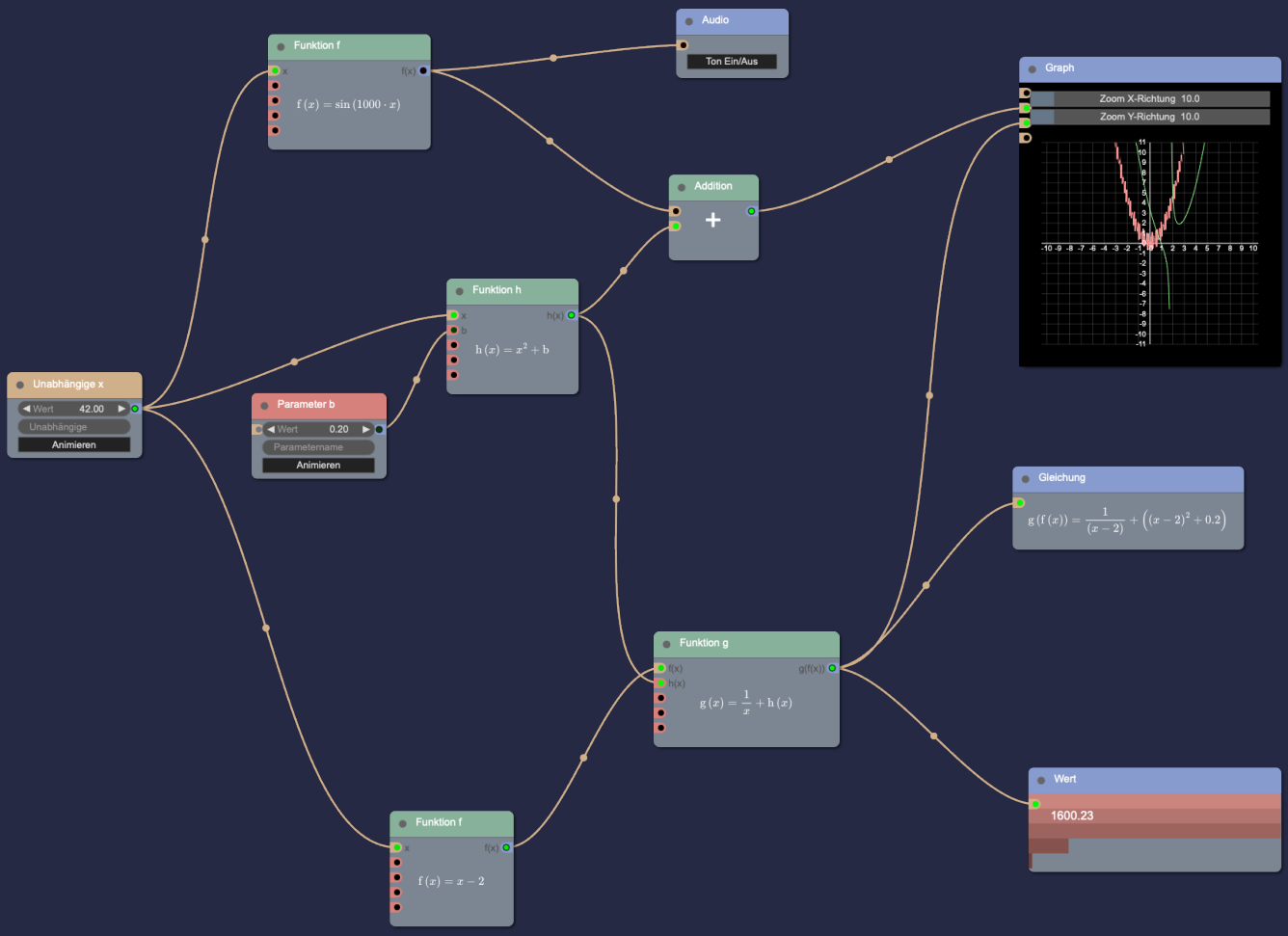
Math-Nodes Workbook

PLAYFUL LEARNING WITH FUNCTION MACHINES

A PROJECT BY NICOLAS REGEL



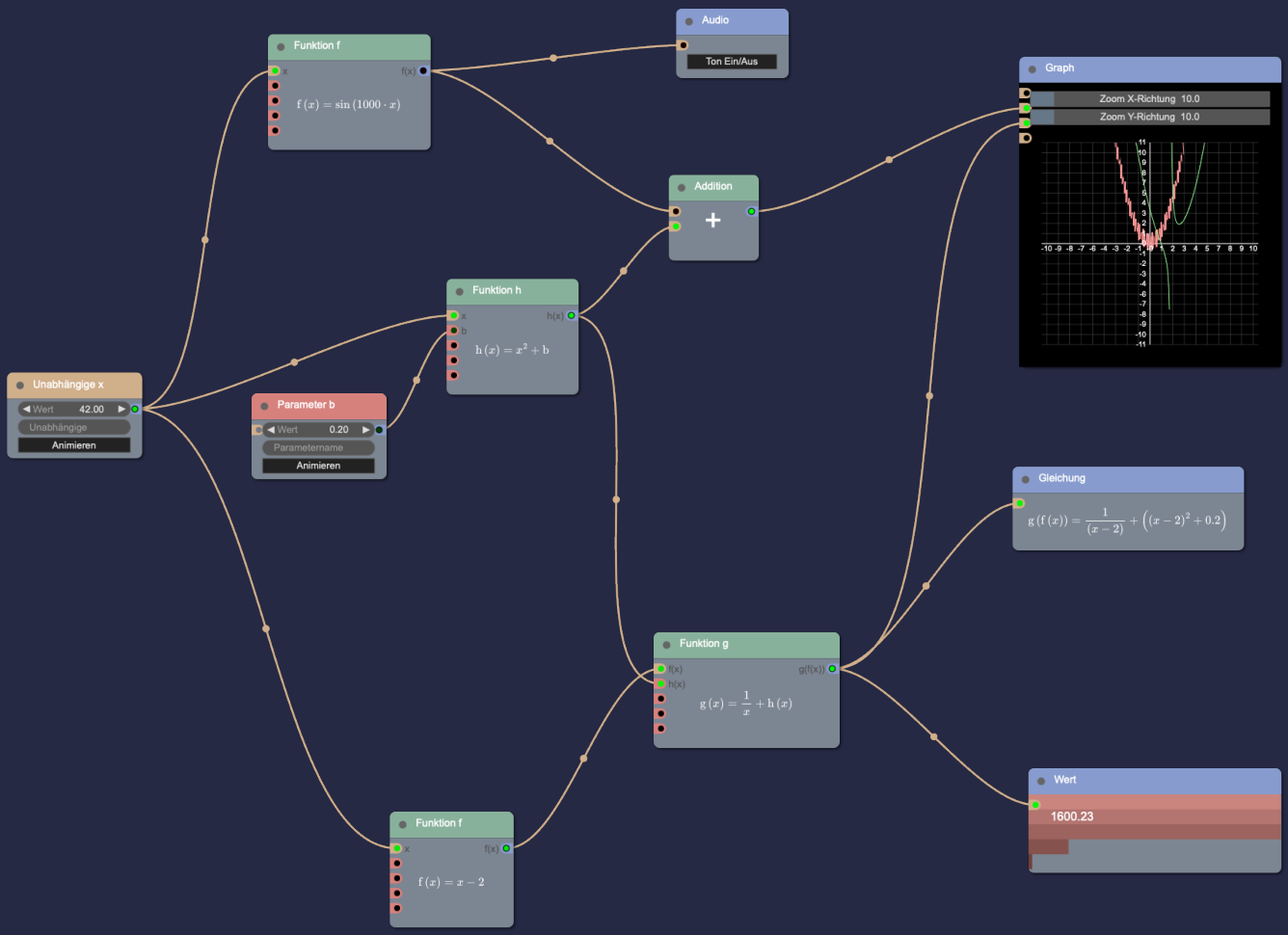
Task Development:	Nicolas Regel, Paul Busse
Concept:	Nicolas Regel
Editing:	Nicolas Regel, Paul Busse
Layout:	Nicolas Regel
Feedback, Correction:	Andrea Hoffkamp, Lisa Nickolaus, Michael Schröder, Laura Degenhardt

Acknowledgments:
Heartfelt thanks go to all colleagues, students, and supporters who contributed ideas, feedback, and motivation to the creation of this workbook.

