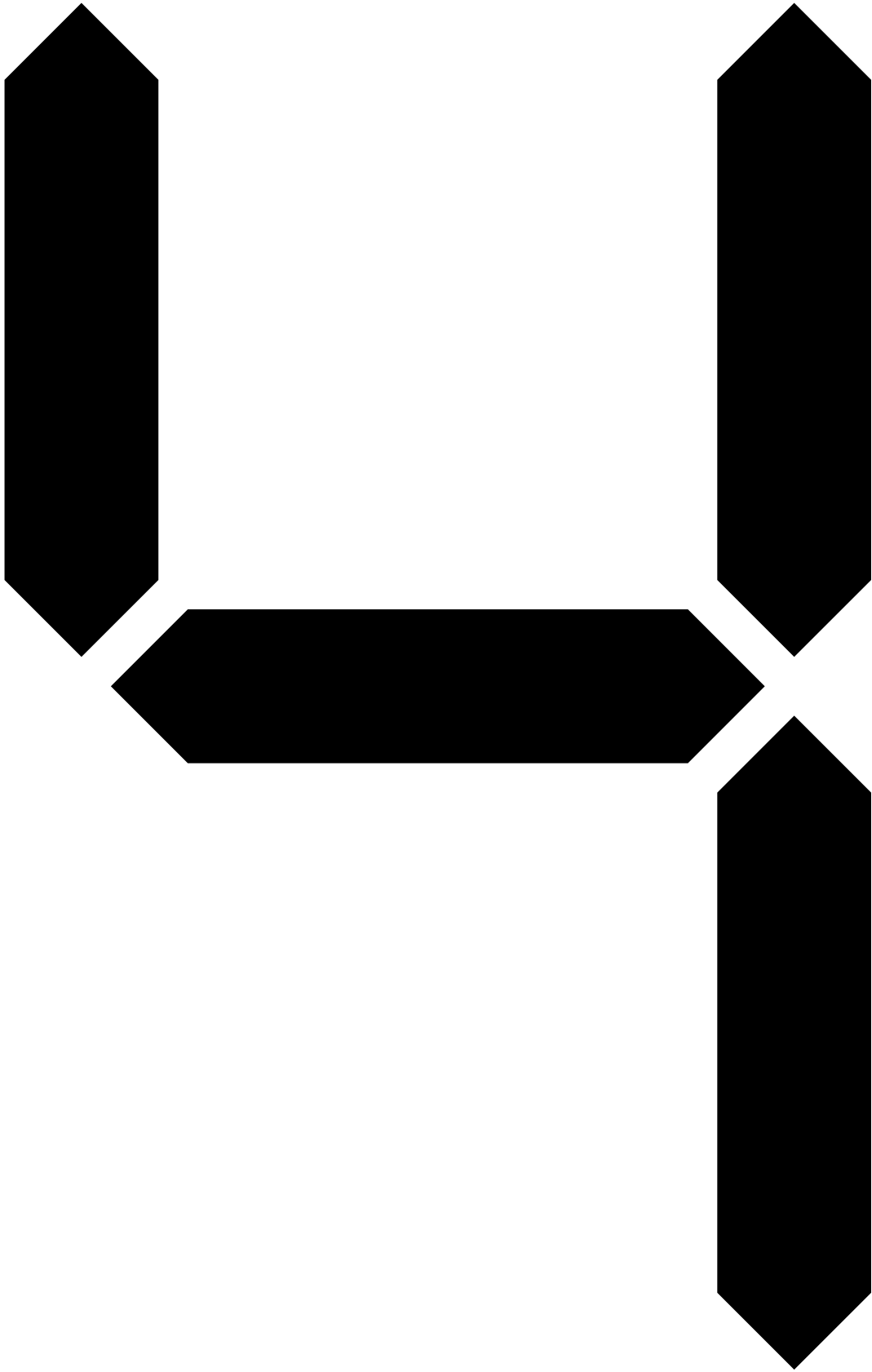
Differentiation - certain numbers and letters are easier to navigate E.a.R.L 2 along

