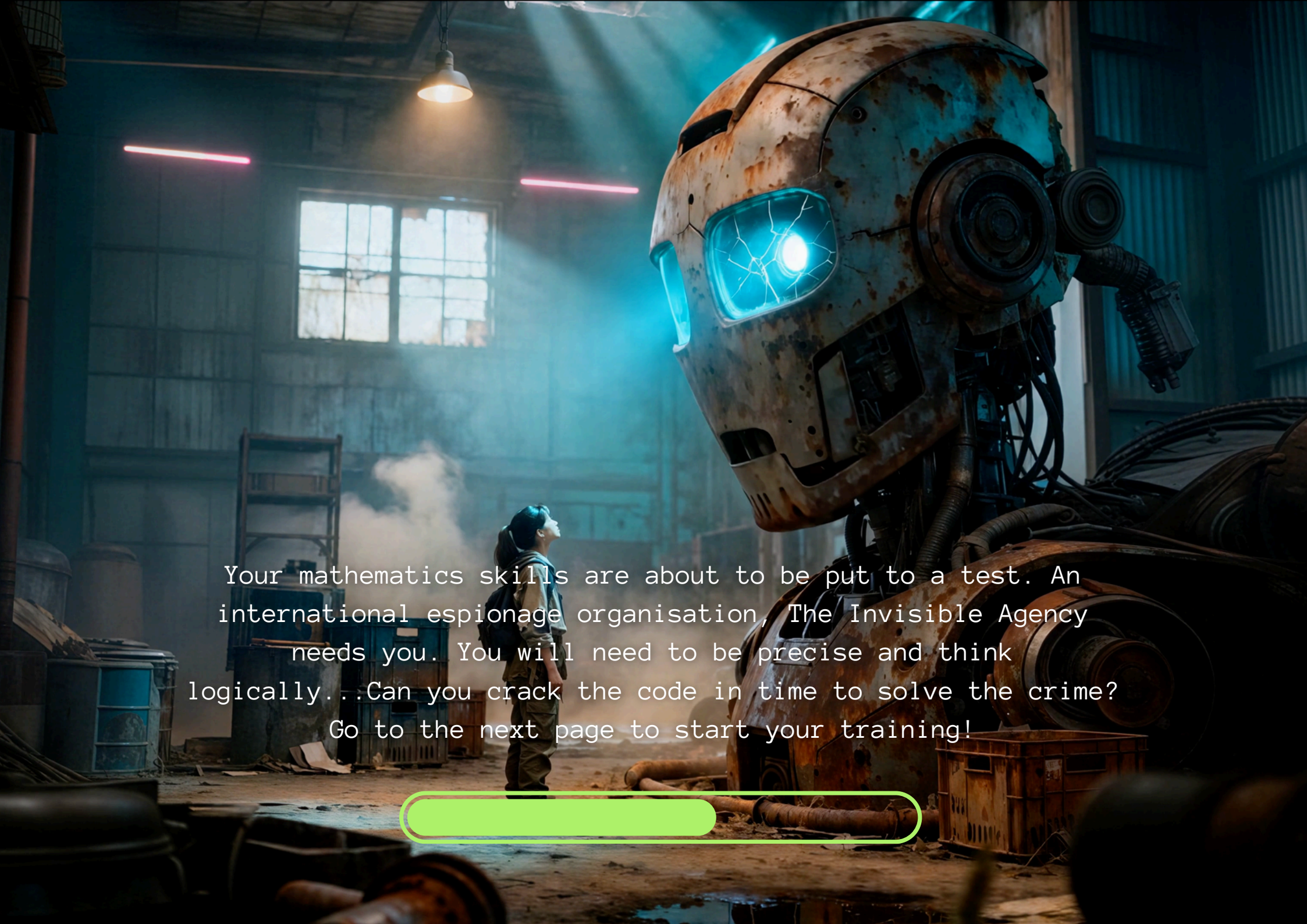


Your mathematics skills are about to be put to a test. An international espionage organisation, The Invisible Agency needs you. You will need to be precise and think logically...Can you crack the code in time to solve the crime? Go to the next page to start your training!

GATEWAYS THE CHALLENGE

2026 Training Activities

CHALLENGE 2 | YEARS 3 & 4

PRIORITY ACCESS CASE ****

Focus: Mathematics

Creator: Emma Carter

CONTENTS

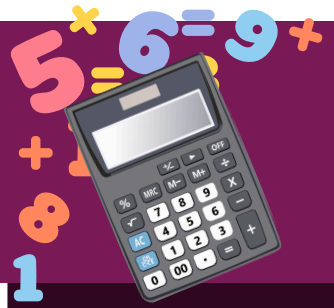
Click on the content you would like to view

Challenge 2 | Priority Access Case

Materials and resources needed for Challenge 2 training activities:

- **Paper and pencils for your ideas**

Please note that the training activities have been designed to be viewed as slides which increases the number of pages. Please consider the number of pages before printing.