Inhaltsverzeichnis

1. Cryptography

Securing secret messages with functions and codes - A chapter by Samuel Weber



Mathematics as the key to security

Encryption is a technique for keeping information confidential. It has been used for centuries. In the age of the Internet, encryption is omnipresent.

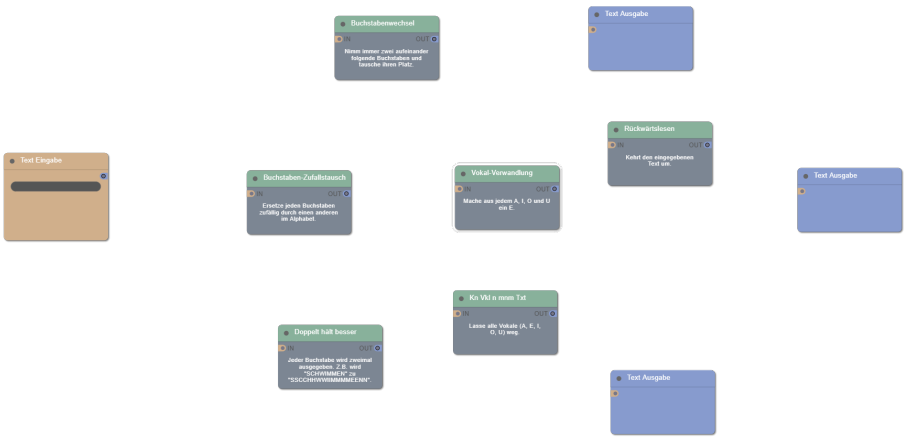
But how does encryption work? What are the fundamentals of cryptography, the science of encryption? And are there limits to encryption?

1 | Task 1a: What is encryption? - The world of Word Machines

You are already familiar with a similar task from the chapter on word machines.

a) Examine the different machines. What happens to your message as it passes through the different cards?

Think about this: In addition to encryption, the goal is also to ensure that the encrypted message can be decrypted without a doubt. Is this possible without a doubt for all cards here? Why (not)?



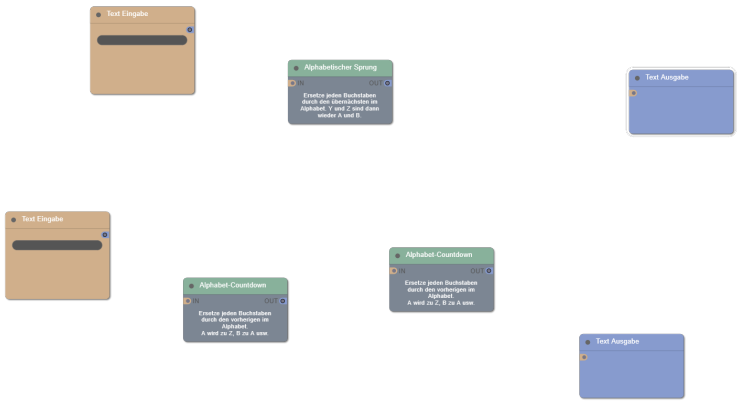
2 | Task 1b: The Caesar Cipher

Ciphers transform messages. One of the most famous ciphers was used over 2,000 years ago - the Caesar cipher. In this process, each letter of the alphabet is shifted by a fixed number of positions.

Build a replica of the Caesar machine using the card titled **Alphabetic Jump**. What happens to the word MATH when each letter is shifted by 2 positions?

Is it possible to decrypt this cipher without a doubt? In Math-Nodes, there is a card for this called **Alphabet Countdown**. Why is that so important in cryptography? And how many of the cards do you have to link together to get your original word back?

Good to know: The Caesar Cipher is what is known as a **monoalphabetic substitution cipher**. This means that only one cipher alphabet is used for encryption, which in this case is shifted by 2 positions relative to the original.