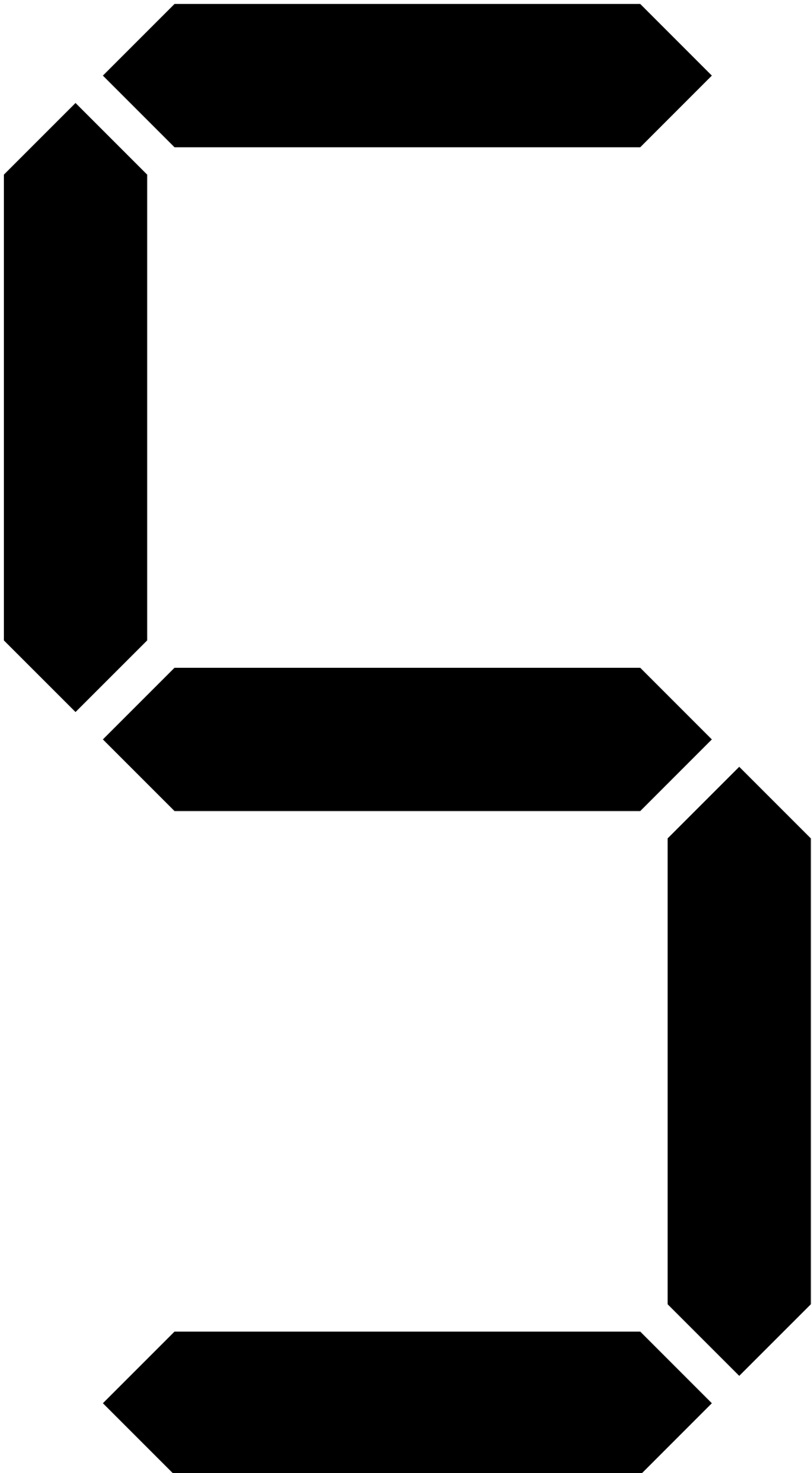


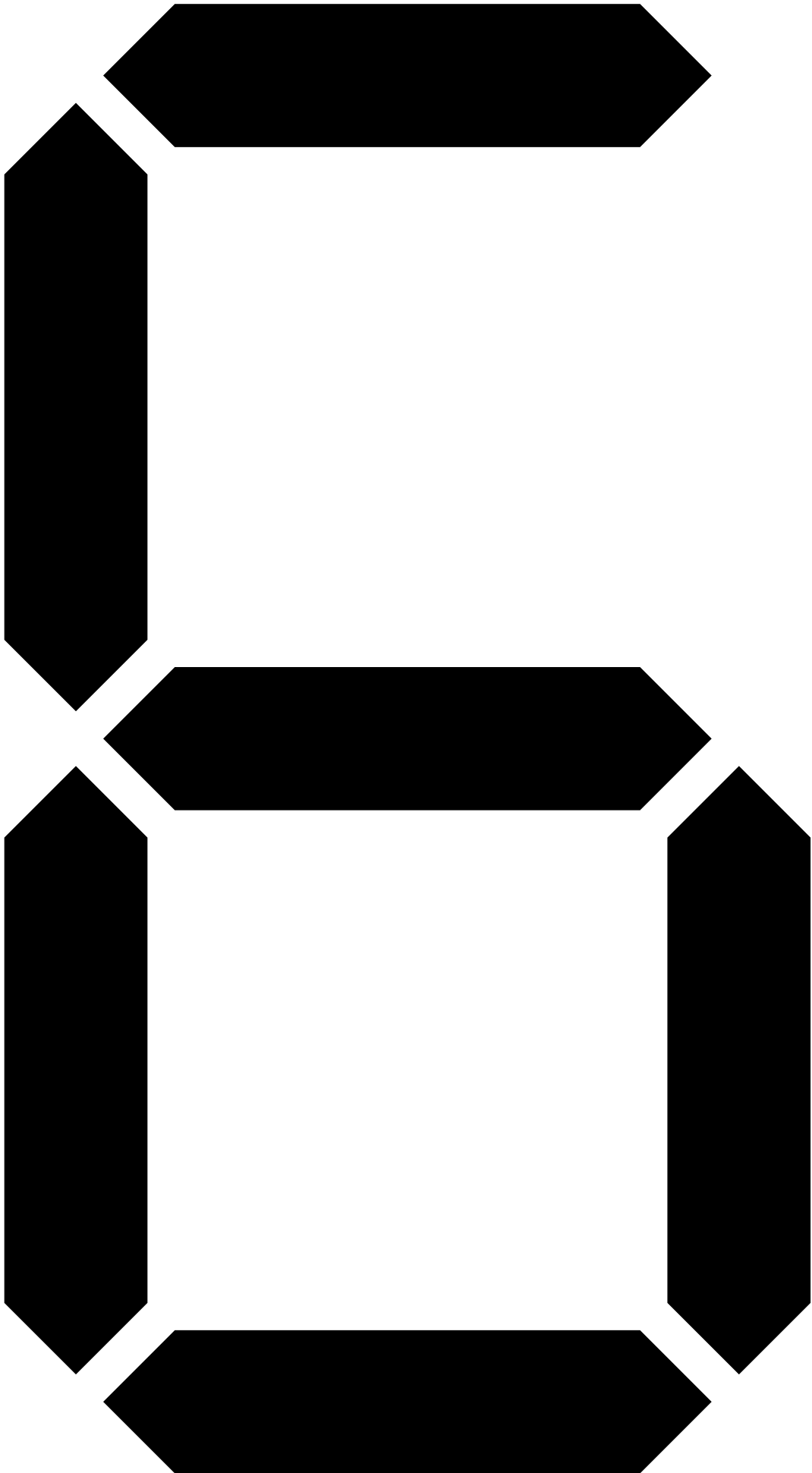


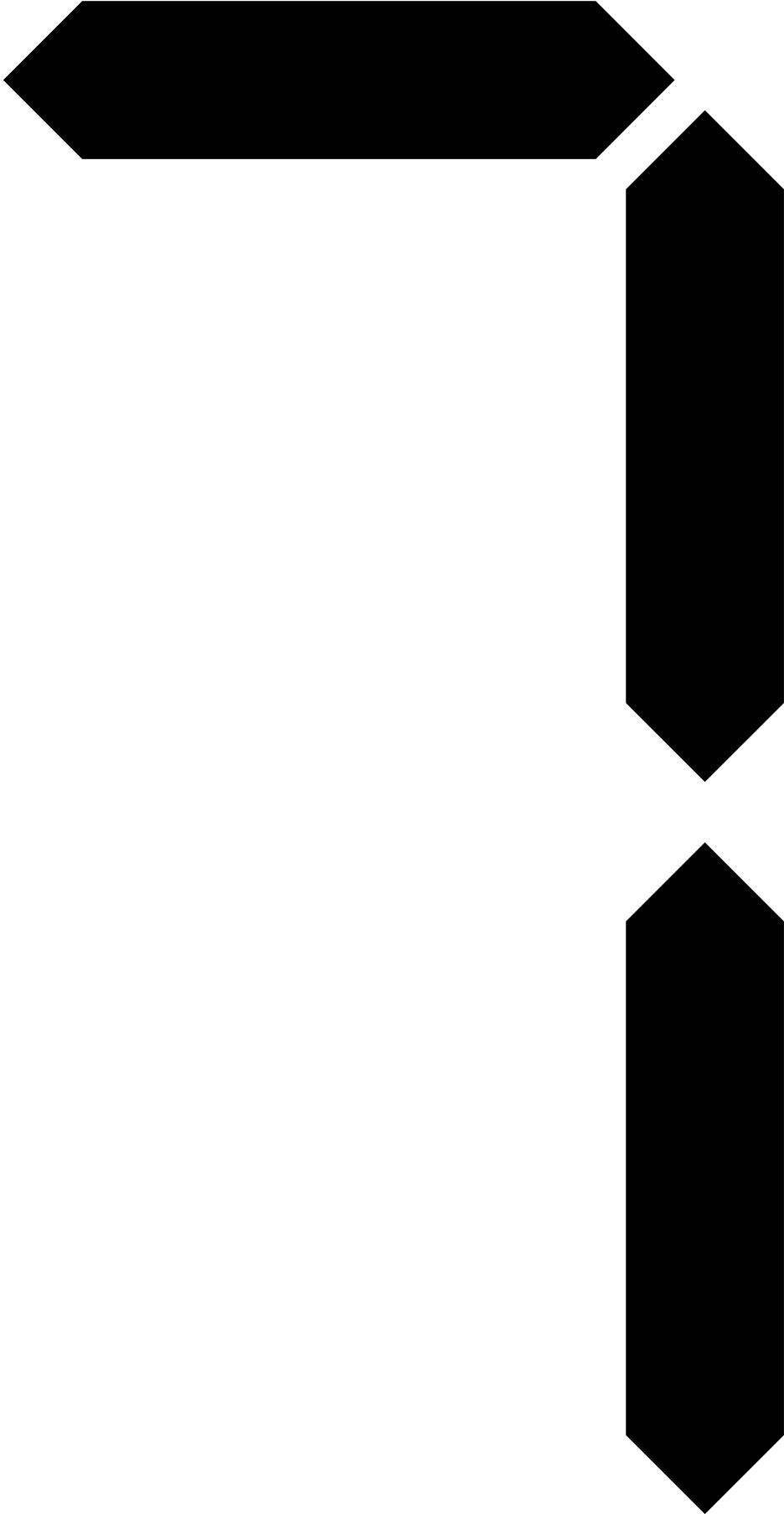


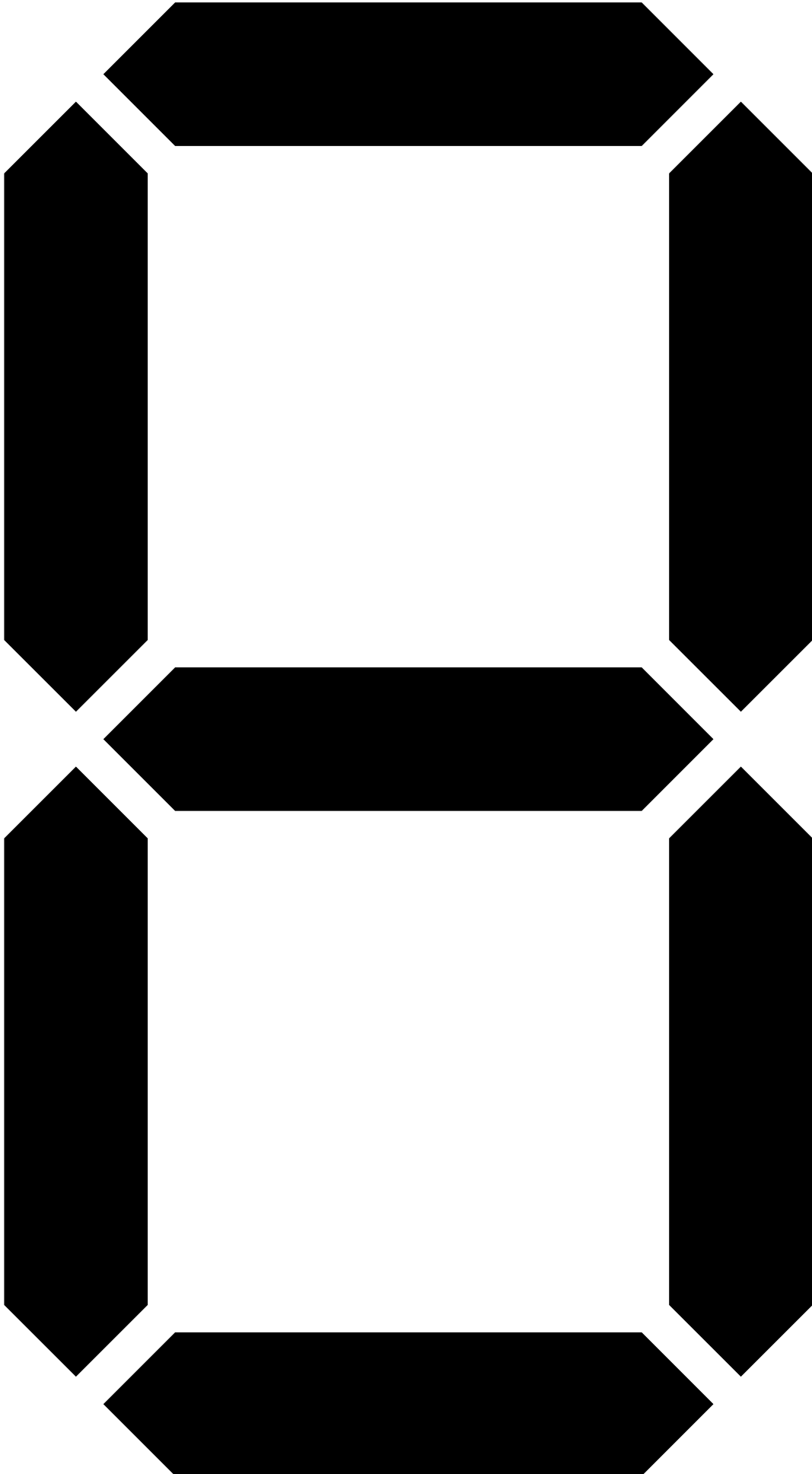


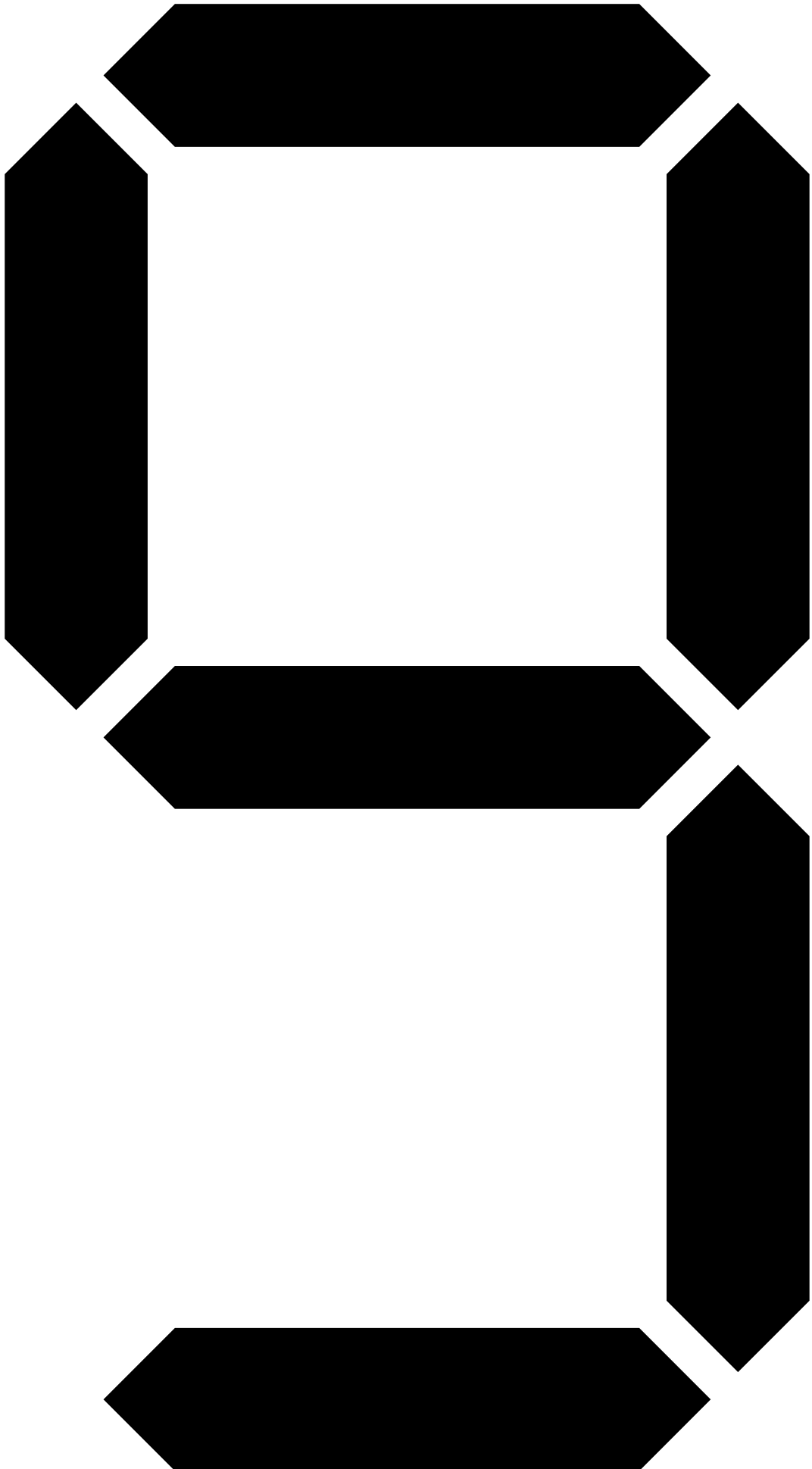