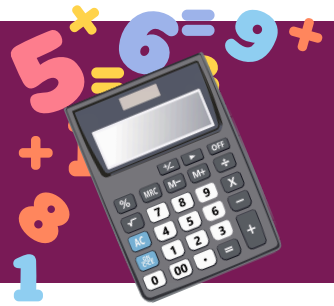


INTRODUCTION

Priority Accesss Case

Welcome, candidates. Your advanced mathematics teacher has selected your team to apply for The Invisible Agency, a secret organization that solves crimes involving international espionage. Your first task is to prove you have what it takes. During the Challenge you will need to unlock the Priority Access Case, a secure case locked with a four-digit code. Inside lies further tasks requiring your precision and logical thinking.

To succeed, you must master two essential concepts. First, Reverse Polish Notation (RPN): a stack-based system where numbers come before operations, like 3 4 + instead of 3 + 4. Second, functions: rules that take an input and produce an output. You will learn how to combine them through function composition—applying one function after another in a specific order. These skills will unlock the code and prove your readiness for The Invisible Agency.



REFERENCE

GLOSSARY

Come back to these glossary pages if you read any terms you are unsure of.

Post-fix notation or Reverse polish notation

A way of writing maths where you write the numbers first and the operation last.
Example: 3 4 + means 3 + 4

In fix notation

The usual way we write maths, with the operation between the numbers. **Example: 3 + 4**

Operand

A number that an operation acts upon. In 3 + 4, the operands are 3 and 4. In RPN, operands are pushed onto the stack **before** the operator appears.

Operator

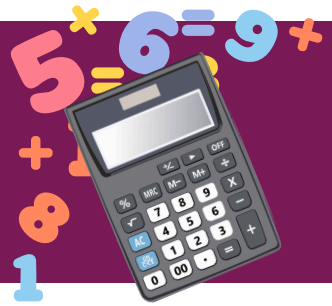
A symbol that tells you what calculation to perform. Common operators are **+**, **−**, **×**, and **÷**.

Function

A rule that takes a number in, does something to it, and sends a number out.

Function Notation

A short way to write a function. The letter is the function's name. The number in parentheses is the input. Example: f(5) means "apply function f to 5.".



REFERENCE

GLOSSARY

Come back to these glossary pages if you read any terms you are unsure of.

Last In, First Out (LIFO)

A simple rule: the last item you put onto a stack is the first one you take off.

Operate

In RPN: take the top two numbers off the stack, do a calculation (like add or multiply), then put the result back on top.

Push

To place a number onto the top of the stack.

Rule

The instruction inside a function that tells you what to do to the input.

Stack

A simple storage system that works like a pile of plates. You add to the top and remove from the top.

Output

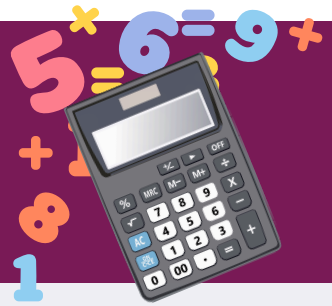
The number that comes out of a function after you give it an input.

Input

The number you put into a function.

Pop

To take the top number off the stack.



ACTIVITY 1

Introduction to Reverse Polish Notation

Adult assistance:
None required

Preparation time:
None

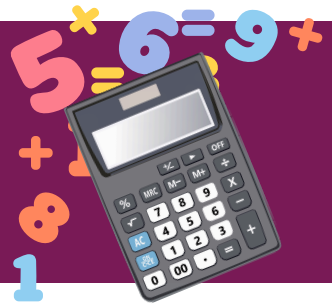
Activity time:
20 – 30 minutes

Materials required:
writing materials and printout of work sheet

How does a computer perform calculations?

Computers don't do maths the same way we do. Instead, they use a ‘stack’ of numbers and instructions.

This special system is called **Reverse Polish Notation** (RPN).
In this system the computer is first given the numbers and then told what to do with the numbers.



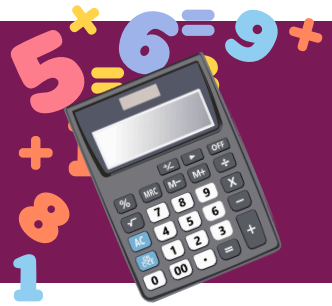
ACTIVITY 1

Introduction to Reverse Polish Notation

Why is Reverse Polish Notation good for computers?

Computers do one thing at a time, and follow exact instuctions. RPN is perfect for this.

When using Reverse Polish notation you never have to use brakets and you never need to know order of operations. So if everyone used RPN, you would **not need to use BODMAS or PEDMAS**!



ACTIVITY 1

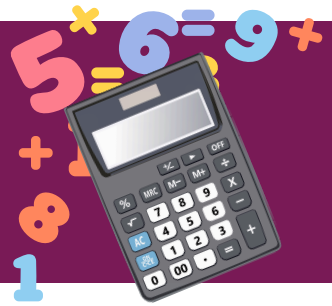
Introduction to Reverse Polish Notation

What Is a Stack?

Imagine a stack of pancakes:

- You cook one pancake and place it on a plate.
- You cook another and place it on top.
- You cook a third and place it on top.

