



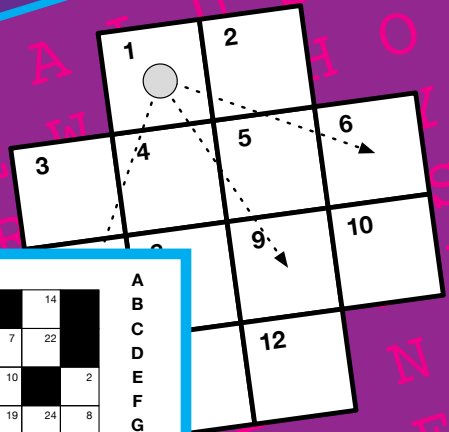
CS4FN PUZZLES

Computational Thinking Puzzles

	9 1001	6 01
3 0011	1 0001	0
12 1100	8 1000	0

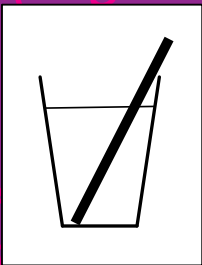
	22		11			11	24	11		8		14			
12	10	14	7	19	8	18		5	16	8	8	7	22		
	4		14			8		16		15	10		2		
11	19	10	3	14	20		6	11	13	11		19	24	8	
	1		11			10	18	9		18	25	8		17	
24	A ¹¹	R ¹⁸	L ¹⁵	8	3		9	16	22	19		22	18		
	18			3	8			20		3		19	11		
21	8	25	22			22	11	19	16	18	11	19	8		
				14			11				19	10	18		
	12	11	1	22			15	10	10	12		18	10	3	8
	25		11		22	8	23				14		14		
22	19	11	22	24		22	10	20	21		23		7	10	
	24		19	8	11	3		10					11		
26	10	15	8	3		11		3	11	26	7	8	19	22	
	7		9			7	14	17		10			8		

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



Card A

E



0 ^T	1 ^T	2	3 ^H	4 ^H
----------------	----------------	---	----------------	----------------



Queen Mary
University of London

Computational thinking puzzles

Computational thinking is a core set of skills that computer scientists develop as they learn to program. It isn't something you can only learn through programming though. Puzzles can be a great and fun way to develop the skills. This puzzle book involves a wide range of puzzles that involve aspects of computational thinking. Some are algorithmic puzzles where the aim is to come up with an algorithm that solves the puzzle. Many like Kakuro and Cut Block puzzles are logic puzzles, that are all about logical thinking. To be good at them, though, involves inventing your own rules and algorithms for solving them. Yet others, like code cracking grids involve computing concepts and algorithms in puzzle form.

If you enjoy these puzzles more can be found at www.cs4fn.org/puzzles/

Puzzles are most of all for fun, but it's always good to be learning useful skills too.

Solutions

Answers to a small selection of the puzzles can be found at the back. Full solutions are at: www.cs4fn.org/puzzles/solutions/

Word search

Words can appear in the grid horizontally, vertically or diagonally, running backwards or forwards. Different names can cross and overlap.

Find the following famous computer scientists' names in the word search grid. Their first and second names may be in different places. Spaces, accents and hyphens are not included.

ADA LOVELACE, ANITA BORG, BARBARA LISKOV, DANA ULERY, DOROTHY DENNING, FRAN ALLEN, GRACE HOPPER, JEANNETTE WING, KAREN SPARCK-JONES, MARISSA MAYER, MUFFY CALDER, URSULA MARTIN, WENDY HALL

ALAN TURING, CHRIS STRACHEY, EDGAR CODD, EDSGER DIJKSTRA, JOHN VON NEUMANN, MAURICE WILKES, MUHAMMAD AL-KHWARIZMI, NIKLAUS WIRTH, PHILIP EMEAGWALI, SERGEY BRIN, TIM BERNERS-LEE, TONY HOARE, VINT CERF

Which extra name appears in the grid: a forgotten genius of Bletchley Park?

```

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

```

A bit of computational thinking wisdom

You may be able to use your natural pattern matching abilities to spot some words. An algorithmic thinking approach may work better though. You need an organised way that is guaranteed to find the words. Take the first letter of a word and start at the top left hand corner, scanning along the rows in turn looking for it. When you find it check in all directions for the second letter. If its not there move on. If it is look for the third letter in the same direction, and so on. Can you improve this algorithm for solving word searches?