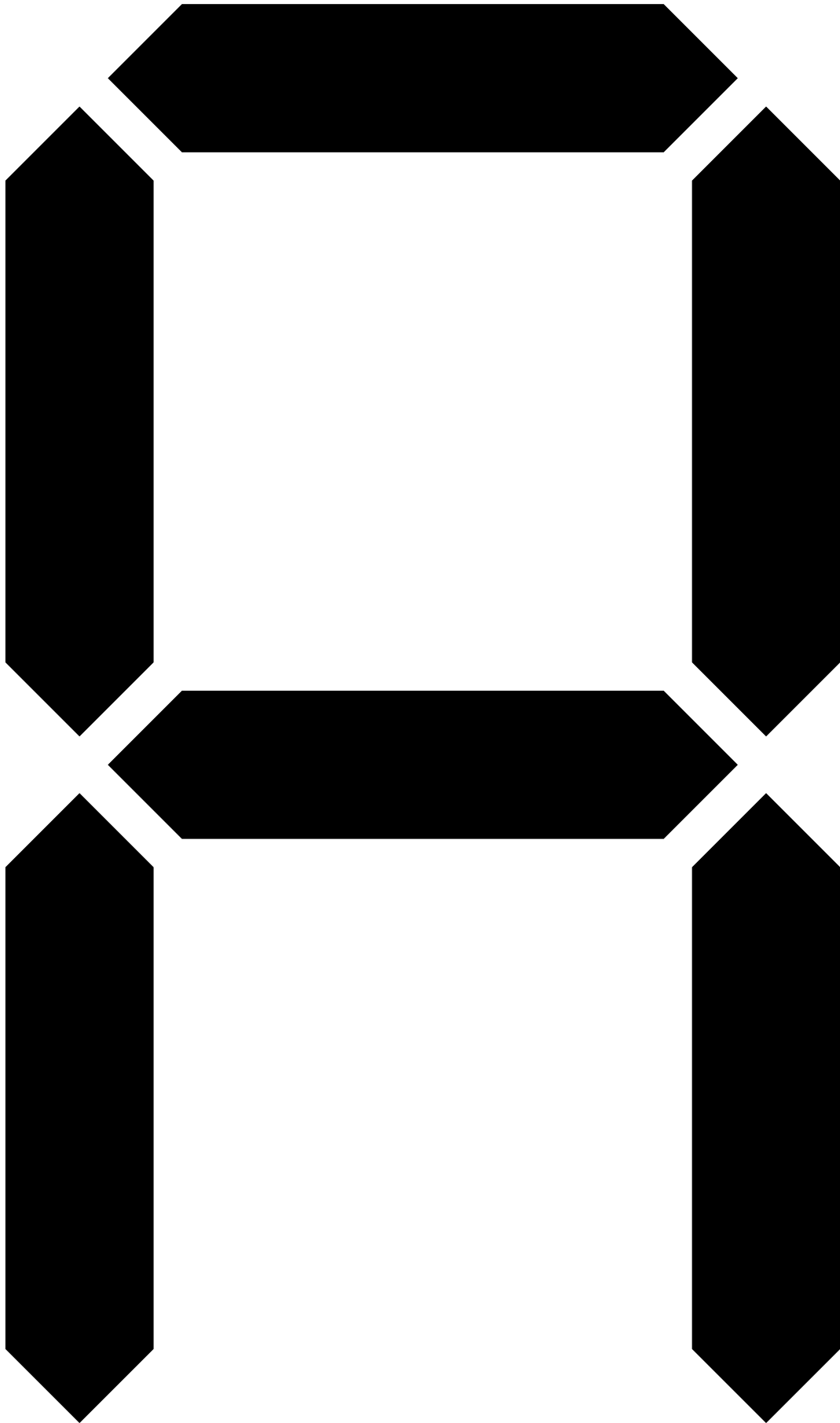


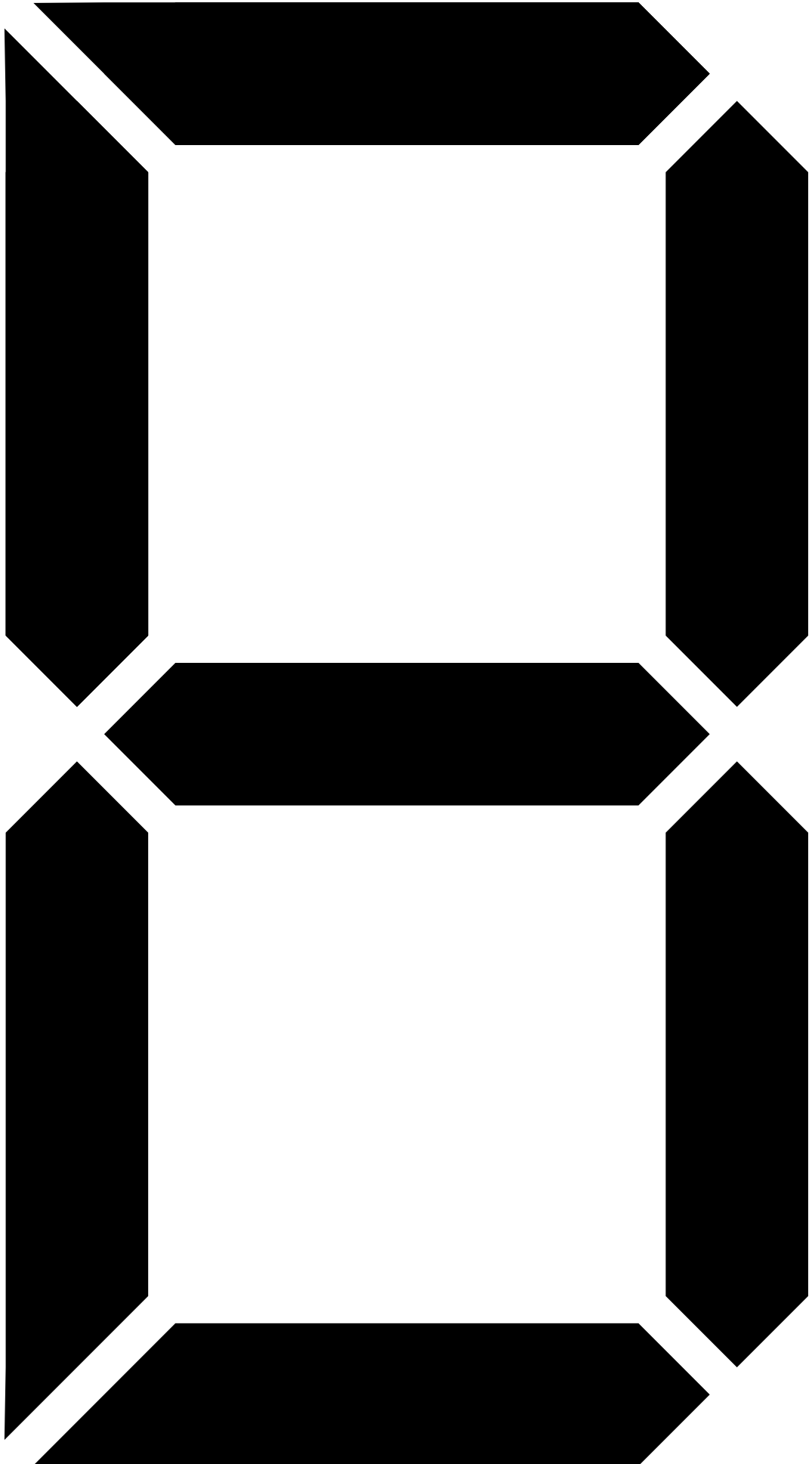


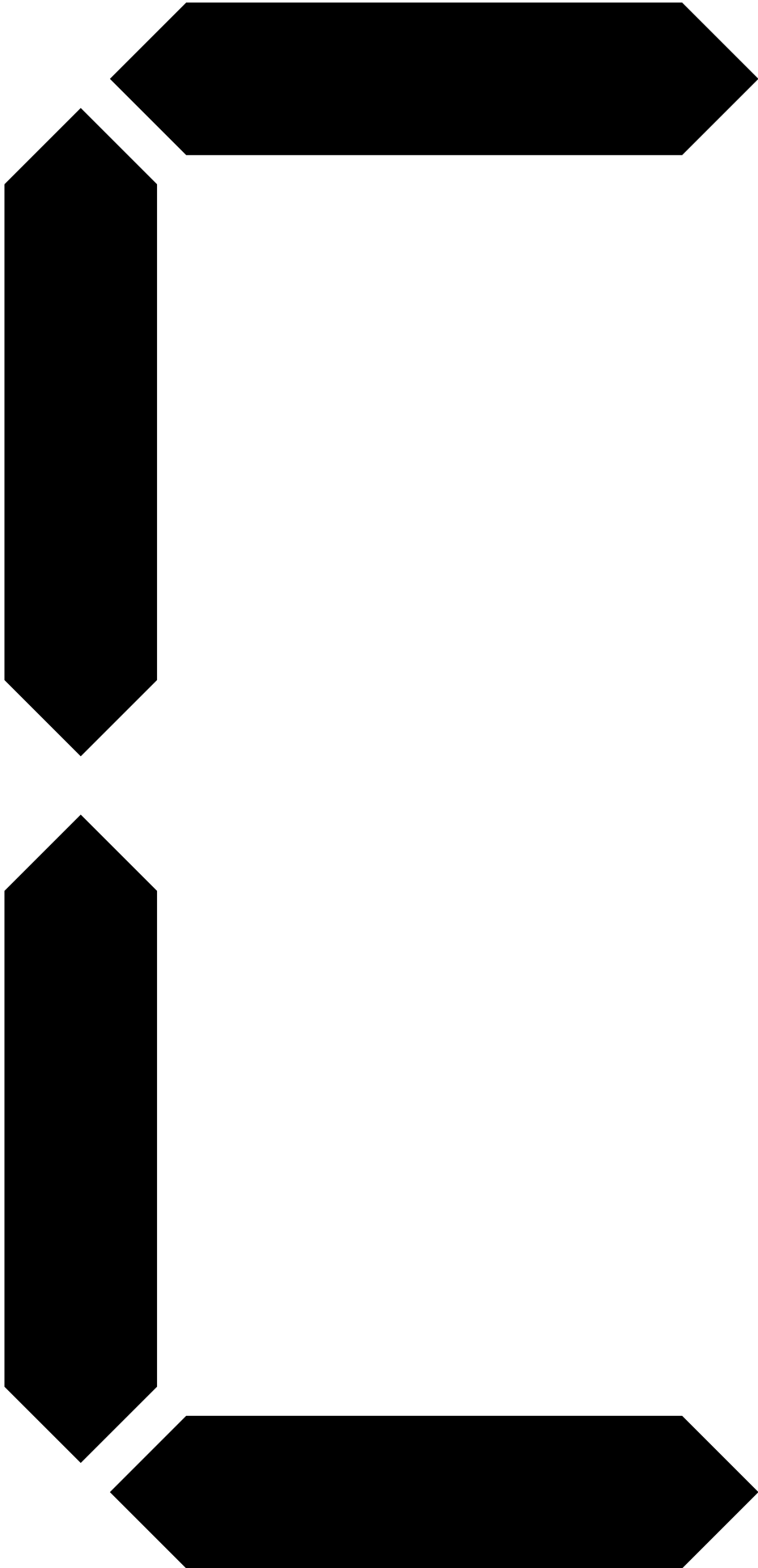


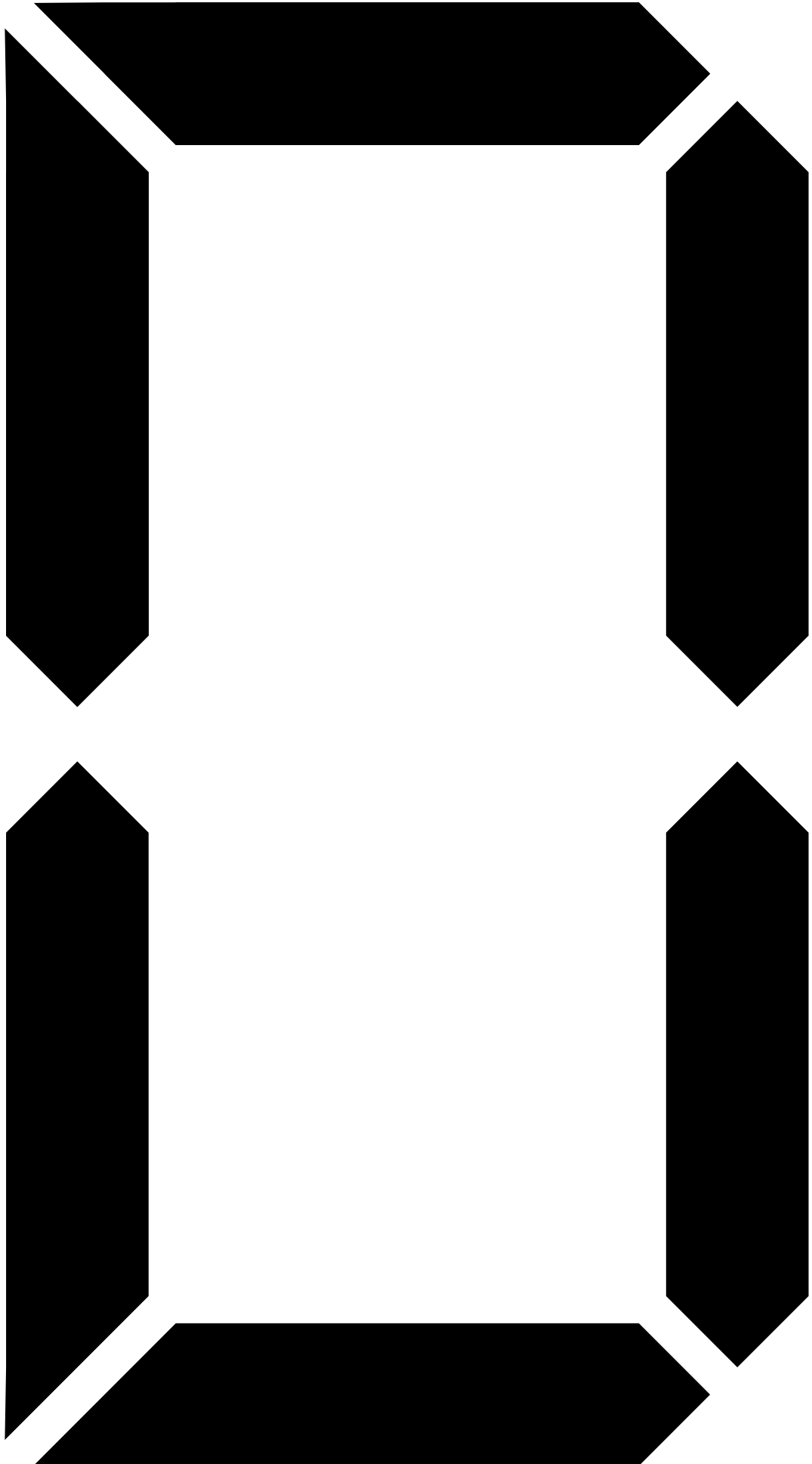


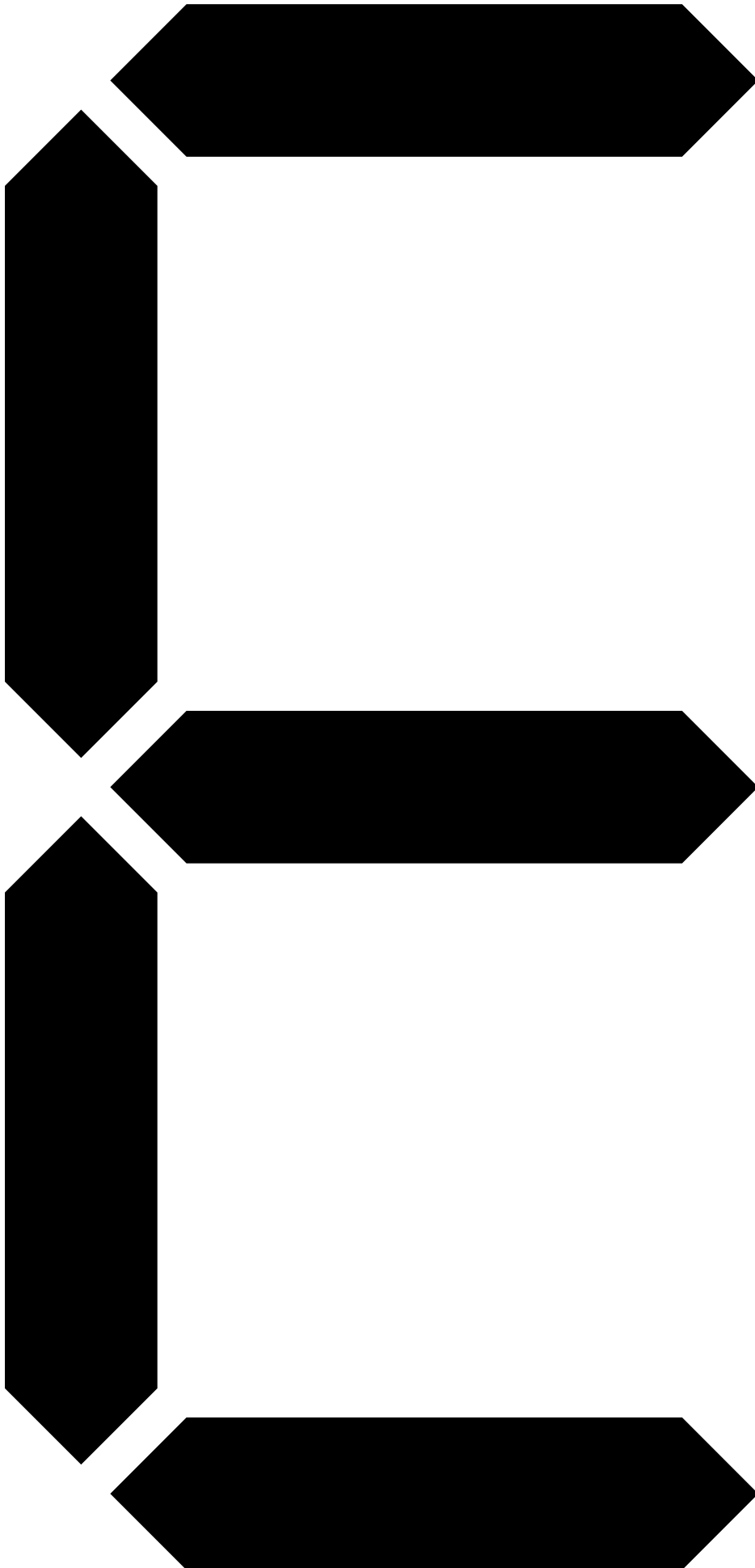