When someone takes a pancake, they always take the top pancake — the last one added. You wouldn't grab a pancake from the bottom of the stack! **This is called "Last In, First Out" (LIFO). A stack in computing works exactly the same way.**



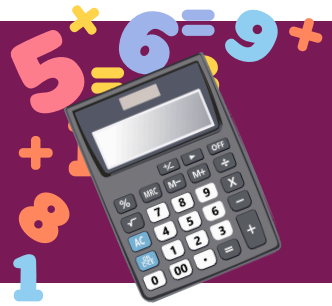
ACTIVITY 1

Introduction to Reverse Polish Notation

Stack Rules

In RPN, we use a stack to store numbers.
There are only **two** actions:

	1	2
ACTION	PUSH	OPERATES
WHAT YOU DO	Add a number to the top of the stack.	Remove the top two numbers, calculate, then PUSH the result back.



ACTIVITY 1

Introduction to Reverse Polish Notation

Example 1

Use Reverse Polish Notation to add two numbers:

STEP 1

PUSH 3 to the stack and cross out the 3 in the expression

~~3~~ 2 +

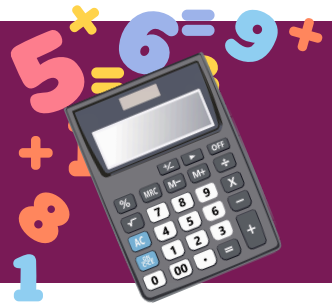
3

STEP 2

PUSH 2 to the stack

~~3~~ ~~2~~ +

2
3



ACTIVITY 1

Introduction to Reverse Polish Notation

Example 1 (continued)

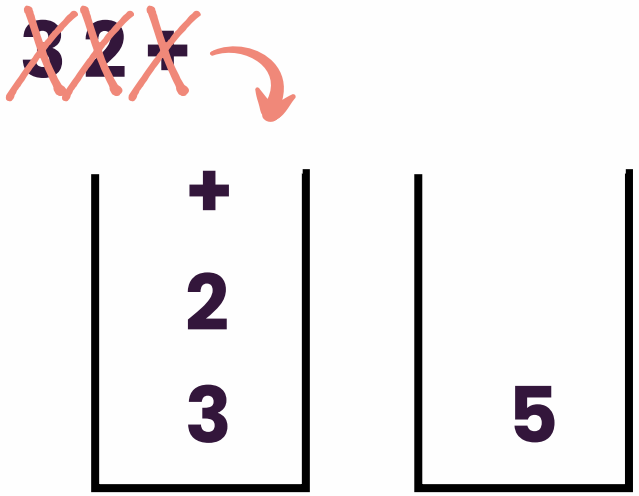
Use Reverse Polish Notation to add two numbers:

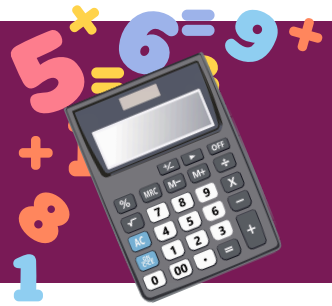
STEP 3

OPERATE: + on the stack which means perform the operation of adding.

You always add the **first number on the stack to the second number** on the stack.

After you add the 3 and the 2, the answer goes back onto the stack.





ACTIVITY 1

Introduction to Reverse Polish Notation

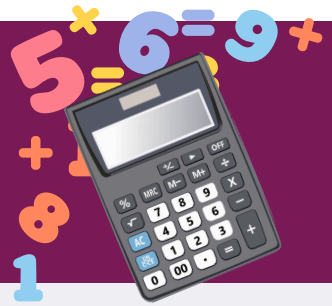
Now your try: make sure that everyone on your team can complete the worksheet questions using a stack.

1.1 5 7 +

1.2 12 7 −

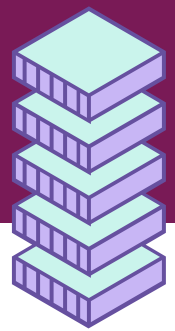
1.3 8 4 ÷

1.4 20 5 ÷



ACTIVITY 2

More complex Reverse Polish Notation



Adult assistance:
None required

Preparation time:
None

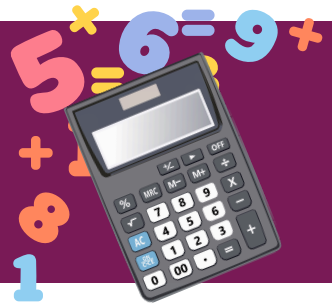
Activity time:
40 minutes

Materials required:
Writing materials and printout of work sheet

We can make longer expressions with **Reverse Polish Notation**. Let's go though an example:

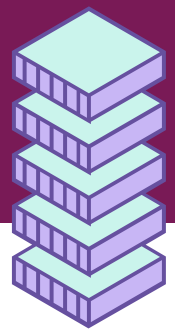
Example 2
Reverse Polish Notation for the following expression:

4 3 + 2 ×



ACTIVITY 2

More complex Reverse Polish Notation



Example 2 (continued)

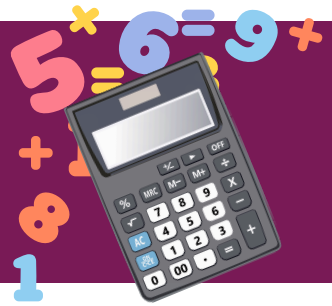
4 3 + 2 ×

Push 4, then push 3 to the Stack and then perform the operation of adding, and put the answer back on the stack.