Cypher breaking grid

Crack the cypher by completing the crossword-style grid, and it will reveal one of the greatest movie messages of all time. In a cypher, individual symbols are replaced by new symbols. Here letters have been replaced by numbers. We've given you a few letters to start. Use the box at the top to record the key to the cypher. All letters of the alphabet appear in both the key and grid ... but which is which letter?

1	2	3	4	5	6	7	8	9	10	A ¹¹	12	13
14	L ¹⁵	16	17	R ¹⁸	19	20	21	22	23	24	25	26

	22		11		11	24	11		8		14			
12	10	14	7	19	8	18		5	16	8	8	7	22	
	4		14		8		16		15	10			2	
11	19	10	3	14	20		6	11	13	11		19	24	8
	1		11		10	18		9		18	25	8		17
24	A ¹¹	R ¹⁸	L ¹⁵	8	3		9	16	22	19		22		18
	18			3	8			20		3		19	11	
21	8	25	22			22	11	19	16	18	11	19	8	
			14		11					19	10	18		
	12	11	1	22		15	10	10	12		18	10	3	8
	25		11		22	8	23			14		14		
22	19	11	22	24		22	10	20	21		23		7	10
	24		19	8	11	3		10					11	
26	10	15	8	3		11		3	11	26	7	8	19	22
	7		9			7	14	17		10			8	

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

14	7	22	12	11	20	8	7	10	10	7	8	
20	11	7	24	8	11	18	25	10	16			
22	20	18	8	11	3							

A bit of computational thinking wisdom

The great Arab scholar Al-Kindi pointed out in the 9th century that you could use frequency analysis to crack simple cyphers. The most common symbols in the cypher are likely to be the most common letters in English and the least common ones the least common letters.

Create a representation of letters used: cross off the letters as you find them.

Cut blocks

There are two rules that must hold of a completed cut block puzzle.

- 1) Each area marked out by darker lines must contain the numbers from 1 up to the number of squares in the area. For example, the top most area in the first puzzle below consists of 5 squares so those squares must be filled with the numbers: 1, 2, 3, 4 and 5 with no repeated numbers. If the area has two squares, like the one bottom left below, then it must be filled with the numbers 1 and 2.
- 2) No number can be next to the same number in any direction, whether horizontally, vertically or diagonally. So in the grid below, the fact that there is a 4 on the side means there cannot be a 4 in any of the 5 squares surrounding it.

Here are two simple cut block puzzles, and then a harder one.

		2
4		
2		

	3	
	5	

		1	4			
			3			
		6				
2		3		5		4

A bit of computational thinking wisdom

As you fill in numbers, turn what you did into more general rules. Look out for similar situations, pattern matching against your rule, to give you quick ways of filling out the numbers without having to do all the same logical thinking from scratch each time.

4

Sherlock syllogisms

A syllogism, from the Greek words for conclusion and inference, is a logic puzzle where you draw a conclusion from particular kinds of purported facts you are given and those facts alone. Given the following facts, identify the letter that best completes the statement.

The game is afoot!

i)
All gems in the game are expensive in-game purchases.
All rubies in the game are gems.

Therefore we can conclude:

- a. some rubies in the game are expensive in-game purchases.
- b. all rubies in the game are expensive in-game purchases.
- c. some gems in the game are expensive in-game purchases.
- d. none of the above.

ii)
No robots are evil.
All mobile phones are robots.

Therefore we can conclude:

- a. All mobile phones are evil.
- b. All robots are mobile phones.
- c. Some mobile phones are evil.
- d. None of the above.

iii)
All bugs are poor computer software.
Some rounding errors are bugs.

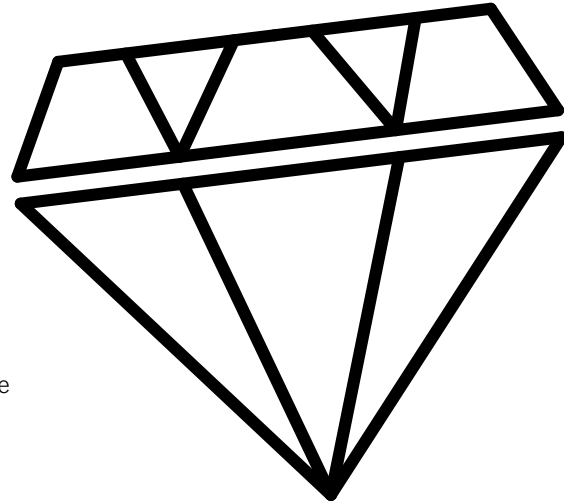
Therefore we can conclude:

- a. All rounding errors are poor computer software.
- b. Some rounding errors are poor computer software.
- c. Some rounding errors are false.
- d. None of the above.

iv)
All educational things are useful.
Some websites are not useful.