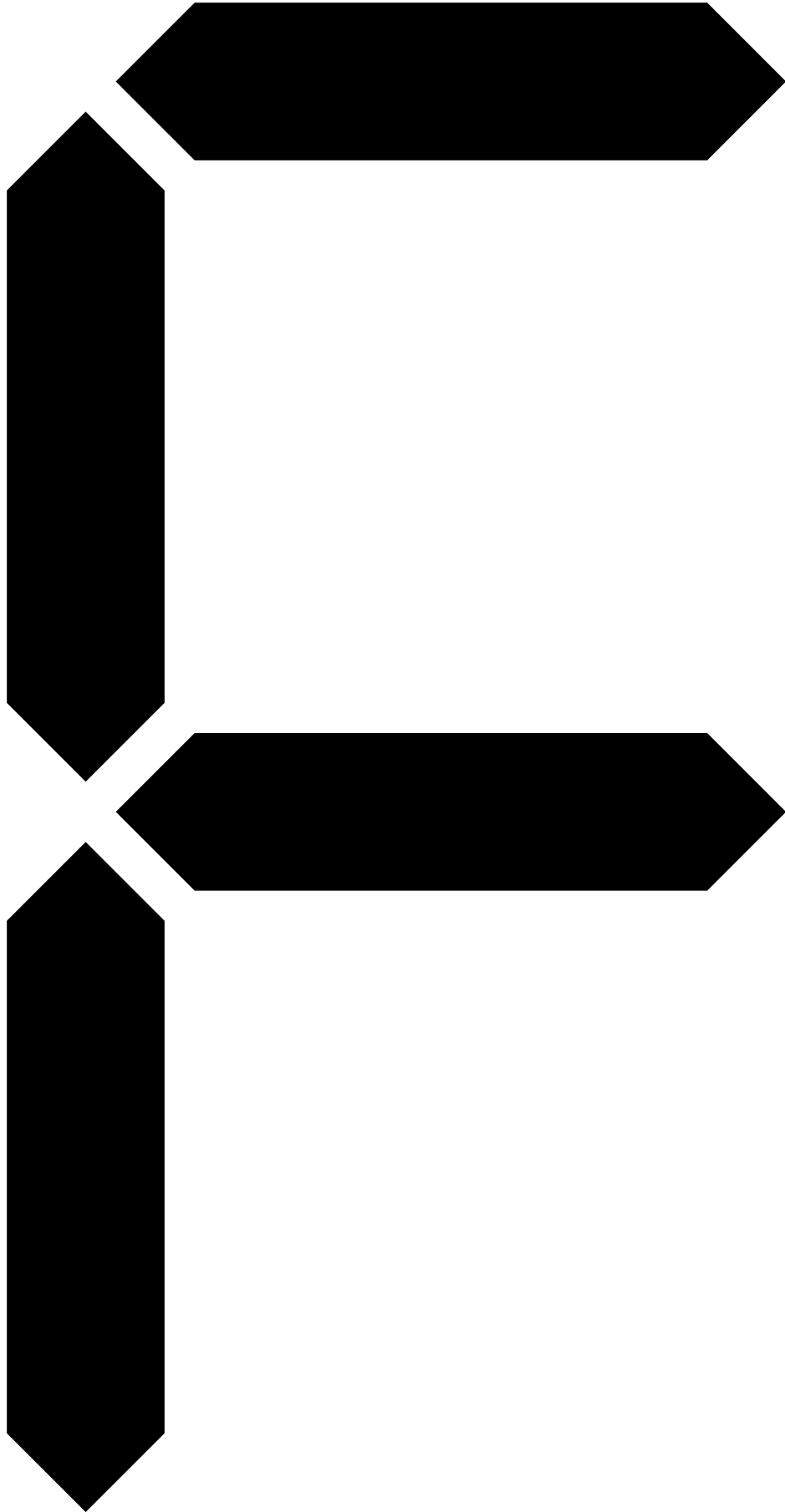


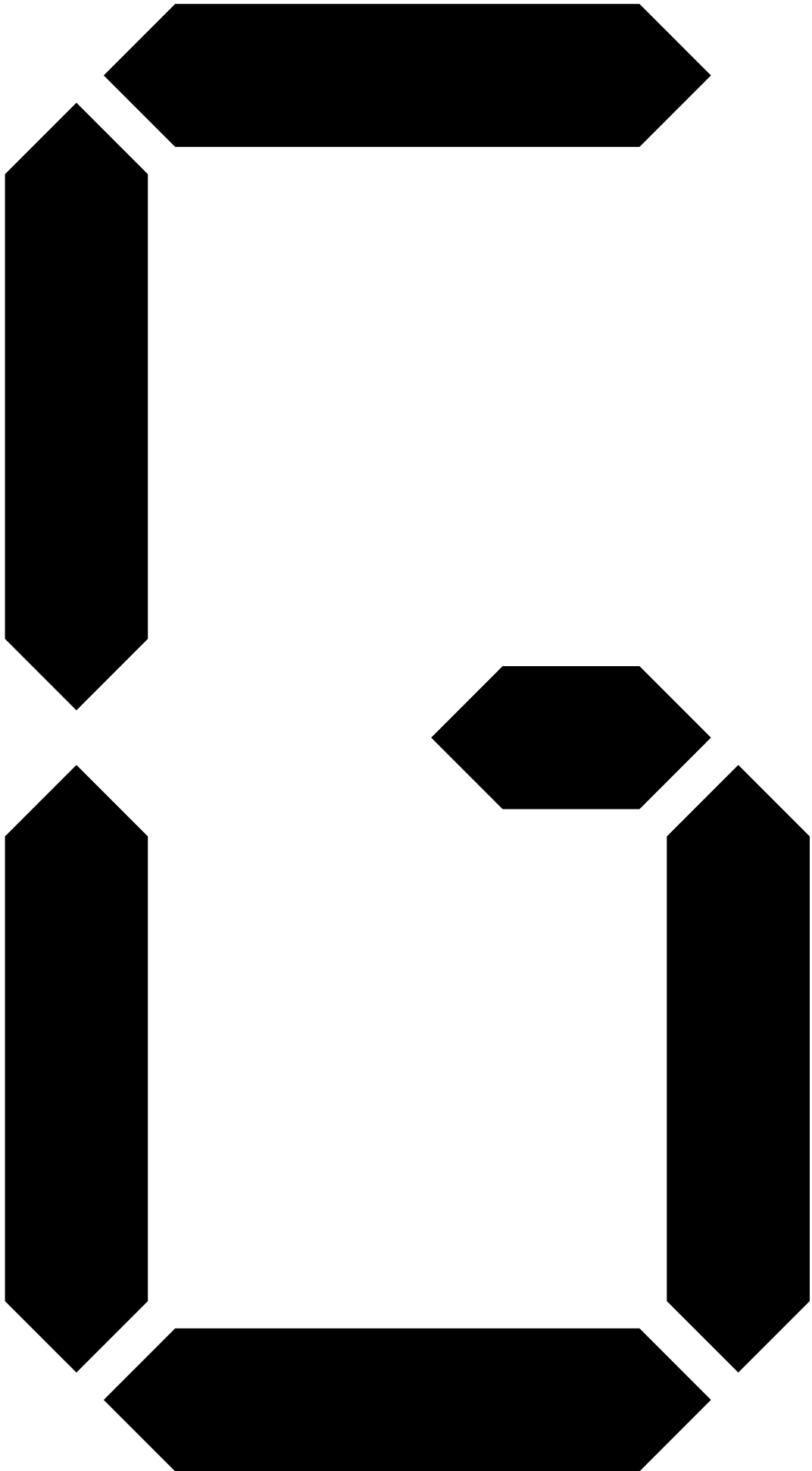


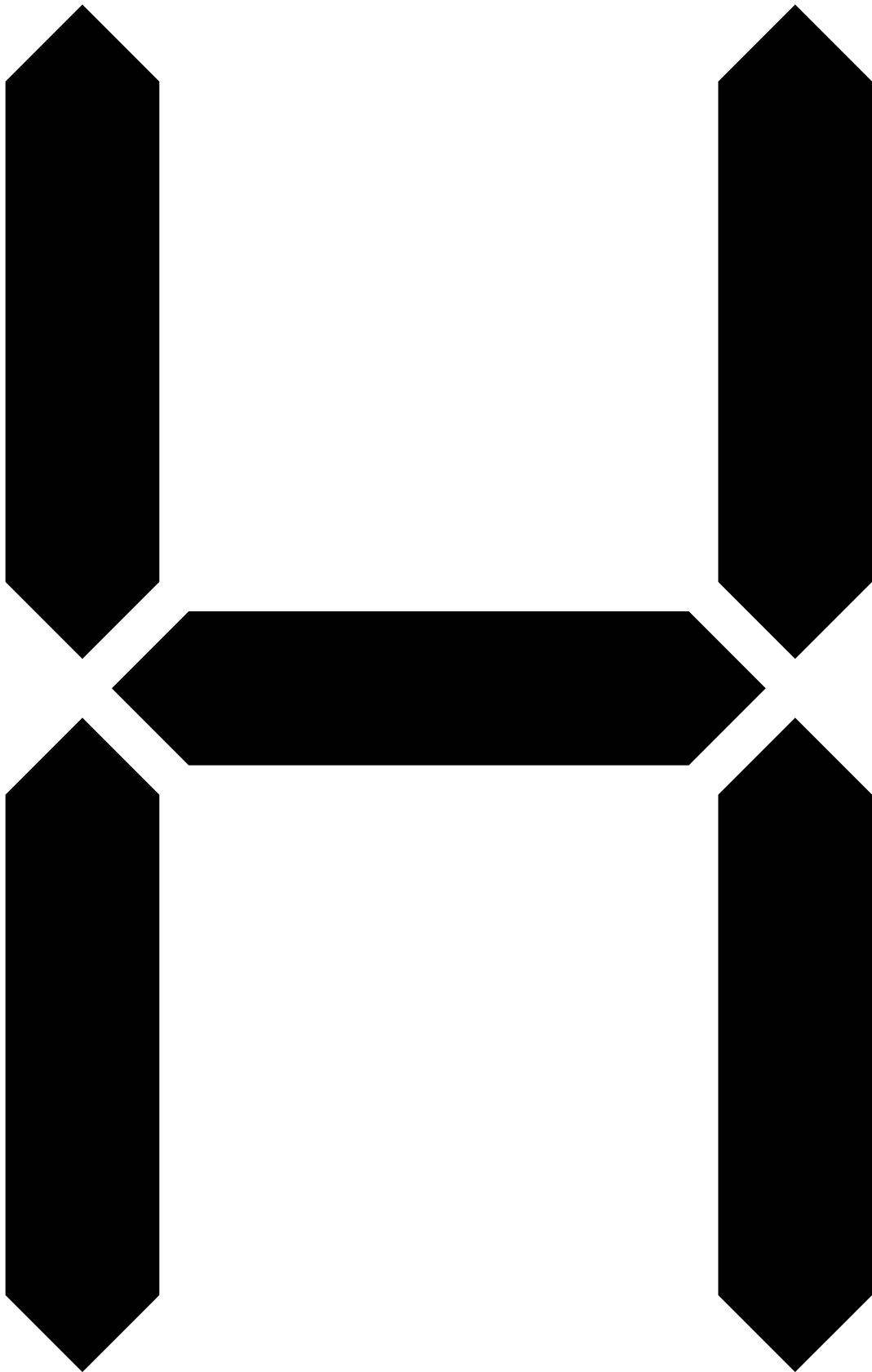


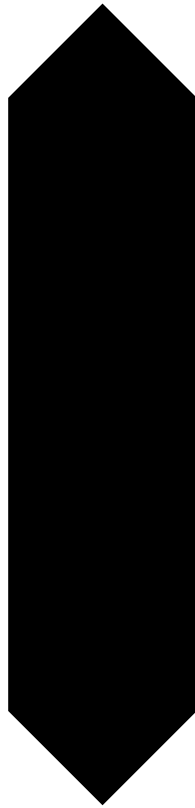




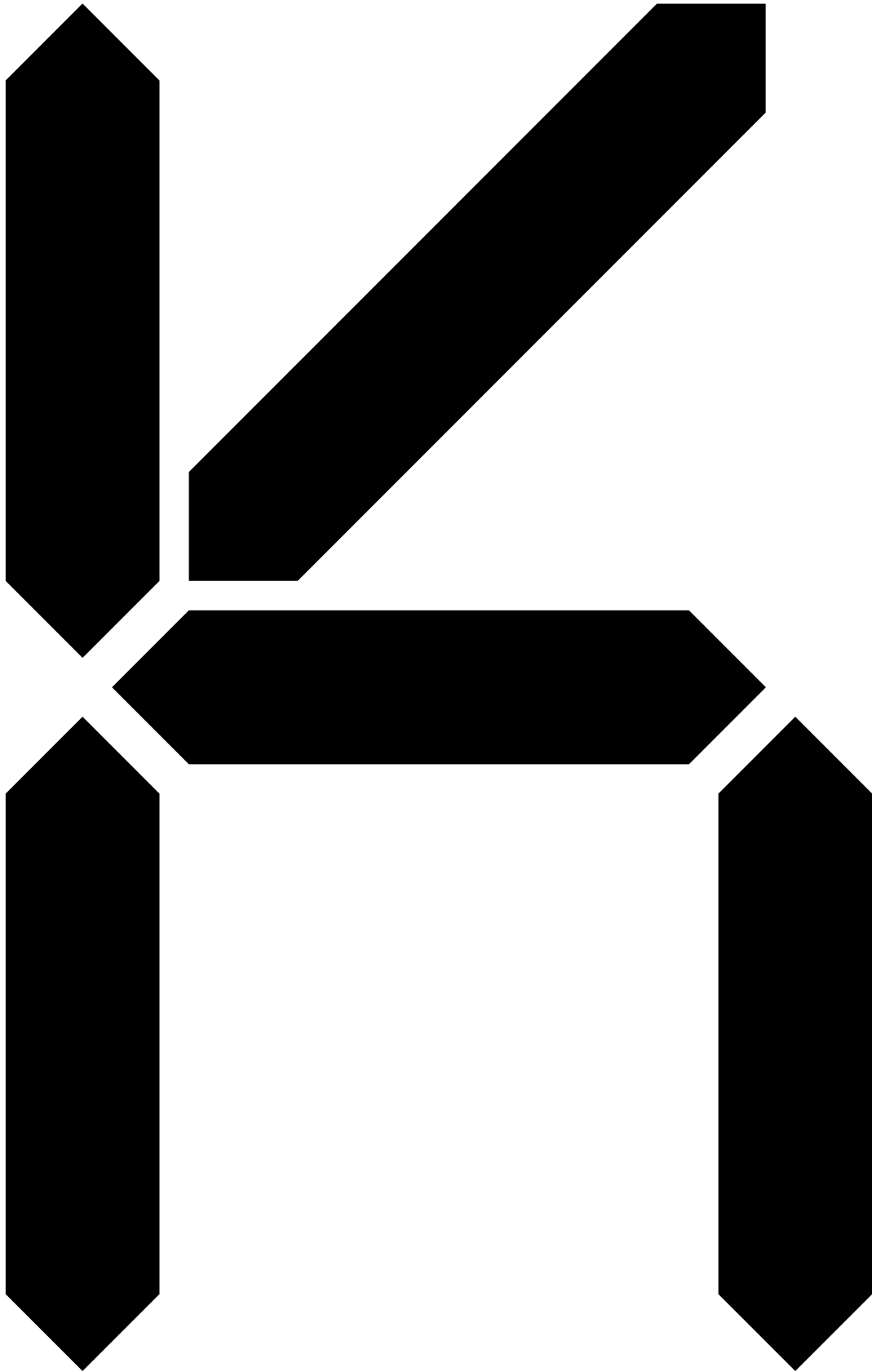


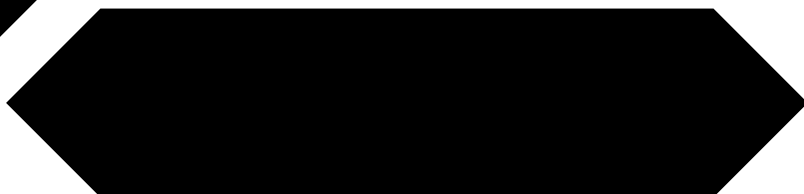