~~4~~ ~~3~~ ~~+~~ 2 ×

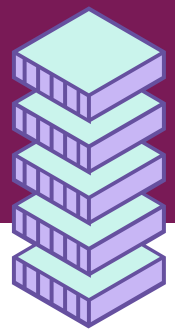
+	
3	
4	7

example continued next page...



ACTIVITY 2

More complex Reverse Polish Notation



Example 2 (continued)

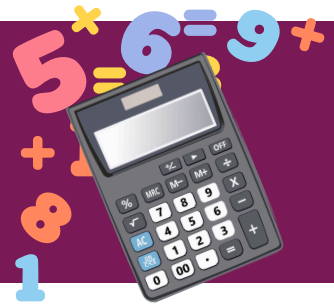
4 3 + 2 ×

Push 2 then next we have the multiply operator.
Perform the multiply operator with the two numbers that are next in the stack.

~~4 3 + 2~~ ×

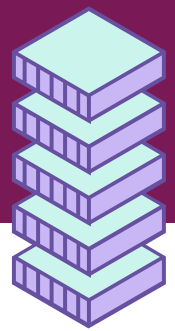
× 2 7	14

So in this case the answer is 14



ACTIVITY 2

More complex Reverse Polish Notation



Example 3

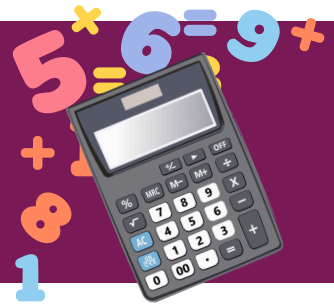
Reverse Polish Notation for the following expression: 4 8 + 1 3 + ÷

Push 4, then 8 to the Stack and then perform the operation of adding, and put the answer back on the stack.

~~4~~ ~~8~~ ~~+~~ 1 3 + ÷

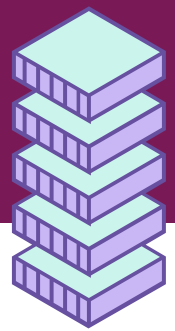
+	
8	
4	12

example continued next page...



ACTIVITY 2

More complex Reverse Polish Notation



Example 3 (continued)

4 8 + 1 3 + ÷

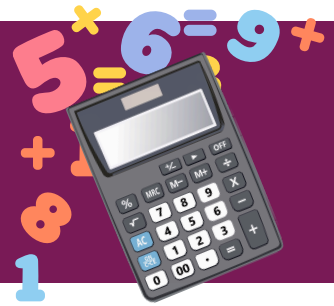
Push 1, then 3 to the Stack and then perform the operation of adding with the next two numbers on the stack, and put the answer back on the stack.

~~4~~ ~~8~~ ~~+~~ ~~1~~ ~~3~~ ~~+~~ ÷

+	
3	
1	4
12	12

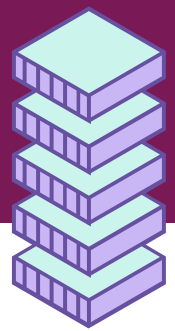
Notice that the third number on the stack, the 12, is unchanged, only the two numbers under the + operation are involved.

example continued next page...



ACTIVITY 2

More complex Reverse Polish Notation



Example 3 (continued)

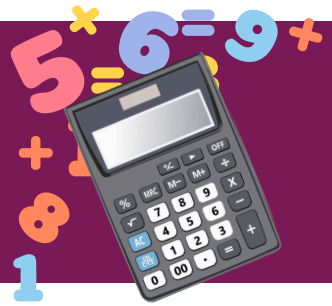
4 8 + 1 3 + ÷

Division Operation:

Always divide the second number (lower in the stack) by the first number (top of the stack), so in this case we are dividing the 12 by 4. The answer is pushed back on the stack and we have finished.

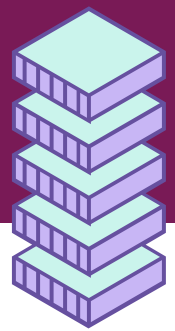
~~4~~ ~~8~~ ~~+~~ ~~1~~ ~~3~~ ~~+~~ ~~÷~~

÷	
4	
12	3



ACTIVITY 2

More complex Reverse Polish Notation



Example 4

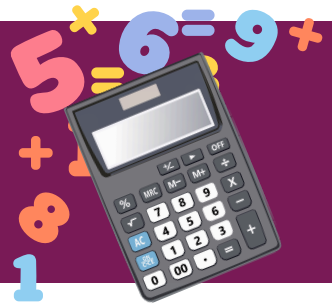
Reverse Polish Notation for the following expression: 12 8 1 3 + + ÷

Push all the numbers to the Stack until you get to an operation. This time we push four numbers to the stack. After that, we can perform the adding operation with the top two numbers (in this case the 3 + 1).

~~12~~ ~~8~~ ~~1~~ ~~3~~ ~~+~~ + ÷

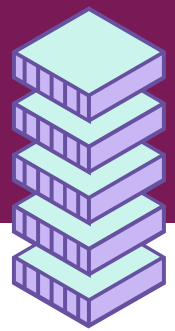
+	
3	
1	4
8	8
12	12

example continued next page...