Therefore we can conclude:

- a. Some websites are not educational.
- b. All websites are educational.
- c. All educational things are not websites.
- d. None of the above.



A bit of computational thinking wisdom

Syllogisms are an important basis of logical thinking. Thinking clearly starts with understanding the 24 syllogisms. Much flawed reasoning involves people believing faulty syllogisms are true. Getting software right can involve this kind of clear thinking about the consequences of the code.

5

Word ladder

Convert the word LISP into the word JAVA in 5 steps or less. You must only change one letter of the word on each step. On every step you should have created a word in the english dictionary.

L I S P

- - - -

- - - -

- - - -

- - - -

J A V A

6

Bit ladder

Convert the binary word 000 into the binary word 100 in 7 steps or less. You must only change one bit of the word on each step. You may only use 1s and 0s.

0 0 0

- - -

- - -

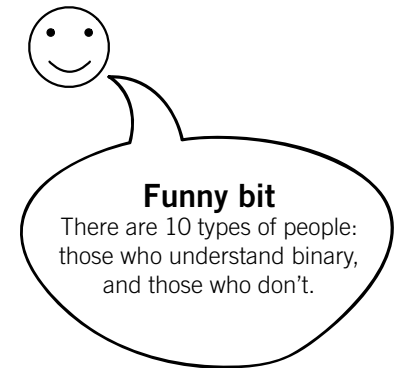
- - -

- - -

- - -

- - -

1 0 0



A bit of computational thinking wisdom

Bit ladders that end up back where they started, cycling through all of a sequence of 'words' by changing one symbol at a time, are also known as Gray codes. They are important as a different way to represent numbers in binary, giving a way of 'counting' through all the numbers by changing one bit at a time. This matters in devices that have mechanical switches like the early telegraph. The mechanism that flips bits won't change different ones at exactly the same time, so you could falsely register the wrong number mid-change if more than one needs to change at once. Gray codes avoid the problem.

Debugging spot the difference

The following is a correct fragment of code (in Python) to input two numbers.

```
# This program adds two numbers provided by the user
# Store input numbers
num1 = input('Enter first number: ')
num2 = input('Enter second number: ')

# Add two numbers
sum = float(num1) + float(num2)

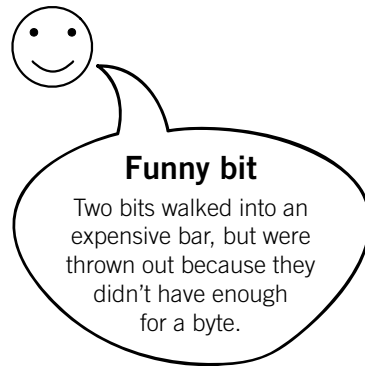
# Display the sum
print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
```

The following version has three mistakes. Can you spot the differences?

```
# This program adds two numbers provided by the user
# Store input numbers
num1 = input('Enter first number: ')
num2 = input('Enter second number: ')

# Add two numbers
sum = float(num) + float(num2)

# Display the sum
print('The sum of {0} and {1} is {2}.format(num1, num2, sum)
```



A bit of computational thinking wisdom
Debugging code involves being able to spot very subtle mistakes like these. The only trouble is you don't have the right answer to pattern match against: the pattern matching is against patterns in your head. The more you see and work with examples of correct code the stronger the patterns in your head will be. Attention to small details is an important part of computational thinking.