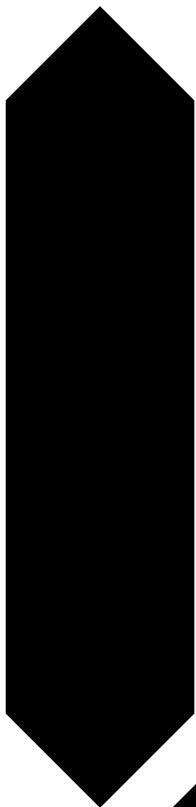
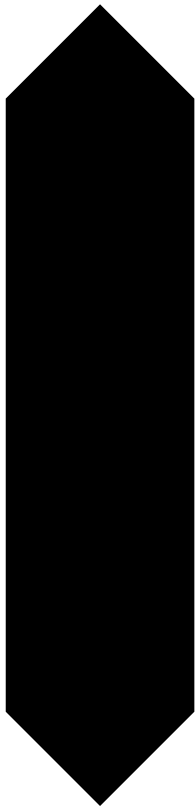


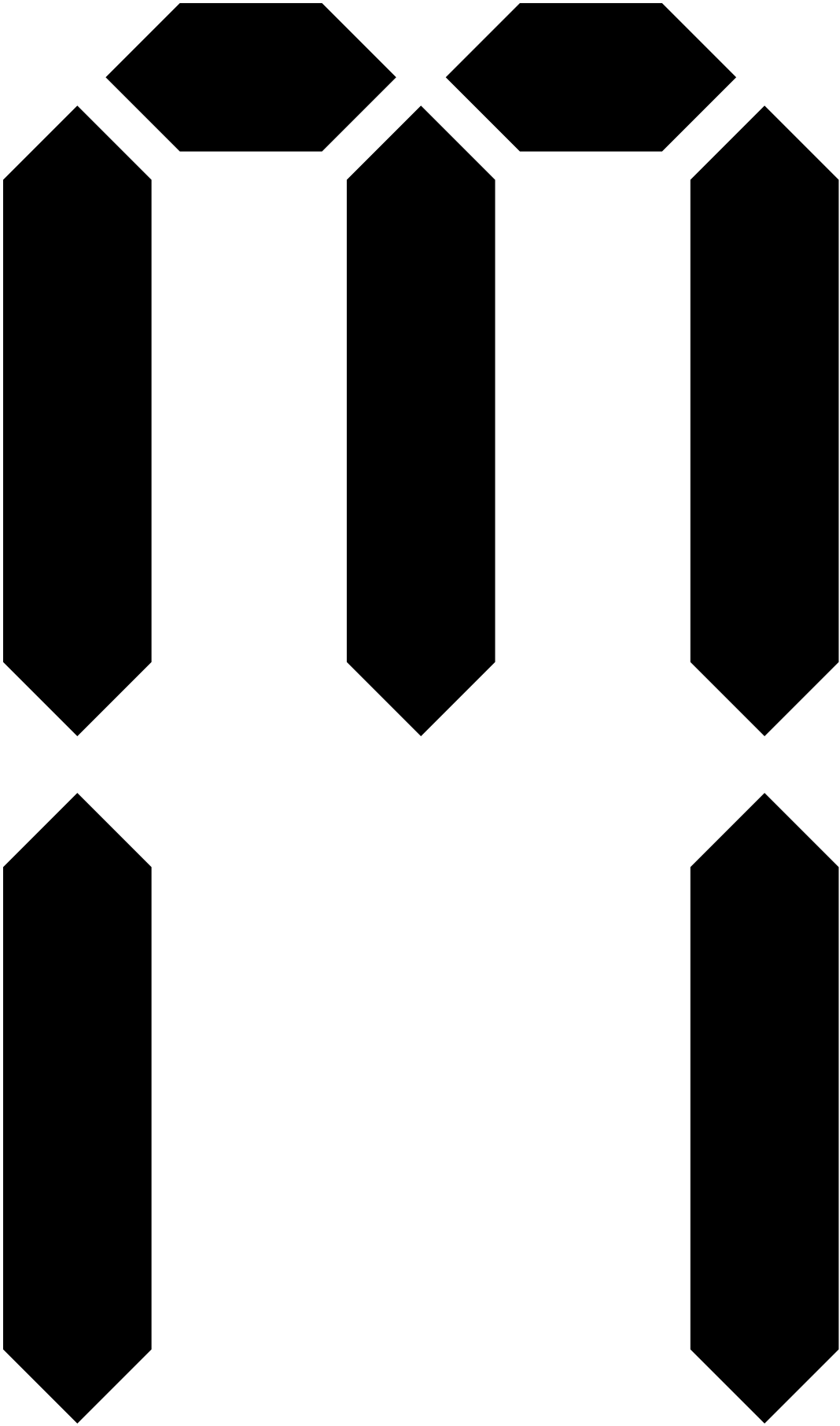


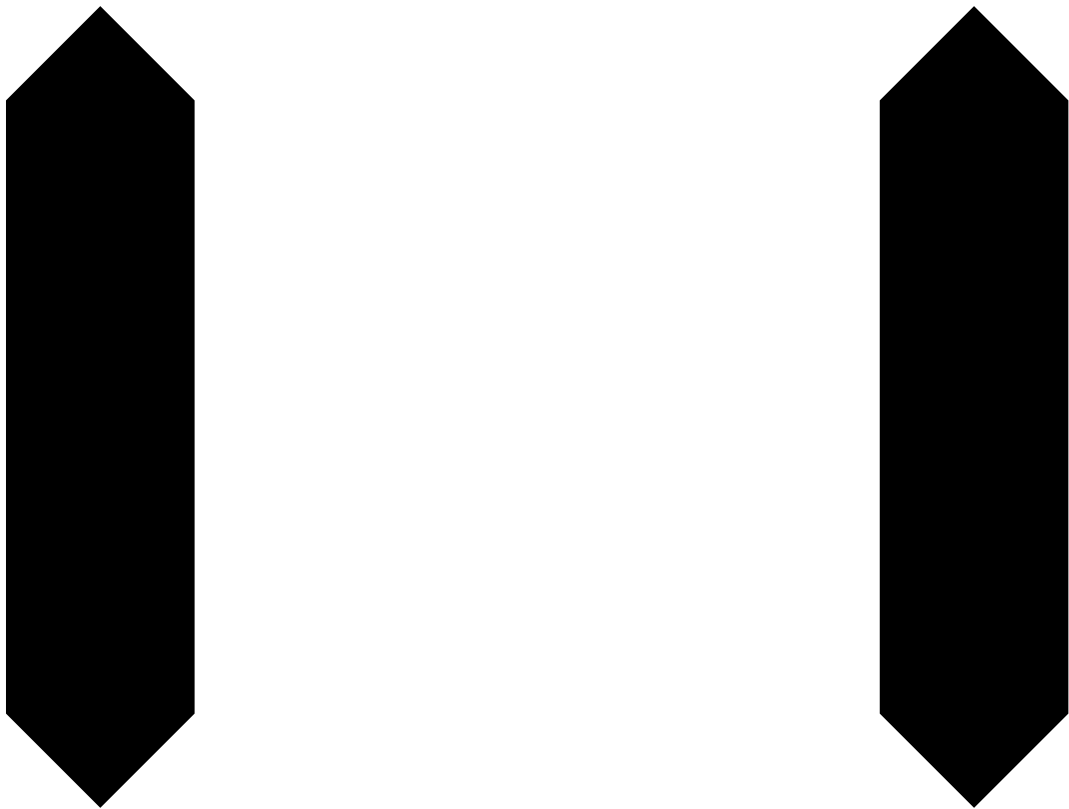
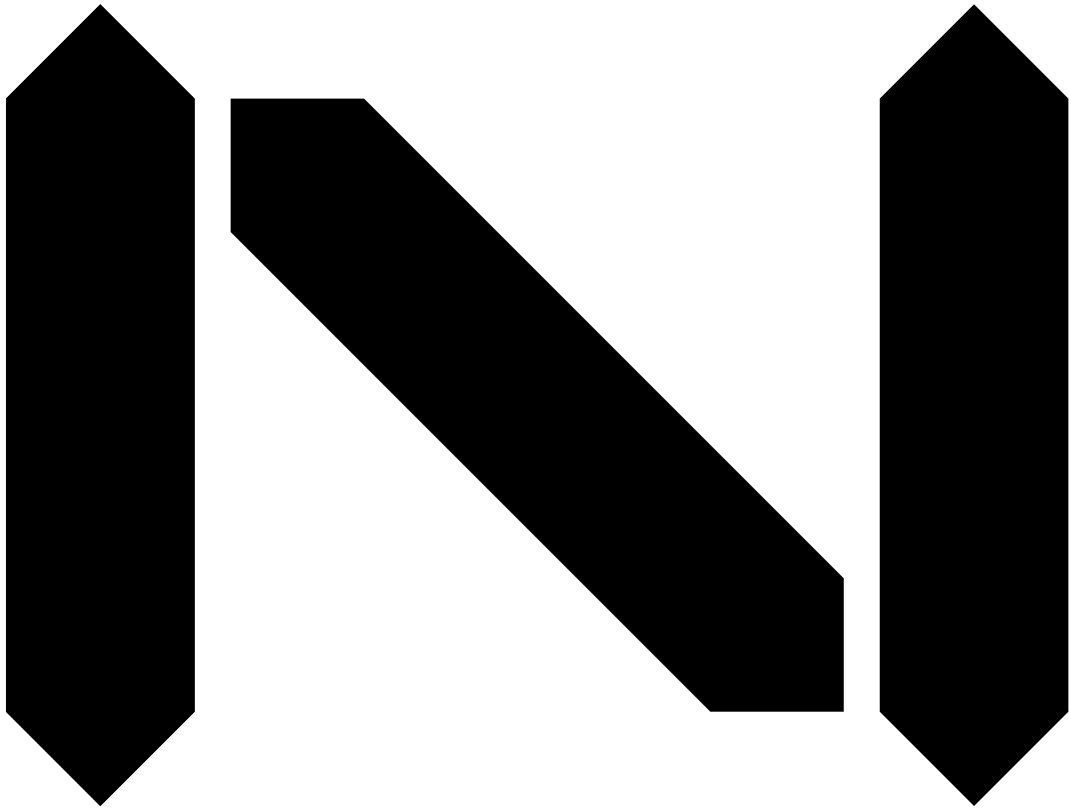