ACTIVITY 2

More complex Reverse Polish Notation



Example 4 (continued)

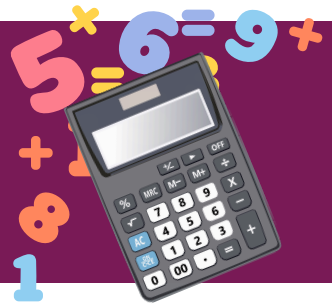
12 8 1 3 + + ÷

The next part of our expression is another operator, the plus. So we add the next two numbers on the stack and push the answer (which is 12) back on the stack.

~~12 8 1 3 + + ÷~~

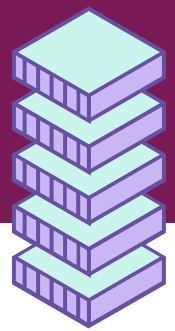
+	
4	
8	12
12	12

example continued next page...



ACTIVITY 2

More complex Reverse Polish Notation



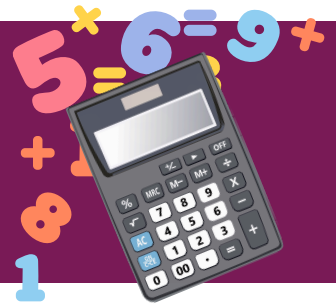
Example 4 (continued)

12 8 1 3 + + ÷

The final part of our expression is the division operator. We divide the second number by the first number. And our answer is 1.

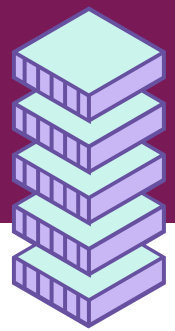
~~12 8 1 3 + + ÷~~

÷	
12	
12	1



ACTIVITY 2

More complex Reverse Polish Notation



2.1 Your turn — choose the correct one

Which one of the following is the correct way to calculate: **7 4 - 2 +**

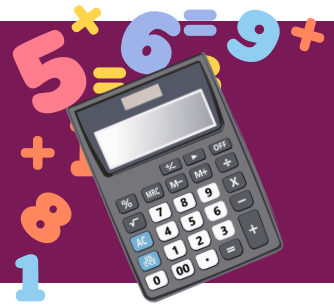
One of these is incorrect. Your task is to choose the correct one.

A)

-	+	
4	2	
7	3	5

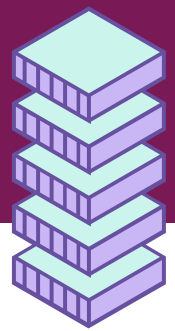
B)

-	+	
4	2	
7	-3	-1



ACTIVITY 2

More complex Reverse Polish Notation



2.2 Choose the correct one

Which one of the following is the correct way to calculate: **7 4 2 - +**

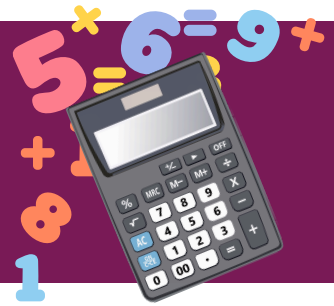
One of these is incorrect. Your task is to choose the correct one. Notice the stack in the middle!

A)

-	+	
2	5	
4	4	
7		9

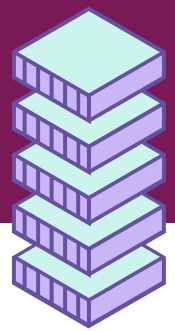
B)

-	+	
2	2	
4	7	
7		9



ACTIVITY 2

More complex Reverse Polish Notation

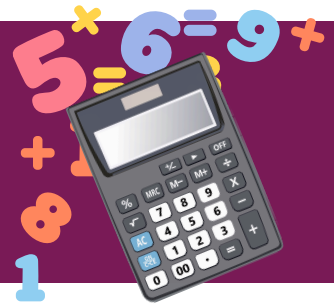


2.3 Find the mistake

Find the mistake in the following calculation of: 3 9 + 4 2 + ÷

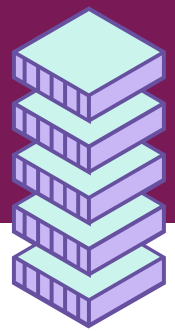
+	+	÷	
9	2	6	
3	4	12	0.5
	12		

2.4 What should the correct answer be?



ACTIVITY 2

More complex Reverse Polish Notation

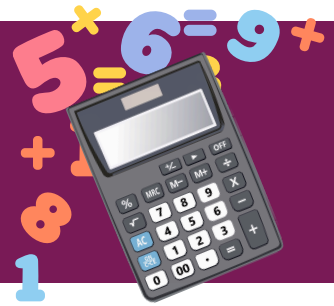


2.5 Find the mistake

Find the mistake in the following calculation of: **3 1 - 6 6 + ÷**

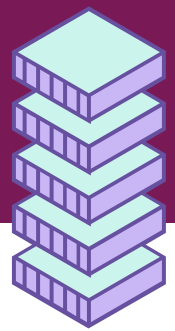
-	+	÷	
1	6	12	
3	2	2	6

2.6 What should the correct answer be?



ACTIVITY 2

More complex Reverse Polish Notation



Calculate the answer for each of the following:

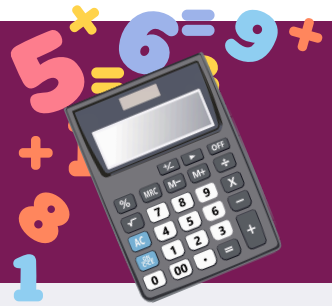
2.6 3 1 6 6 − + ÷

2.7 3 1 4 + ×

2.8 9 4 − 5 +

2.9 11 8 − 4 4 + ×

Congratulations on solving some very difficlt Reverse Polish Notation Problems!



ACTIVITY 3

Functions

Adult assistance:
None required

Preparation time:
None

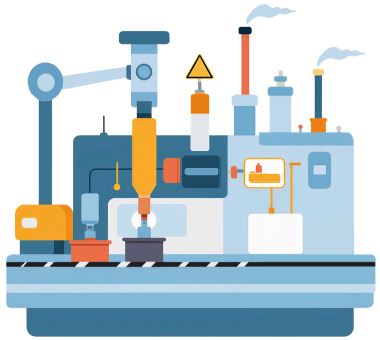
Activity time:
20 – 30 minutes

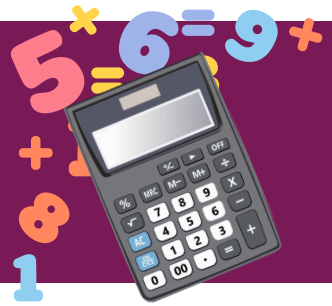
Materials required:
Writing materials and printout of work sheet

What is a mathematical function?

A mathematical function takes an input and produces an output. In this Challenge, both inputs and outputs will be numbers.

Think of a function as a machine with a rule. You feed a number into the machine, the machine applies its rule, and a new number comes out.





ACTIVITY 3

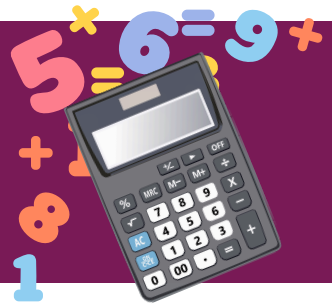
Functions

For example

Consider a function whose rule is "add three." Feed in 1, and the machine outputs 4. Feed in 2, and it outputs 5.

We write a function like this: **f**(input) = output

The letter **f** names the function. The value inside the parentheses is the input. The expression after the equals sign tells you the rule.



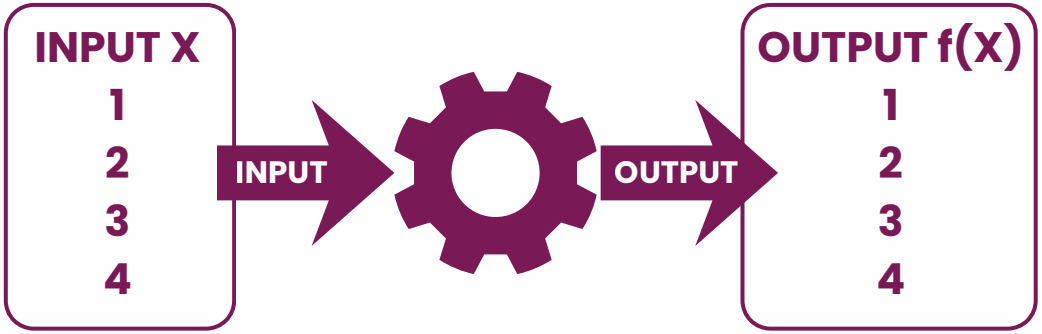
ACTIVITY 3

Functions

If **f** has the rule **'add three'**, we can write:

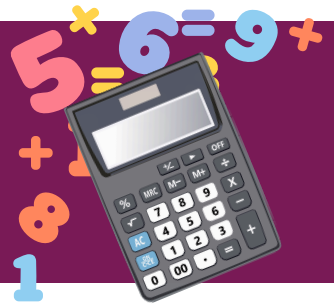
$$f(x) = x + 3$$

Here, x stands for any input we choose. Once we pick a specific input, we can calculate the output.



Functions can also combine multiple operations. For example, a function might double the input and then subtract one. That rule would be written as:

$$g(x) = 2x - 1$$



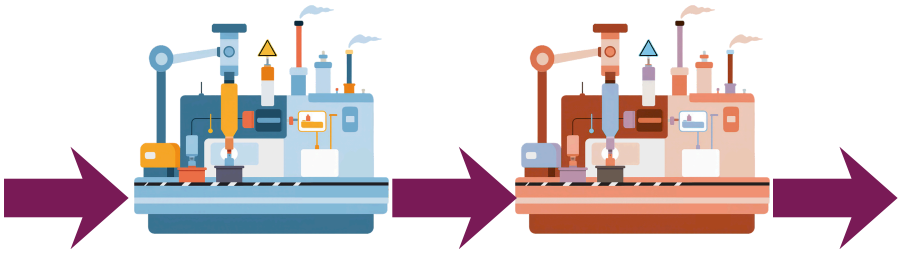
ACTIVITY 3

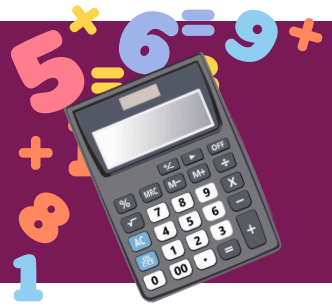
Functions

What are functions of functions?