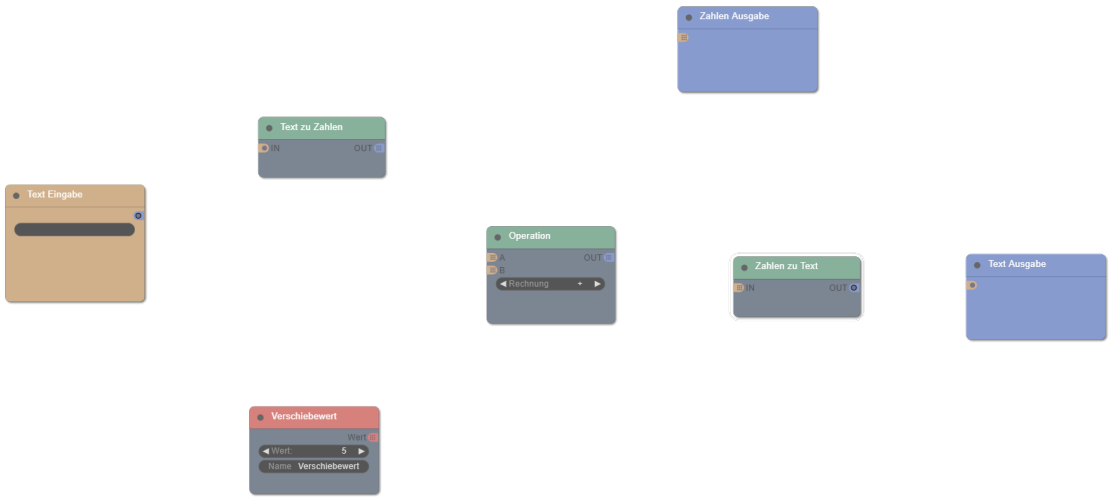


3 | Task 2a: The Caesar Cipher with variable shift

Until now, the shift was fixed in the machine. Now we use a **controller**, parameter **a**, for this. In order for the machine to be able to perform calculations with letters, they must first be converted into numbers. To do this, connect the **Text to Numbers** card to an **Operation** (addition) and connect the parameter to it as well.

X-ray vision: You can also connect the **Numbers Output** card to the output of the operation. This allows you to look directly inside the machine.

What happens to the word MATH when you change the value on the controller? Observe both the number and text outputs.



4 | Task 2b: The Modulo Clock - Arithmetic with Remainders

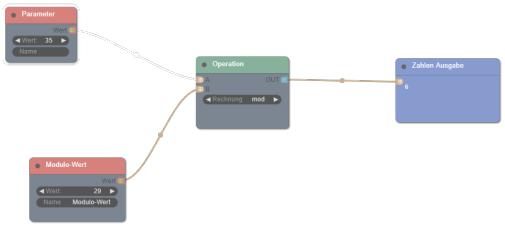
Addition alone, as in **Problem 2a**, is not enough: If the result is greater than 28, the letter falls out of our alphabet. This is where the **modulo operation** comes in. Examine the modulo principle in detail. The **Operation** card has two inputs: **A** and **B**. To understand the clock arithmetic, you need to figure out how these two interact.

Experiment with the two parameters.

1. Set the controller **B to 29** and change **A**. What happens to the number 29?

2. Now set **A to a fixed number** and change **B**. What do you notice?

a)



The Modulo Calculation:

In modulo calculation, the number is divided by the specified number, and the remainder is displayed. Think of modulo calculation as a clock.

In Math-Nodes, our alphabet consists of a total of 29 characters. This is why we perform calculations here using **mod 29**. As a result, the machine operates only in the range from 0 to

28. If a number exceeds 28 during a calculation, the counter wraps around past 28 and simply starts over at 0. This ensures that every number can ultimately be assigned to a valid character. Mathematically, the result is simply the remainder of a division.