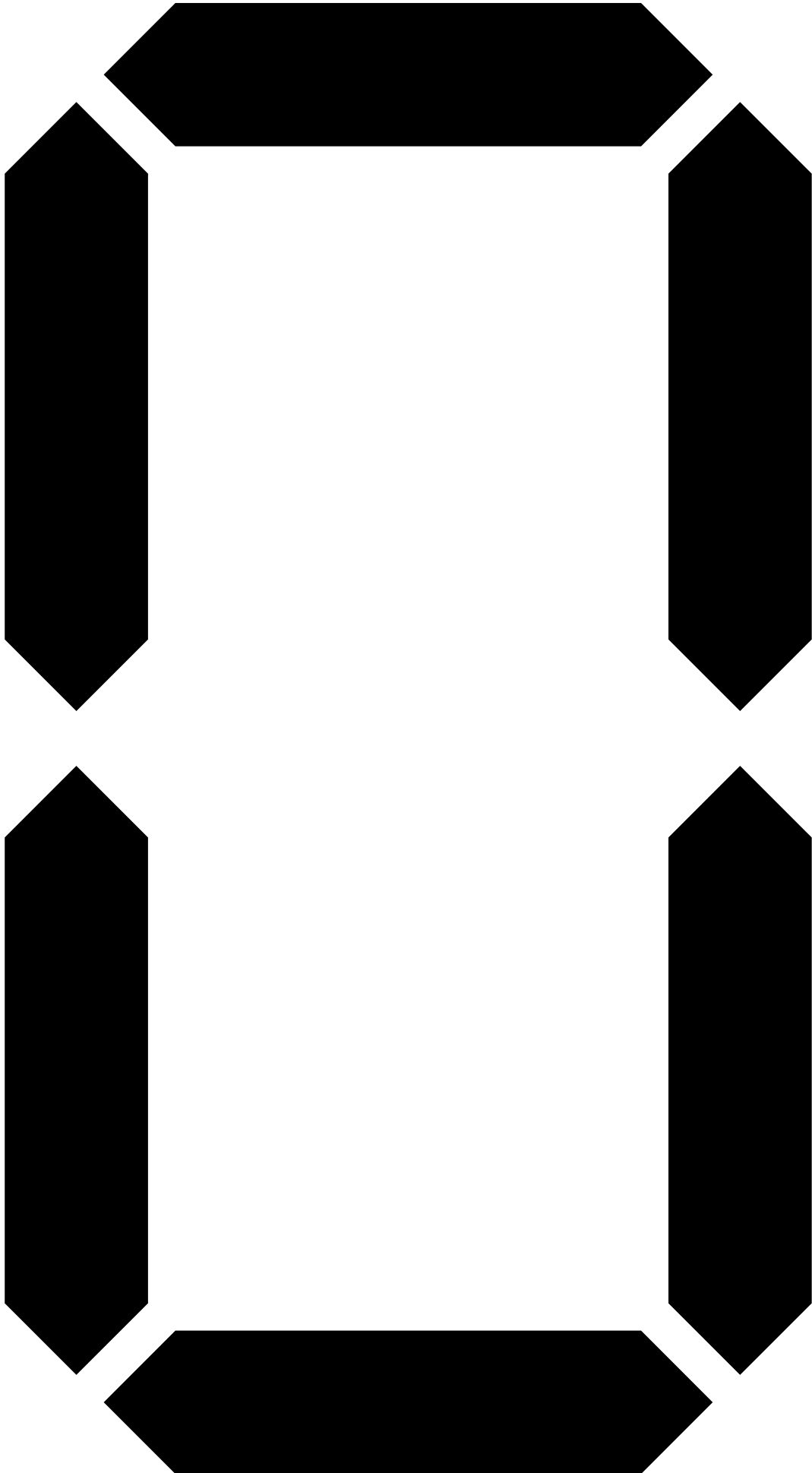


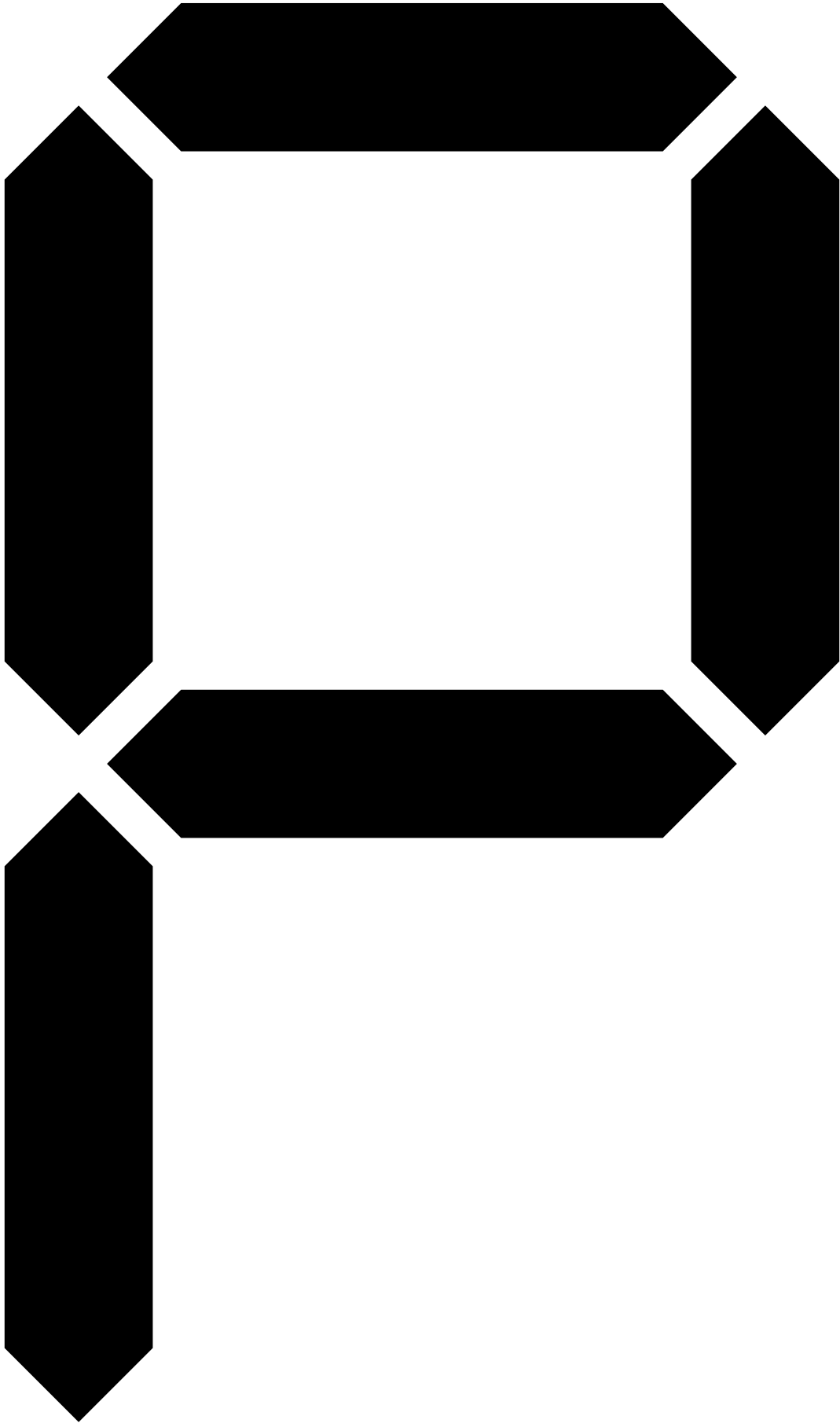


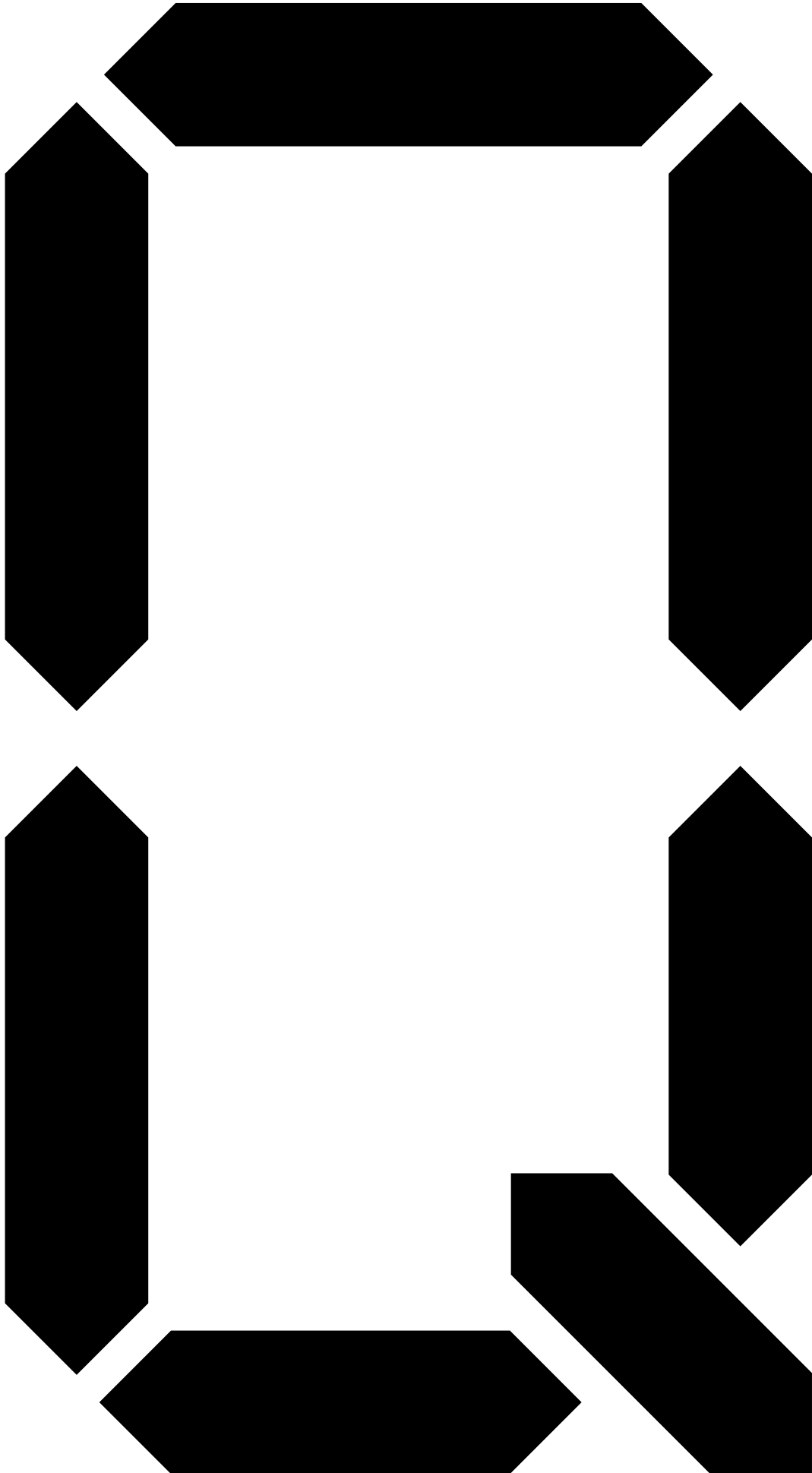


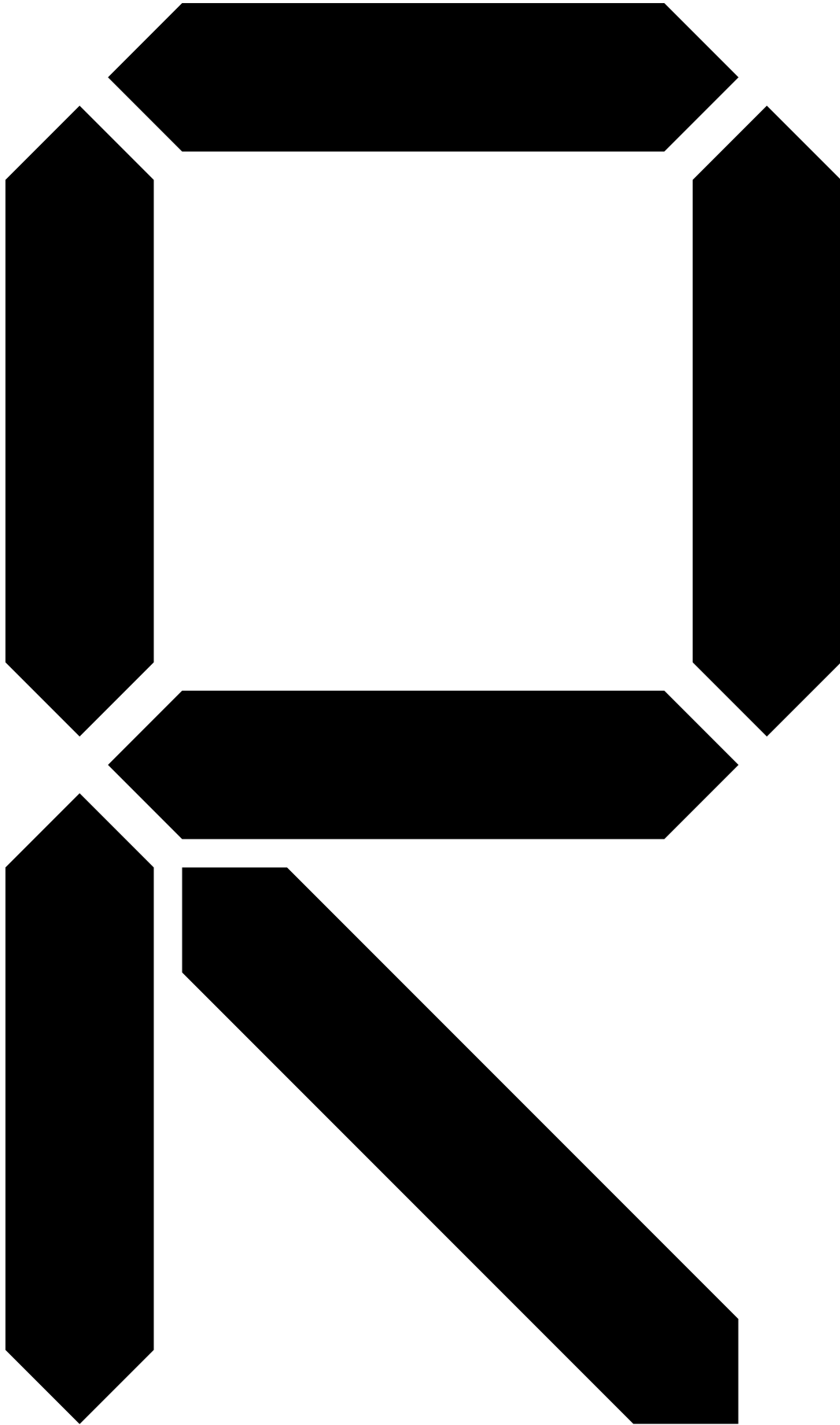


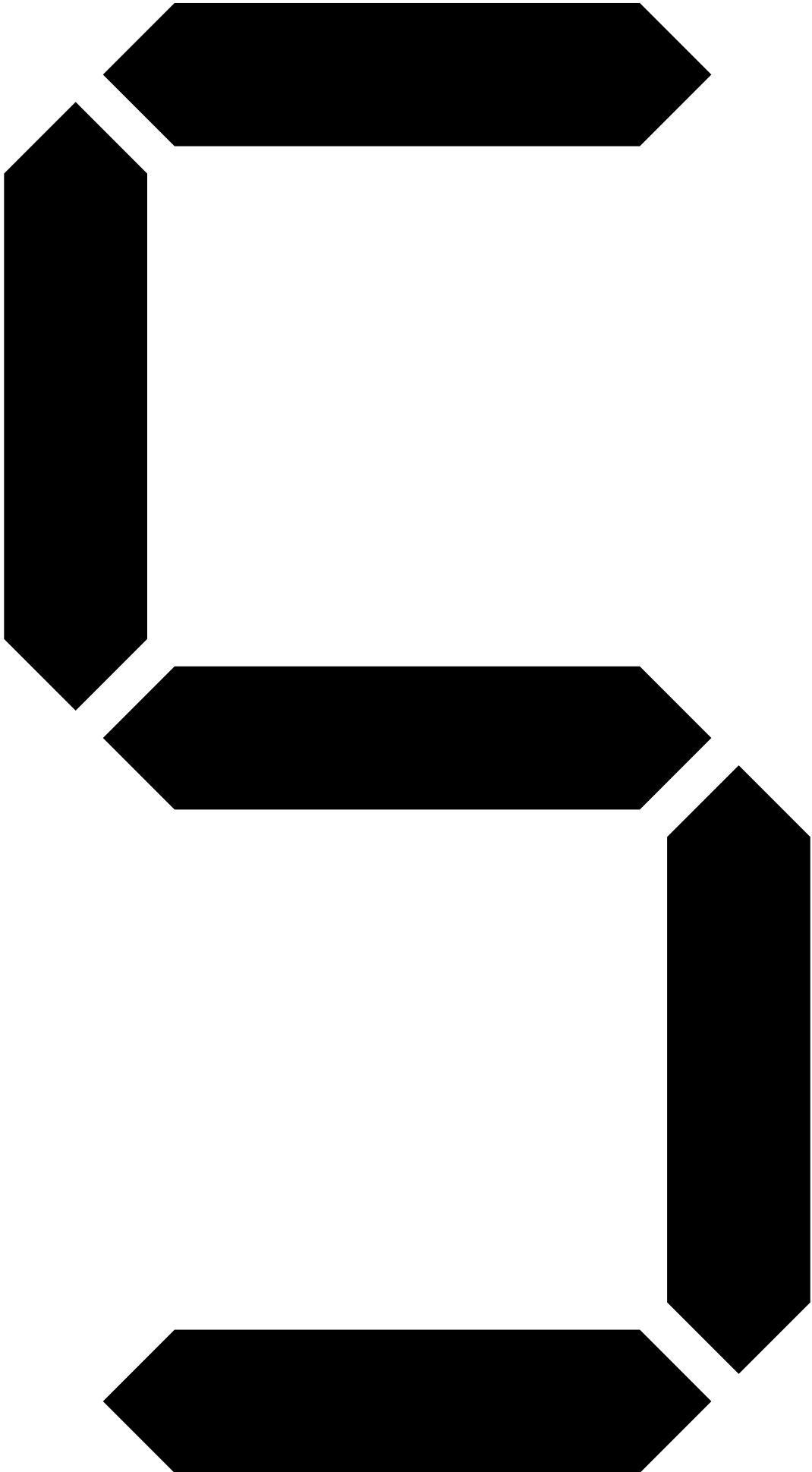