Kakuro

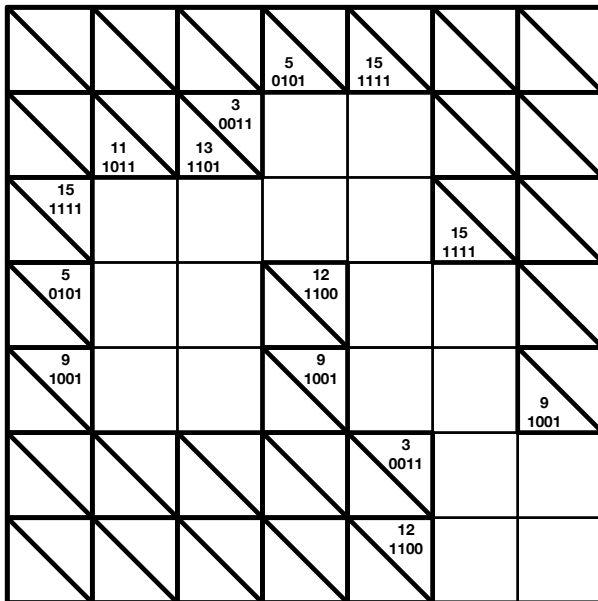
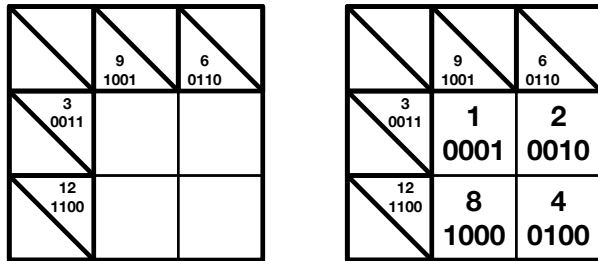
A Kakuro is a crossword-like grid where each square has to be filled in with a digit from 1-9. Each horizontal or vertical block of digits must add up to the number given to the left or above, respectively. All the digits in each such block must be different.

	4	3	15		7	6	10	15	
7				13					
6				10					
	3	6	7			5			4
10					10				
8				10			3		
	17	16		6				7	6
16			21						
30							3		
		16					5		

A bit of computational thinking wisdom
The same underlying logical thinking goes into Kakuro as in other logic based puzzles, just with arithmetic too.

Bakuro / Revelations

Bakuro or Revelations puzzles are binary versions of the well known Kakuro puzzle. The empty cells of the grid must be filled with the numbers 1,2,4 and 8 (i.e., only powers of 2). As with Kakuro, the numbers in each block in a column or row must add up to the number given in the clue above or to the left, respectively. No number can be used twice within any sum. The clues are given in both binary and decimal. The answers must also be written in both binary and decimal.



A bit of computational thinking wisdom
 Binary numbers are just a way of making up numbers by adding powers of 2 (1, 2, 4, 8, ...) together, just like decimal numbers involve adding powers of 10 (1, 10, 100, 1000) together.

Extreme logical thinking: eating at Quonk

A group of friends, two women (Alice and Babs), and two men (Zach and Yabu), like to pair up to go out on dates to cool restaurants. There are four combinations they date in (Alice-Zach, Alice-Yabu, Babs-Zach and Babs-Yabu). The favourite restaurant of one of the men and one of the women is a place called Quonk. However if those two eat together they always try new restaurants as do the other pair if together. Therefore when exactly one, and only one, of the particular man and woman in question is on a date they eat at Quonk. When Alice goes out with Zach they go to Quonk.

Which, if any, other pair from those below eat at Quonk:

- 1) Alice and Yabu,
- 2) Babs and Zach,
- 3) Babs and Yabu, or
- 4) none of the other pairs eat at Quonk?

Extreme logical thinking: a trip to market

A farmer is on her way to the local village with her sheepdog, Mist, who goes with her everywhere. To get to the village she has to cross a fast flowing river. An inventor living on the village side of the river has created a contraption to make it easier to get across. It consists of a rope and pulleys, with a seat hanging from the rope just big enough for one person. The locals have agreed to always leave the seat at the village side where the inventor lives so that it is easy for her to pack it away each evening: after all she is not charging anyone to use it. When she gets to the river the farmer pulls the seat across from the far side using the rope. She gets in, hugging Mist, then pulls herself across and continues into the village.

On one particular day she buys a new hen and a sack of corn. Returning home later in the day she arrives back at the ravine, and quickly realises she has a problem. She can only carry one thing across with her as she crosses using the seat. She will have to make several trips. The trouble is, if she leaves the hen and the corn alone on either side, the hen will eat the corn. Similarly if she leaves Mist and the hen together on one side the dog will worry the hen and may mean it stops laying eggs. Mist doesn't eat corn so it will come to no harm if left with him.