For example: $50 : 29 = 1 Remainder 21$

So: $50 \pmod 2 9 = 21$

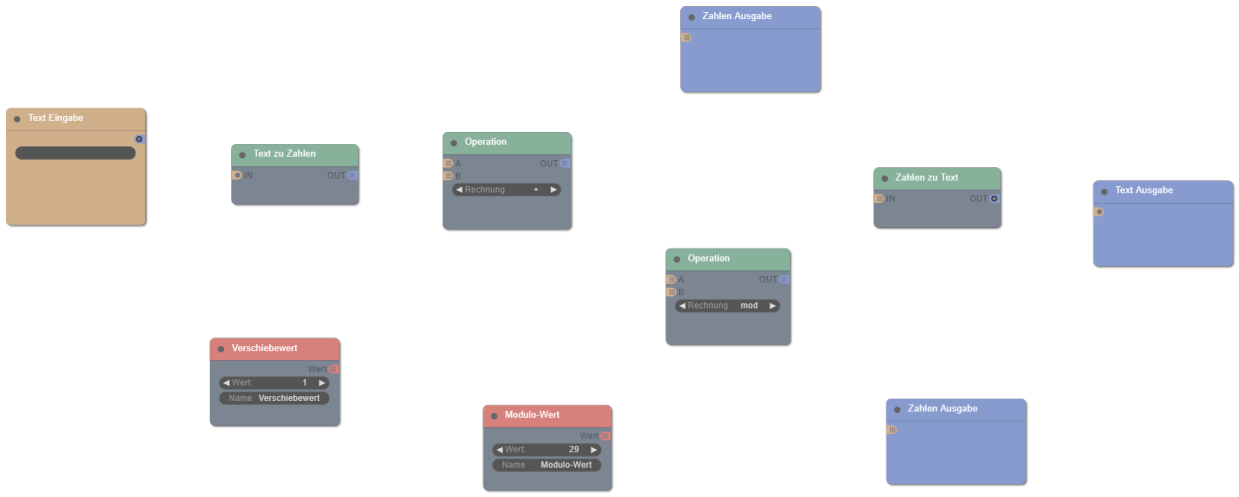
5 | Task 2c: The Complete Caesar Cipher - Exploring the Alphabet

Now you need to put your problem-solving skills to the test: Assemble all the parts to create a complete encryption machine.

Connect the cards so that your text is first converted into numbers, then added with the parameter **a**, secured using the modulo operation, and finally converted back into text.

Secret Hunt: There are 29 characters in Math-Nodes. Can you figure out which characters are hidden behind the numbers 26, 27, and 28? Use the controller **a** to find out. Make sure the modulo value (Input B) is set to the correct clock size.

a)



Think about this:

The most common letter in the English language is **E**.

So which word was most likely encrypted here, and what is the value of the parameter **a**?

JMMSMMXMZ.

Check your answer using the machine.

The Polyalphabetic Substitution

So far, the key has been the same for the entire word. But what happens if the key changes for each letter?

This is how a "polyalphabetic substitution" works. It is based on multiple alphabets, and each letter is shifted differently.

In the image titled **Vigenère Cipher** in the workbook, you can see what happens when one letter is encrypted using another letter.

6 | Task 3: Dynamic keys - The Vigenère Encryption

The Caesar cipher has a major weakness: Since each letter is always shifted by the same amount, attackers can easily crack it. The solution: A changing **keyword**!

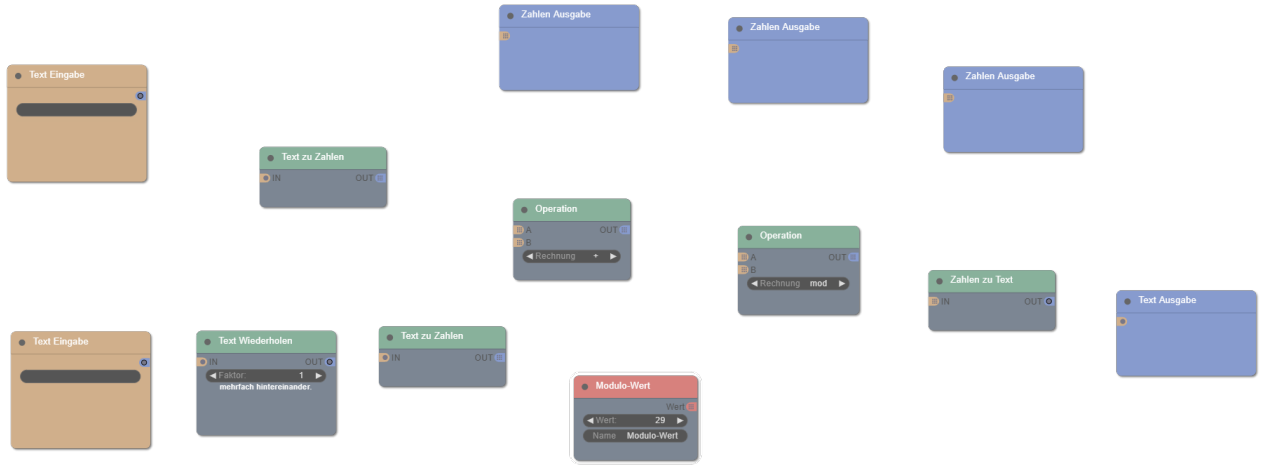
Here you can see a system in which two messages come together: your text and a key. Connect the machines so that the numerical values of both words are added together.