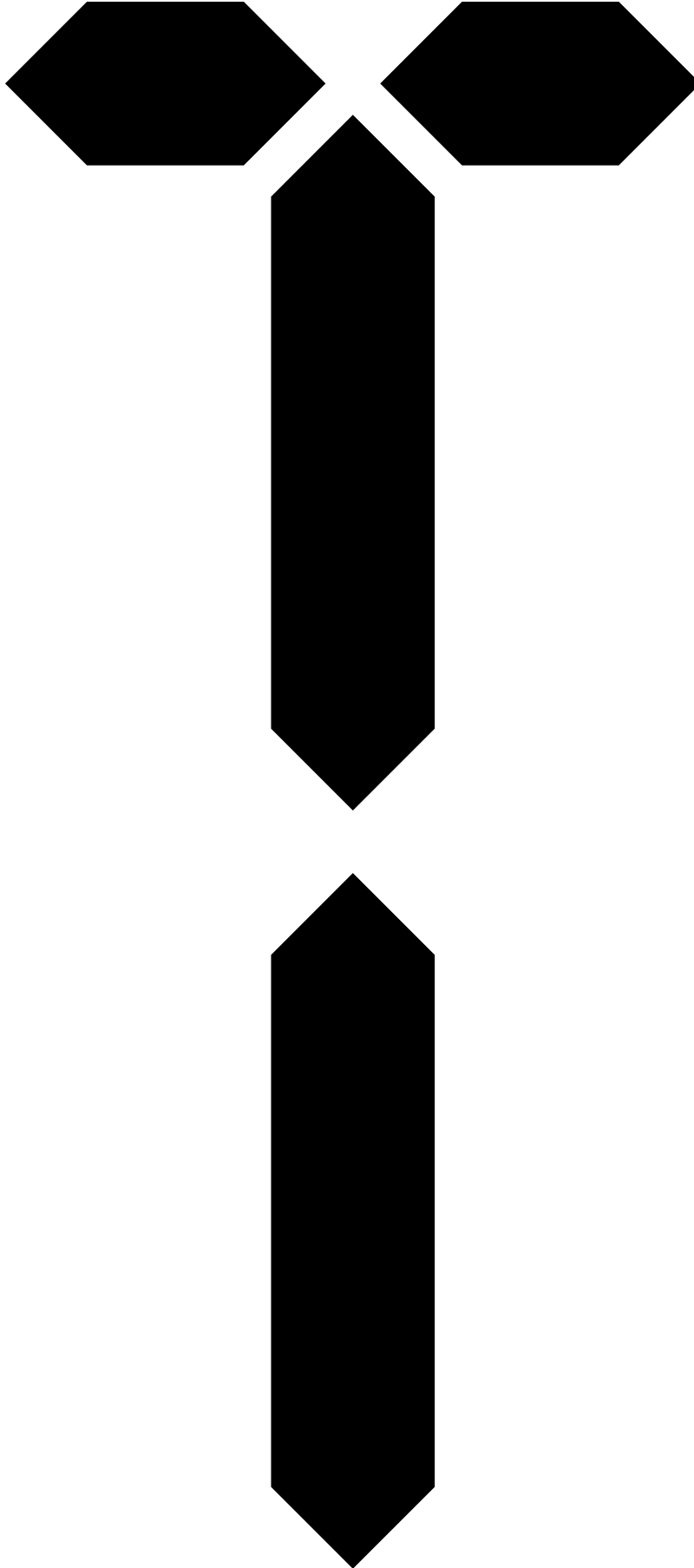


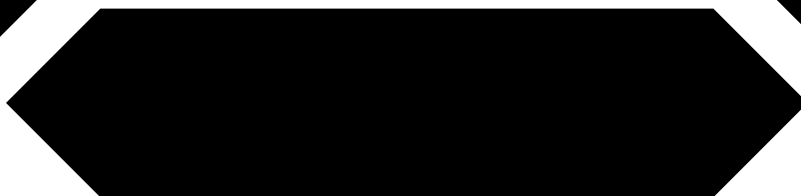


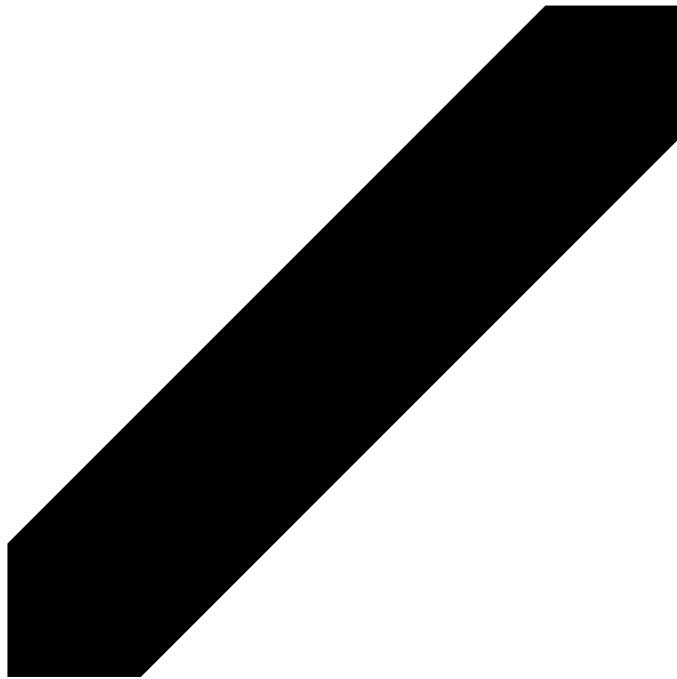
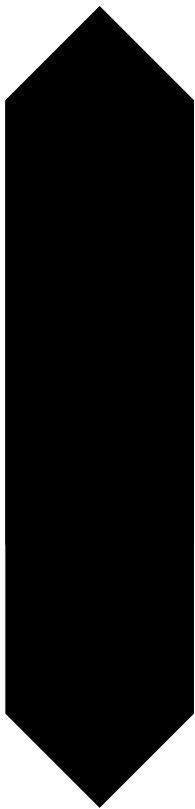


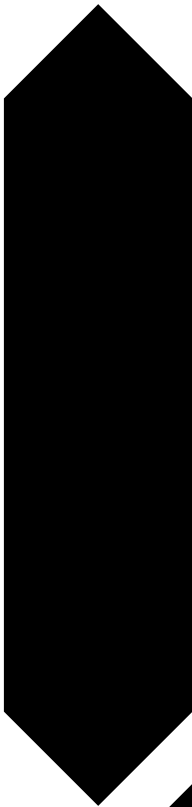
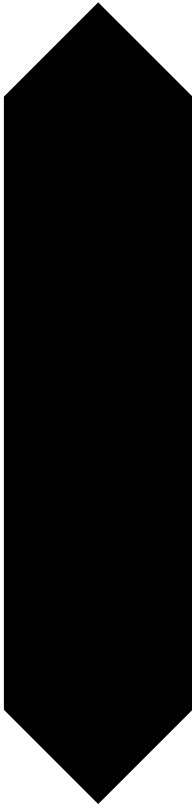