Write down the series of steps (the algorithm in computer science jargon) that she must take to get everything across so she can continue on her way.

A bit of computational thinking wisdom
 People make mistakes. That is why evaluation is important, very carefully checking that every last detail of a solution is right.

Compression codes

Repeatedly replace the characters in the following compressed messages by those from the corresponding codebook (the characters between the square brackets) to reveal a computing quotation and a poem credited to Augustus De Morgan, the great mathematician and tutor of 'first programmer' Ada Lovelace.

Message a

76FB5D
7CCCF0B9D

Codebook a

0 [...]
1 [,]
2 [aim]
3 [approach]
4 [development].]
5 [fast]
6 [fire,]
7 [Ready,]
8 [software]
9 [slow]
A [to]
B [(the]
C [21]
D [3A84]
E []
F [2E]

Message b

9JI
D8CA
37 JI
35AE

369IB
K40F8A
G6HK41A
341A 35 8.

Codebook b

0 [fleas]
1 [still]
2 [have]
3 [and]
4 [greater]
5 [so]
6 [those]
7 [lesser]
8 [on]
9 [great]
A [,]
B [themselvesA in turnA]
C [their backs to bite 'em]
D [up]
E [ad infinitum.]
F [to go]
G [while]
H [again]
I [0A]
J [0 K7]
K [2]

A bit of computational thinking wisdom

When files are compressed (e.g. zip files) they use similar codes to this one. Each common word or phrase is replaced by a shorter sequence of symbols. A long file can be made much shorter if it has lots of similar sequences, just as these message has been shortened.

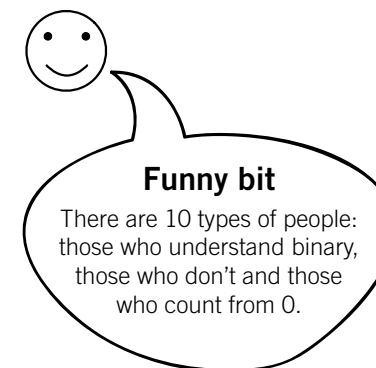
Debugging spot the difference

The following is a correct fragment of code (in Java) for sorting an array.

```
public static void Sort(int[]a, int n)
{
    for (int p=1; p <= n-1; p++)
    {
        for (int i=0; i < n-p; i++)
        {
            if (a[i] > a[i+1])
            {
                swap(a, i, i+1);
            }
        }
    }
}
```

The following version has mistakes. Can you spot 18 differences (i.e. bugs)?

```
public static Sort(int{}a; int m)
{
    for (int p=0, p < n-1, p++)
    {
        For (int i; i < n-q; i++)
        {
            if (a[i] < a[i+1];
            {
                swap(a, i, i+1)
            }
        }
    }
}
```



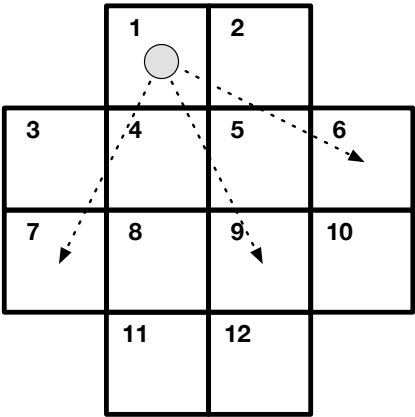
A bit of computational thinking wisdom

Actively checking for particular patterns where things go wrong, like "is the inequality right?" is an important debugging technique.

16

The Knight's tour

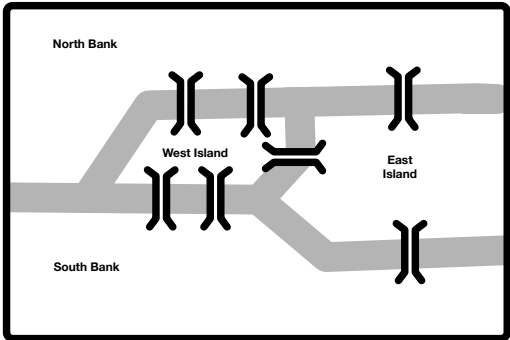
A single chess Knight is able to move on the small cross shaped board below. A Knight can move two spaces in one direction and then move one square at right angles, or vice versa, as shown. It jumps to the new square without visiting any in between, and must always land on a square on the board. Find a sequence of moves that starts from Square 1, visits every square exactly once by making such knight's moves and finishes where it started.