Sometimes you want to apply one function, then apply a second function to the result. This is called function composition, but you can also think of it as "functions of functions."

Imagine two machines connected in a line. The first machine takes an input and produces an output. That output becomes the input for the second machine. The second machine applies its own rule and produces a final output.

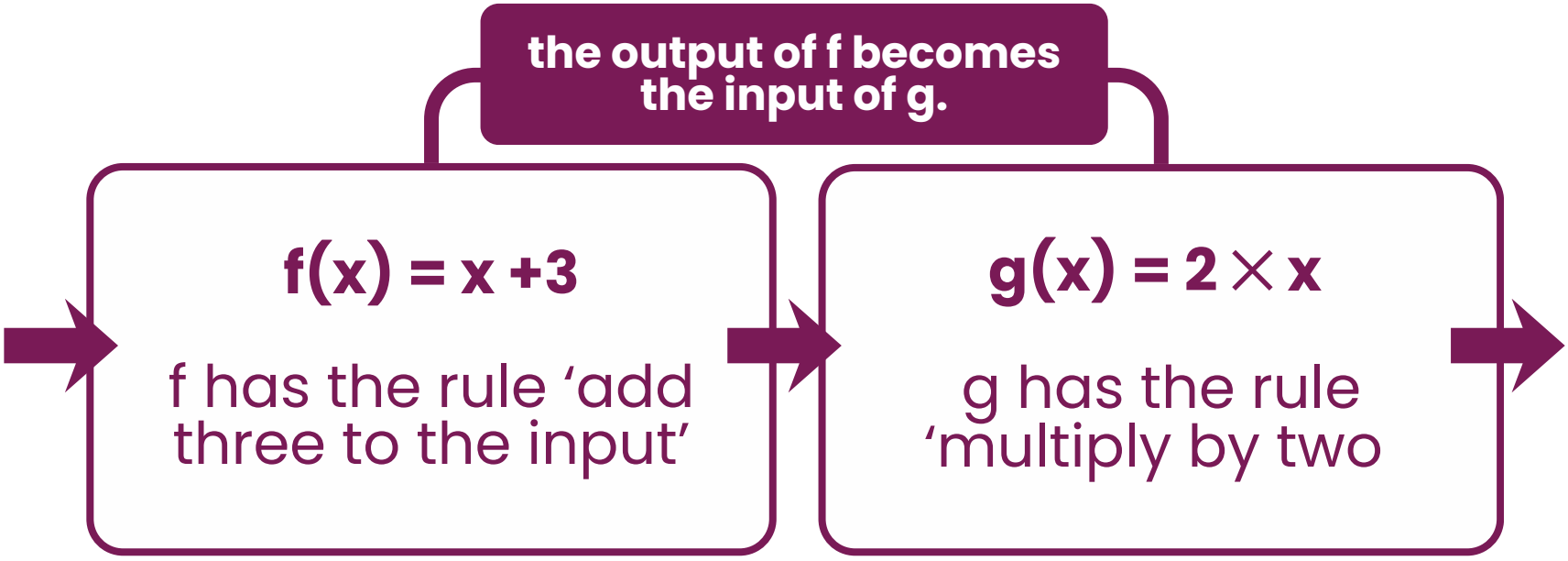




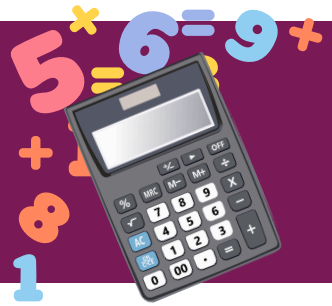
ACTIVITY 3

Functions

We can replace these machines with mathematical functions:



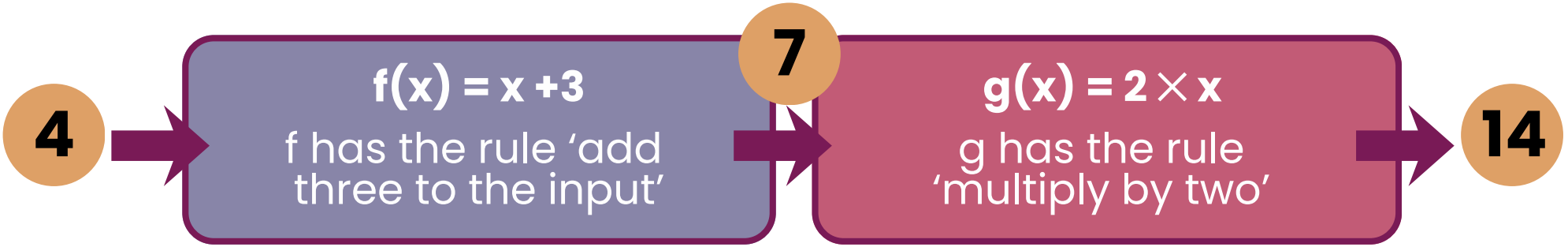
This means: "Take x, put it into f, then take the result and put it into g."