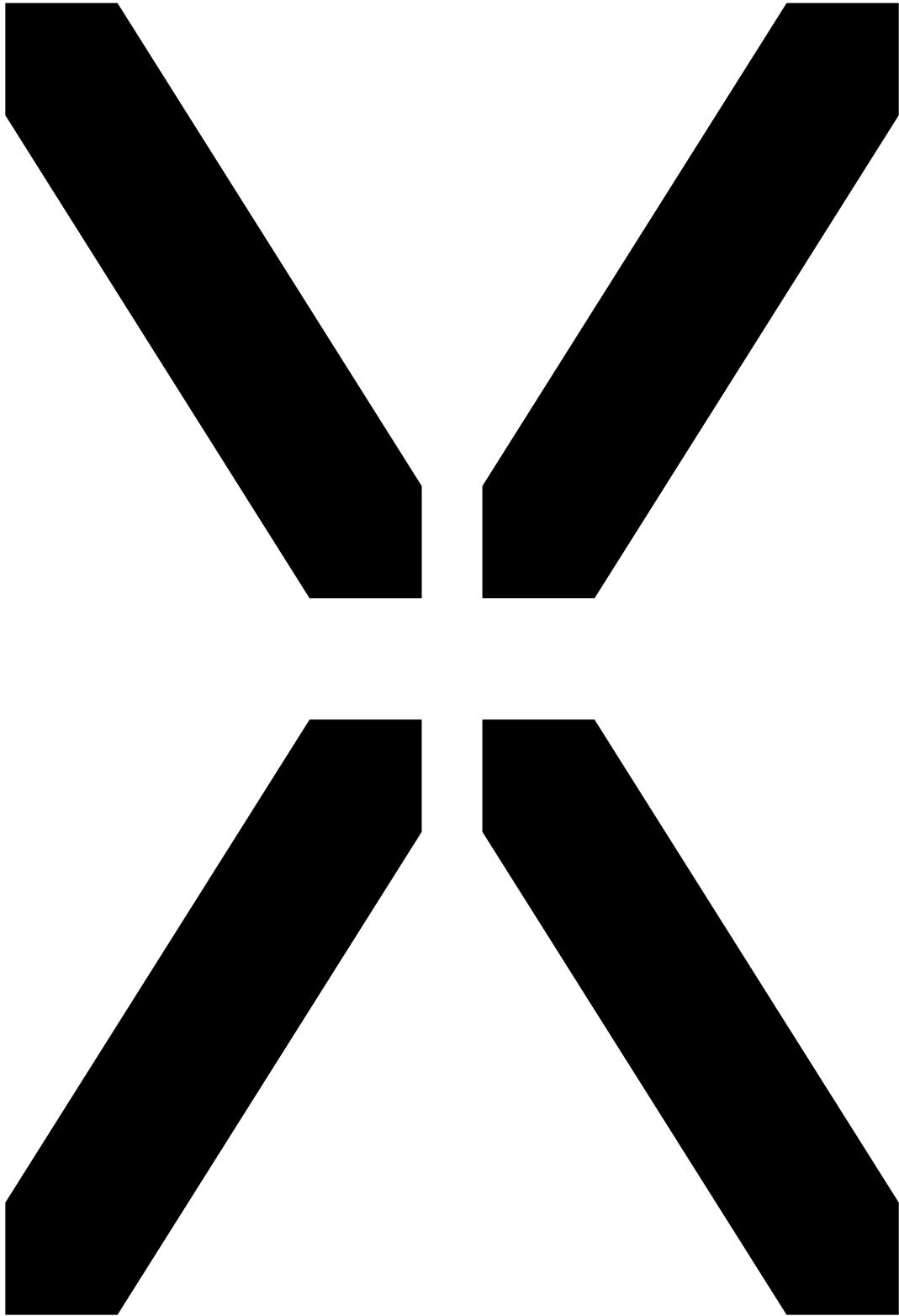


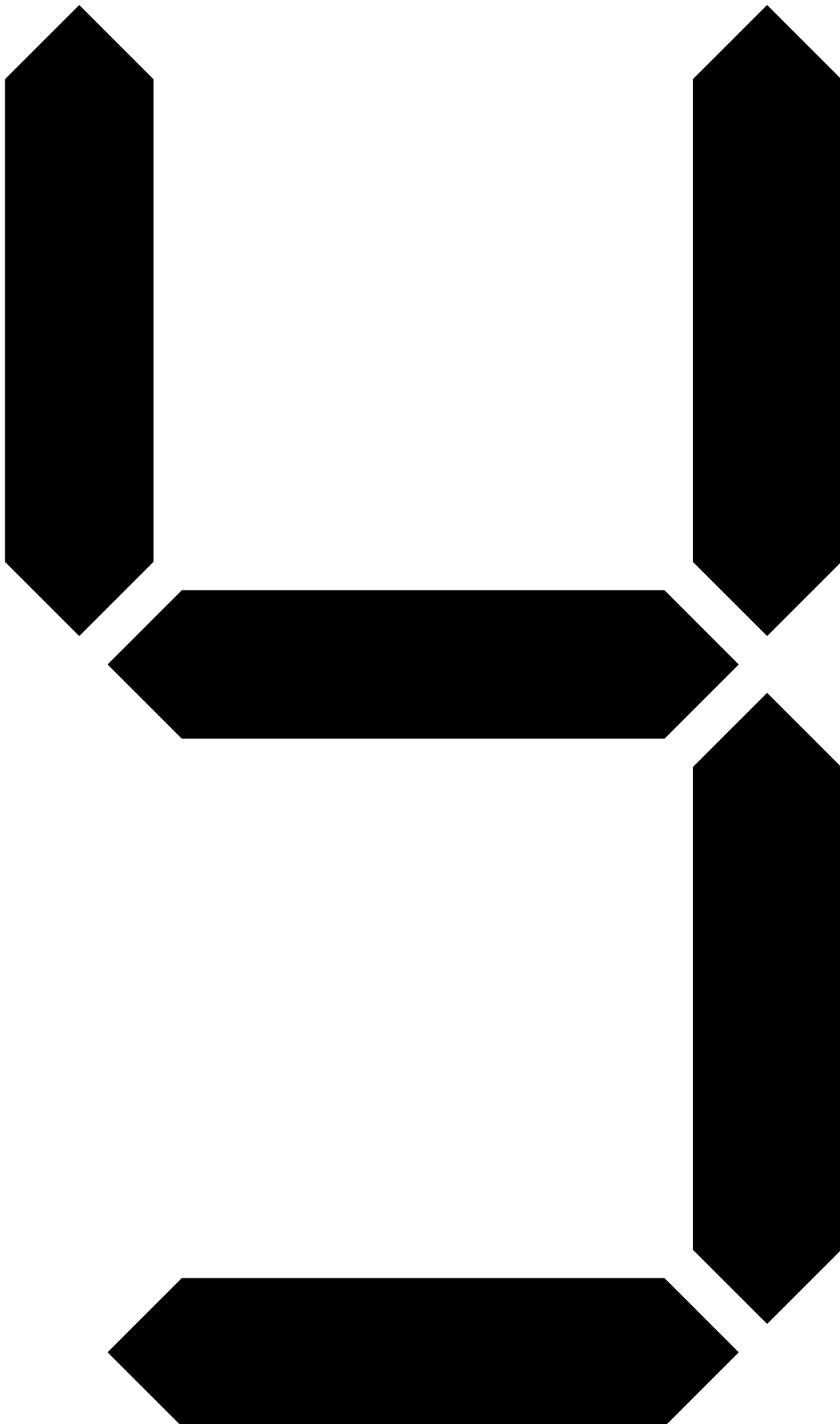


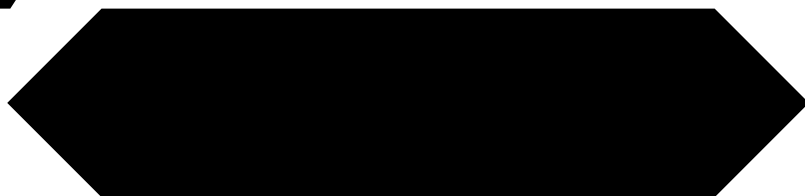
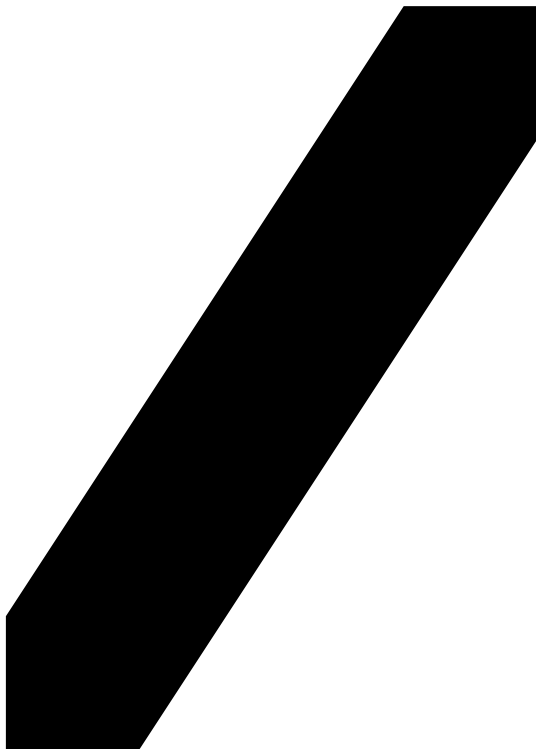
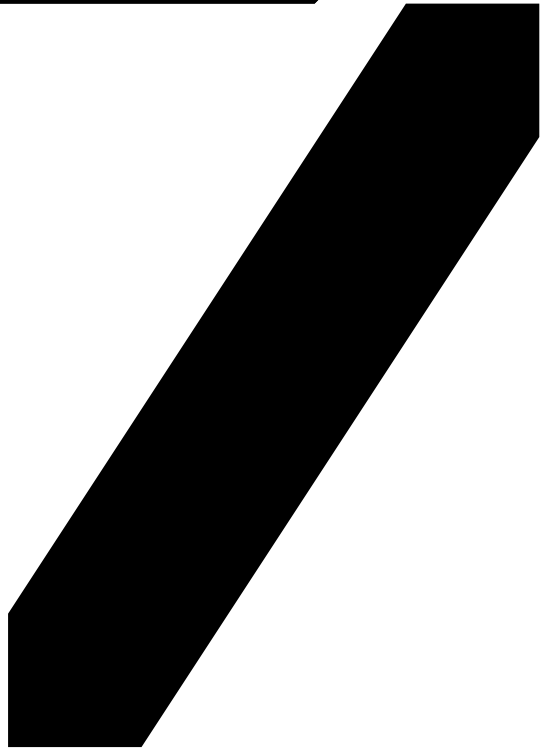









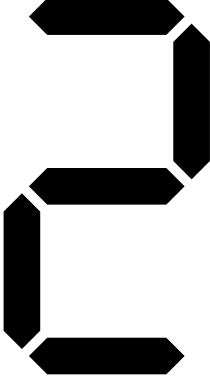
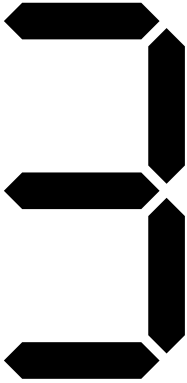
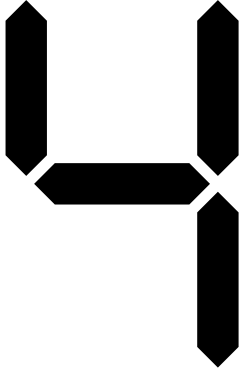
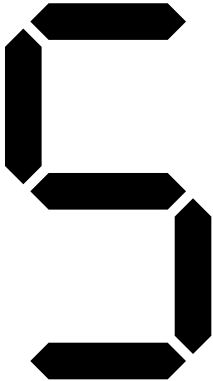
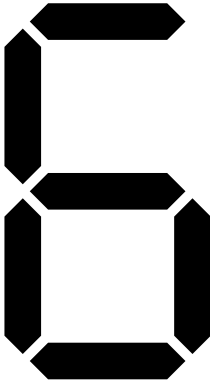
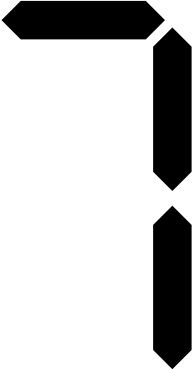
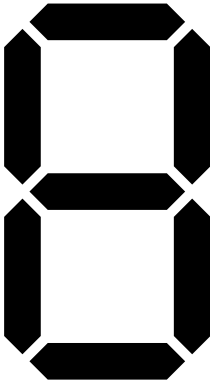


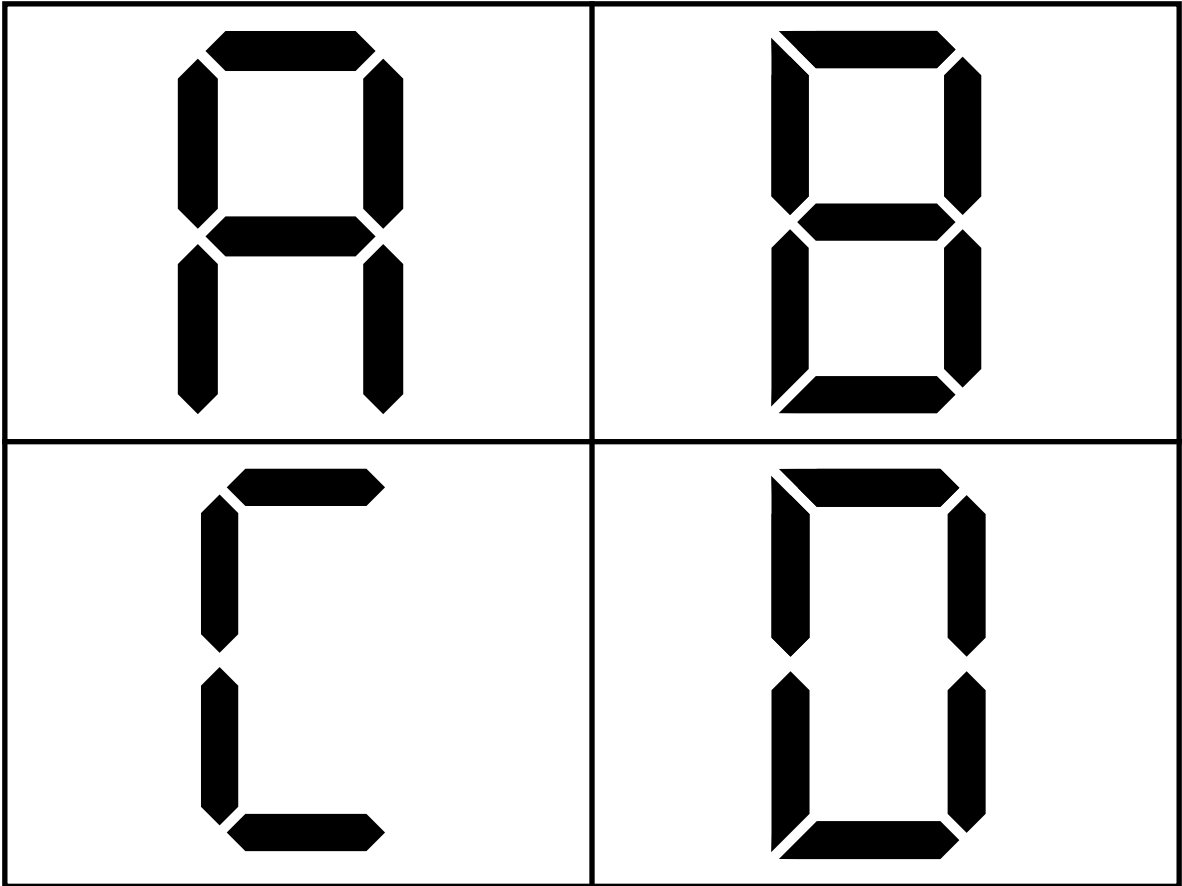




Appendix 4

E.a.R.L 2 mat challenge card - street scene challenge 1 (page 30)





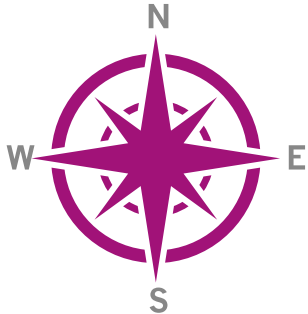
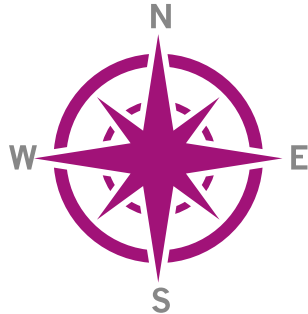


	
	
	
	





Appendix 5

E.a.R.L 2 mat challenge card - solar system challenge 4

<p>NORTH</p> 	<p>NORTH</p> 
 <p>EAST</p>	 <p>EAST</p>
<p>WEST</p> 	<p>WEST</p> 
 <p>SOUTH</p>	 <p>SOUTH</p>



Floor Robot Scheme of Work

Deliver exciting computer science lessons
suitable from EYFS through to KS2