17

The bridges of Königsberg

Below is a map of the city of Königsberg, showing the river that runs through the middle, its two islands and the seven bridges that cross the river. The tourist information centre would like to publish a route that visits each part of the city (both banks and both islands) and that crosses each bridge once (and no more). It should start and end in the same place. You have been asked to advise them. Either provide a route or if you can't, explain why it is not possible.



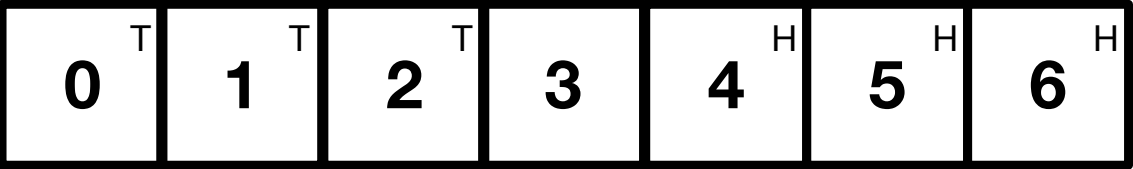
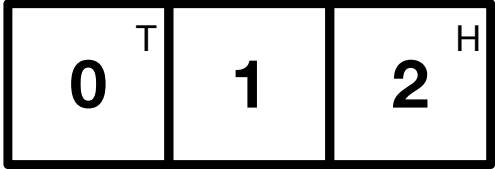
18

The swap puzzle

Place small coins heads up on the squares marked H and tails up on the squares marked T. Swap the positions of the Heads for the Tails in as few moves as possible.

- There are two ways to move a piece:
1. Move left or right to an adjacent empty square
 2. Jump over a single adjacent piece into an empty space.

There are three increasingly larger boards that get harder. Complete the first in 3 moves, the second in 8 moves and the third in 15 moves.



A bit of computational thinking wisdom
Harder problems can be made easier by tackling simpler version first, then generalising the solution.

Word search

Words can appear horizontally, vertically or diagonally, running backwards or forwards. Different names can cross and overlap.

Find the following computing words in the word search grid below.

BAR ENGINE KEY NETWORKING SAVE SCAN SCANNER SCREEN SCREENSHOT SCRIPT SCROLL
SEARCH SECURITY SERVER SHAREWARE SHELL SHIFT SNAPSHOT SOCIAL SOFTWARE SPAM
SPAMMER SPREADSHEET SPYWARE STATUS STORAGE SUPERCOMPUTER SURF SYNTAX

```

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

```

A bit of computational thinking wisdom
Perhaps you can improve your algorithm for finding words so that it is more efficient? For example, some letters may appear less often in the grid than others, so are quicker to search for. Perhaps searching for the first letter of a word is not the best idea.

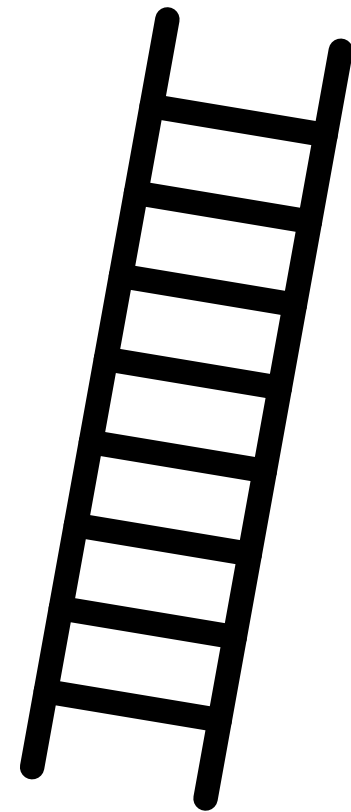
Word ladder

Convert the word ALICE into the word GENES in 13 steps or less. You must only change one letter of the word on each step. On every step you should have created an English word that is in the dictionary. ALICE in Wonderland was written by Lewis Carroll, who also invented this kind of puzzle. The binary version of the word ladder is called Gray code - a way of encoding numbers. Our bodies encode information using the 4 symbols of DNA rather than the 2 of binary. Its all in our GENES.

```

A L I C E
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
G E N E S

```



Lovelace's life ladder

Ada Lovelace is famous as the 'first programmer' and also predicted computers would one day compose music. Born on the 10th December 1815, she died at the age of 36. Use the clues to fill in the missing numbers that completes the move from life to death, changing only one digit at a time?