The length trick: What happens if your key is shorter than your text? Use the **Repeat Text** card to repeat the key as many times as necessary until it covers every character in your message.

To test: What is the result of encrypting the word MATH with the key ABCD?

Reference in the workbook: To help you understand the encryption better or check your answer manually, you will find an overview of the Vigenère square in the workbook. But be careful, because it consists only of the 26 letters.

a)



b)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

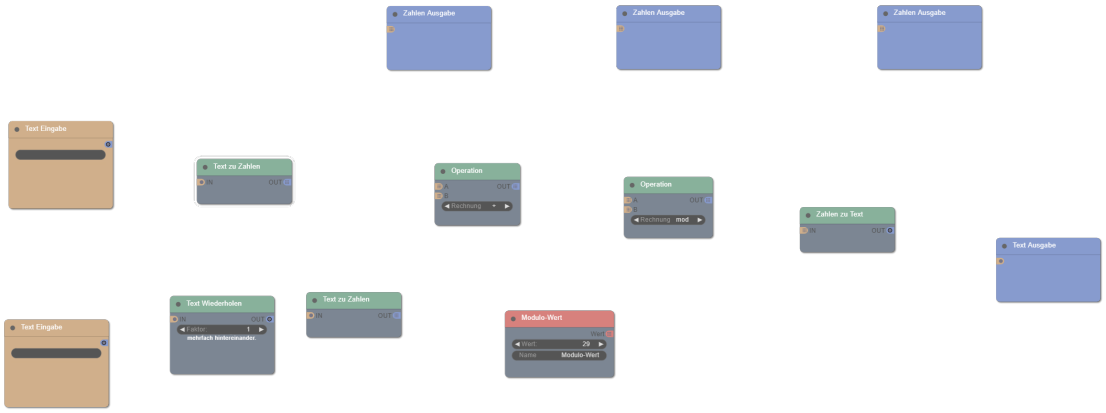
7 | Task 4: Finding the way back - The Vigenère Decryption

Encryption is one thing, but how does the intended recipient get the original message back? To do this, you have to logically reverse the encryption process.

All the components are here, but they are not yet connected and have been reset to their default settings. Build the decryption machine based on the model from Exercise 3.

Logic Check: Which mathematical operation do you need to select on the **Operation** card to undo an addition?

Test: Can you decrypt your message from the previous task? Use your ciphertext and the corresponding key. Do you get MATH back?



Modern Encryption Methods

So far, you have hidden messages by shifting letters. But modern computers use a different method: They first convert any text into numbers and then pass those numbers through one-way mathematical functions.

The RSA algorithm is at the heart of modern internet security. It works like a lock that anyone can lock, but only the owner of the correct key can unlock again. Now we want to build this lock ourselves.

In theory, you could start here the same way you did in the previous exercises and enter a word. For ease of explanation, we will start directly with the number. However, you could imagine that your message contains, for example, a D, which is converted to a 4 and then at the end, the number is converted back into a letter.

8 | Task 5: Modern Encryption - The RSA One-Way Street

The RSA algorithm uses a public key for encryption, which anyone can know.

Step 1: The Foundation. Choose two small prime numbers p and q (e.g., 3 and 11) and enter them into the parameter cards. Link them to the cards for the **Free Calculation rule** so that the machine calculates the modulus $n = p \cdot q$ and the phi function $\phi = (p-1) \cdot (q-1)$.