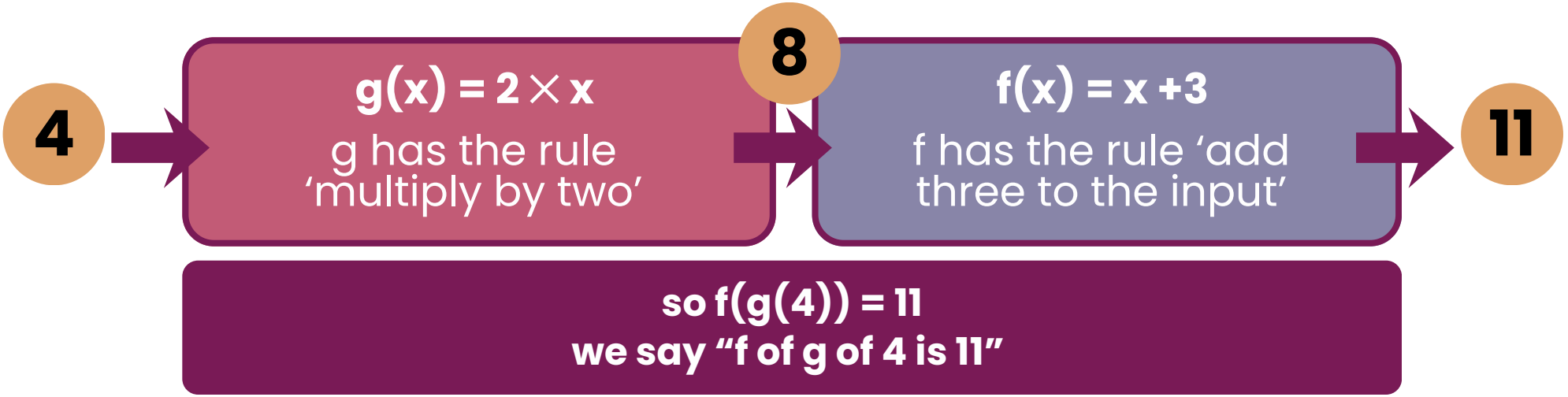
ACTIVITY 3

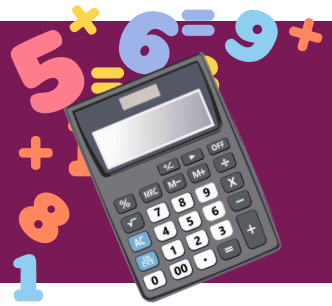
Functions

The order of the functions matters. Let's test what happens with x = 4.



so g(f(4)) = 14, If we **reverse** the order and do **g first** and **then f** we get:





ACTIVITY 3

Functions

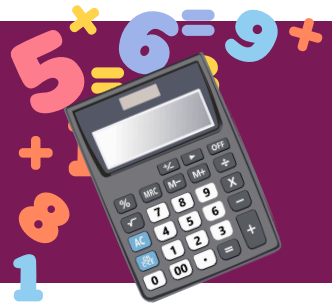
Functions of functions allow you to build a complex set of rules from simple building blocks. This is a powerful idea in mathematics, computer programming and ... in the Invisible Agency Challenges you are preparing for. Your turn:

3.1 The function f(x) is to multiply by two, in other words: **f(x) = 2 × x**

calculate f(7)

3.2 The function g(x) is to add three. In other words: **g(x) = x +3**

calculate g(7)



ACTIVITY 3

Functions

3.3 If $f(x) = 3 \times x$ and $g(x) = x + 1$, **then what is f(g(5))**?

Do the inside function first — so g(5) comes first.
You might like to think of it as:

g(x) = x + 1
g has the rule 'multiply by two'

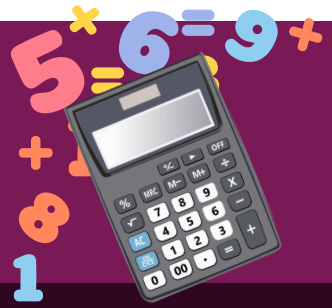
→

f(x) = 3 × x
f has the rule 'add three to the input'

→

3.2 If $f(x) = 2 \times x$ and $g(x) = x + 3$, **then what is g(f(4))**?

Do the inside function first — so f(4) comes first.



GOOD LUCK TEAMS!

Your Training Is Over

You have trained like a true candidate for The Invisible Agency. You now understand how computers think using Reverse Polish Notation, how to track numbers through a stack with precision, and how to build complex rules by combining functions. These are not just mathematical tricks—they are the foundations of logical thinking, programming, and code-breaking.

When you face the Priority Access Case in the Challenge, remember what you have practised here. Work through each RPN expression step by step. Track your stack carefully. Apply functions in the correct order. Your team has the skills. Now trust them. The code is waiting to be unlocked.