Date	Clue
1 8 1 5	Ada Lovelace was born
----	Tchaikovsky's famous Overture
----	Ada Lovelace died

Cryptogram

The following message about Ada Lovelace as a child is in code. Crack the cypher and decode the message. Each letter of the alphabet is given a distinct number. Not all letters appear in the message.

Message

6	25	24	20	8	9	5	21	7	2	7	2	23	24	7	3	6
8	19	10	24	12	5	21	7	10	16	24	6	8	19	16	20	
7	5	2	7	22	22	23	8	7	17	25	24	2	6	25	24	
22	23	8	10	16	24	3	8	19	25	24	23	24	7	23	6	25
10	8	9	5	2	13	6	7	6	24	18	12	6	25			
6	20	22	12	17	7	16	21	9	13	6	8	7	5	2	7	
13	17	12	24	5	6	12	19	12	17	24	20	24				

Key

1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26

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 bit of computational thinking wisdom

Count the number of times each number appears. The Allies won World War II through computational thinking - they cracked Hitler's cyphers. That meant the Allies' generals often knew what the Germans would do. One way was by looking for 'cribs': words or phrases that would appear in the messages and looking for a corresponding pattern in the cyphertext. 'Heil Hitler' was often used as a crib as it appeared all the time in Nazi messages. What might be a crib to use in a message about Ada Lovelace?

Extreme logical thinking: card sorting, well sort of

a) One day you are shown the four cards below. You are told that each card has a number on one side and a letter on the other. You are also told that every card that has a vowel on one side has an even number on its opposite side: a nice and simple fact, but is it true? Which card or cards must you turn over to prove whether or not the vowel/even rule is true? Explain why.

Card A	Card B	Card C	Card D
E	G	2	3

b) On another day, you are shown a different set of four cards as below. This time each card has the details of a person in a shop on it. On one side is their age and on the other is what they are buying. If a person is buying fireworks then they must be over 18. Which cards should you turn over to check everyone is shopping legally.

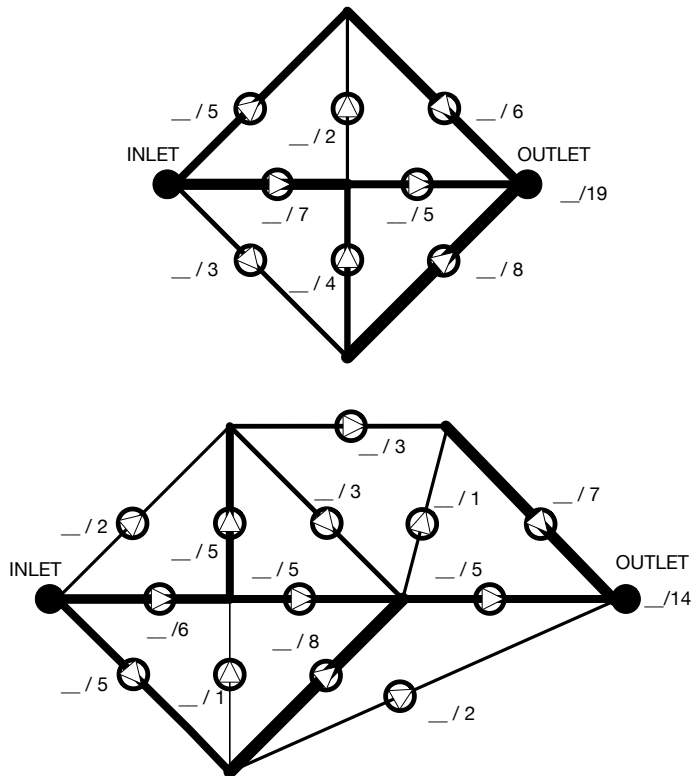
Card A	Card B	Card C	Card D
Age 25	Buying Rockets	Age 16	Buying Milk

Girls and boys?

We know that the Professor has exactly two children. The older of the two is a girl. What is the probability that both of the Professor's children are girls?