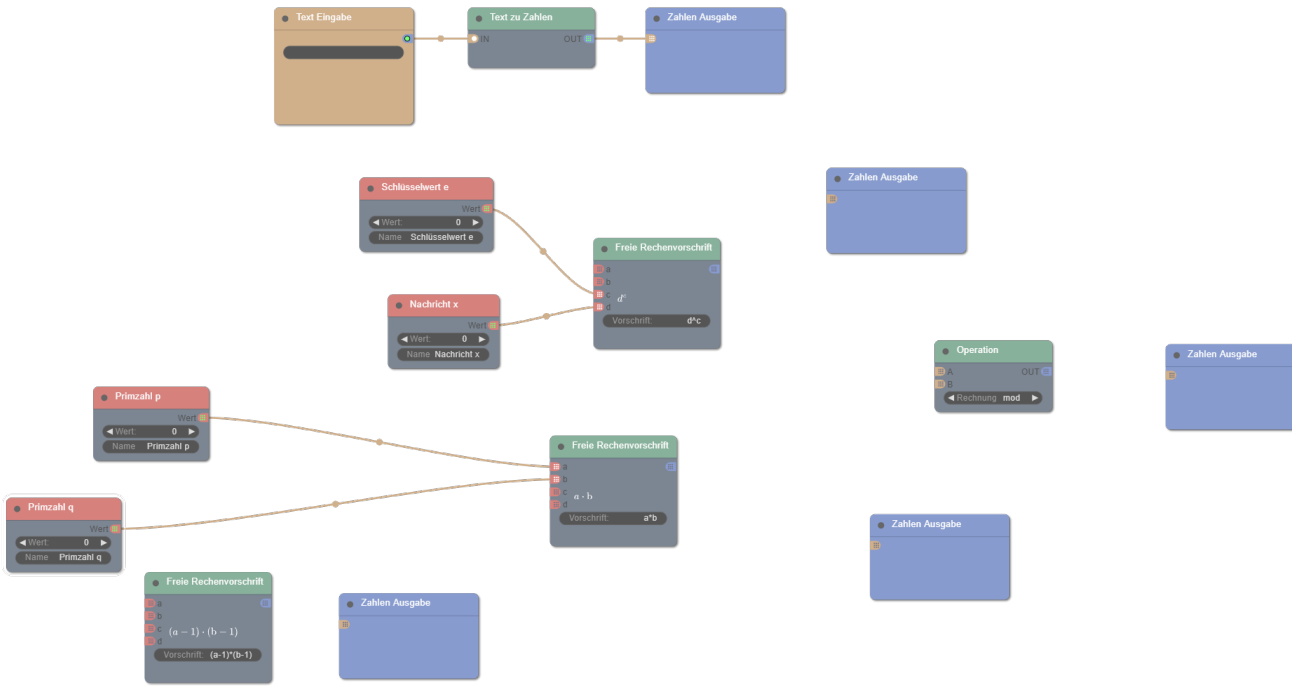
Step 2: The Key. Choose a number e that is relatively prime to ϕ ($ggT(e, \phi) = 1$). Only then is your lock secure!

Step 3: The message. Enter your message x as a number (it must be less than n). Perform the **exponentiation** (x to the power of e) and then the **modulo operation** (mod n).

Your task: Write down the values you chose and the resulting secret value c that the machine outputs.

Tip: Use the number outputs as a guide to check the intermediate steps for n and ϕ!

a)



The Euler phi function ϕ:

This mathematical function indicates how many numbers are relatively prime to your modulus n. This means they have no common divisors with n other than 1.

Normally, for large numbers, it is almost impossible to count these coprime numbers individually. However, we use the **mathematical trapdoor**: If we know both prime factors p and q that make up n, then ϕ can be calculated easily using the formula $\phi(n) = (p-1) \cdot (q-1)$.

This value is the secret knowledge of the machine. It must remain strictly confidential, since only with ϕ (and the public value e) can the corresponding private decryption key d be calculated.

9 | Task 6: Using the Trapdoor - Decrypting RSA

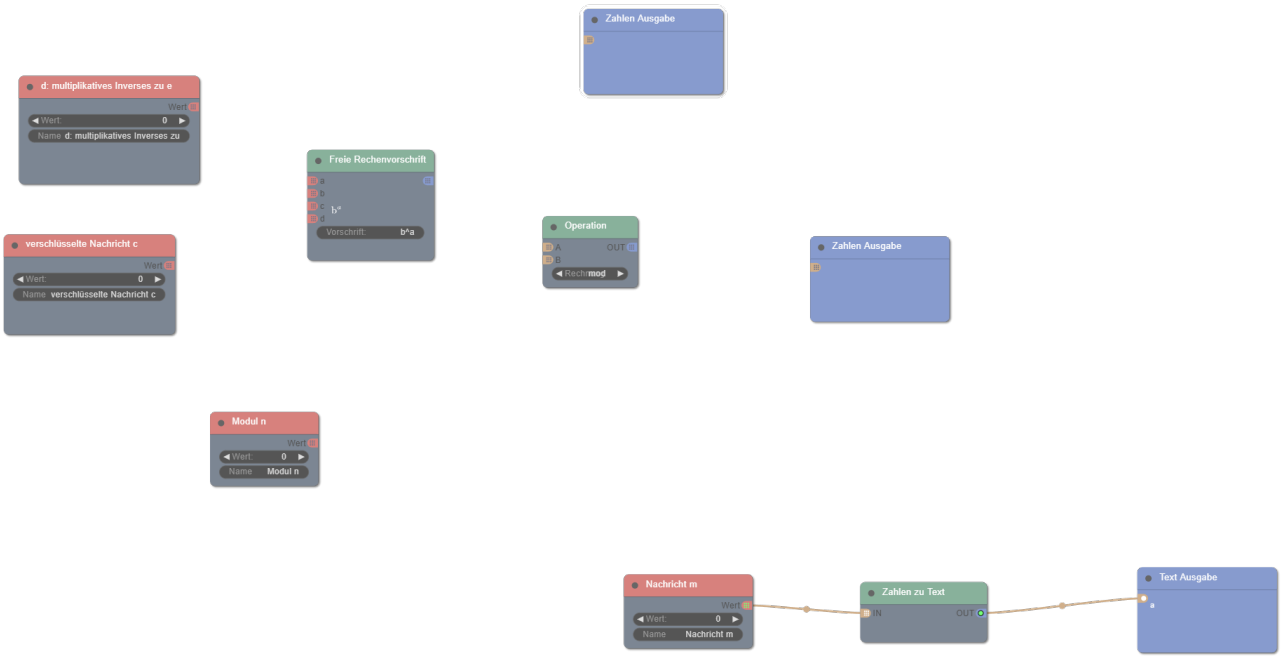
You have converted a message into the number c. But how does the recipient get the original text back from it? Simply working backward will not work here, since the modulo machine only displays remainders. You need the **trapdoor**: the private key d.

Step 1: Find the code. Use the online calculator described in the **workbook** to find the multiplicative inverse of e. Enter your e and your ϕ from Task 5 there.

Step 2: The decryption machine. Wire the decryption machine according to the formula $m = c^d \pmod{n}$. Enter your secret value c as the message and use your newly calculated d and the modulus n as parameters.

Check your work: What number does the machine display? Do you get your original message x back?

a)



The multiplicative inverse:

In the Modulo world, we are not looking for decimal numbers, but exclusively integers. The multiplicative inverse of your encryption value e is the number d, which mathematically exactly reverses the encryption step.

It works like the right key to a lock: When you multiply e and d, the result must yield a remainder of exactly 1 on the modulo-ϕ-clock. Mathematically, this means: $e \cdot d(\ (\mathrm{mod}\ \phi)) = 1$. This means that the product $e \cdot d$ is exactly one greater than a multiple of the secret number ϕ

Example: If $\phi = 20$ and $e = 3$, then your private key is $d = 7$. This is because $3 \cdot 7 = 21$, and $21\ (\mathrm{mod}\ 2)0 = 1$, since $21 : 20 = 1$ with a remainder of 1.