Pinky's pipe pickle

Pinky the plumber is often called to houses where the water comes out of the shower in only a trickle. The network of pipes have taps to increase and decrease the flow of water in each pipe. The taps are marked as circles with triangles showing the direction of the flow. To turn the shower in to a working power shower, you must set each of the taps in a way that means the maximum amount of water flows from the inlet to the outlet of the whole circuit. The pipes are different sizes so have different maximum flows. The numbers against each pipe show the amount of water currently flowing down it and the maximum it takes. So, for example, 2/6 means that pipe takes 6 litres a second but at the moment only 2 litres per second are going through it. When you arrive, all pipes are set to 0. You cannot change the maximum flow on any pipe, or the direction of the flow. However, by turning the taps you can change the amount flowing through each pipe, as long as you don't go above its maximum. You must never leave a junction with more water flowing in to it than is flowing out. The pressure will build up and you could then end up needing a bucket, mop and wellies. What is the maximum flow at the outlet for the following networks?



A bit of computational thinking wisdom
 Network engineers of all kinds face similar problems. For example, in computer networks the direction packets of data are routed must be adjusted to avoid congestion. Complex algorithms constantly adjust the traffic in the network to try and ensure everyone's data arrives as quickly as possible. Find an algorithm to solve maxflow puzzles for computer networks and you have one for plumbers too. That's generalisation.

Codeoku

Codeoku are exactly like Sudoku except they use symbols instead of numbers. You must place the 9 symbols in the grid so that every row and every column contains all the symbols. Also every marked 3x3 square should contain one of each symbol. Complete the grid using only the following symbols found in computer programs: [;] > % \$ & # @

%	}	@	\$	&				
;	>	&		}	[\$		#
\$		[%	@	&	
#		%		;	>	}	\$	
	&	\$	%				@	[
			&				>	%
		#	}	[%	;	>
		>			;	[}	\$
	;	}	>	%	\$	&	#	@

A bit of computational thinking wisdom
 Logically this puzzle is exactly the same as one with numbers in place of the symbols, so logically it is no harder. However, the way the information is represented has changed (from numbers we are very familiar with to symbols that are less familiar). The symbol representation actually makes it harder as we have lost information that a human may use. In particular, these symbols have no natural order, so its harder to keep track of which symbols you have checked. For example, with a number version, one strategy is to work through the numbers in turn looking for possible moves. With no natural order that is harder. A well chosen representation of information can make a task easier. A poorly chosen one can make it harder.

Navigating the numbers

Using the clues given and the program bank below, work out the number answers that are hidden in the number grid. Then find them in grid! Take care because the answer numbers in the grid can run horizontally, vertically or diagonally, forwards or backwards!

Program Pr1

```
Start Program
  Get length, l,
  Get width, w
  Compute area = l * w
  Print area
End Program
```

Program Pr2

```
GET number A
GET number B
Sum = A + B
Print Sum
```

Program Pr3

```
X=591
FOR J = 1 TO 10
  Print J
NEXT J
X=J*X+2
Print X
END
```

Program Pr4

```
IF W > N
  THEN Print X
  ELSE Print Y
ENDIF
```

Program Pr5

```
IF W > N
  THEN Print X
  ELSE IF X > Z
    THEN Print Y + Z
  ENDIF
ENDIF
```

Program Pr6

```
i = 10;
while ( i > 0 ){
  print("Hello");
  i = i - 1;
  if( i == 6 )
    break;
}
```



Funny bit

There are 10 types of people: those who understand binary, those who don't and those who didn't expect this joke to be in ternary.

Question bank

- Apple releases the Apple Watch in this year
- Binary 10001111 in octal (base 8)
- Binary 10010010010111 in decimal
- Ada Lovelace was born in this year
- Bytes in a kibibyte
- Charles Babbage was born in this year
- Clive Sinclair introduces the first pocket calculator
- D80 Hex in decimal
- Ada Lovelace died in this year
- Decimal output of Pr1 with l=9C hexadecimal and w= 21 Octal.
- Decimal output of Pr2 with A=1111101001 binary and B=10 binary
- Number of bits in a byte multiplied by gibibyte in a tebibyte
- Number with 2, 3 and 23 as prime factors
- Output from Pr4 where N=101, X=2567 Y=3421 and W=203
- Output from Pr4 where N=101, X=89353 Y=34835 and W=10**2
- Output from Pr5 with N=100, X=2345 Y=1243, Z=2344 and W=99
- Last output from running Pr3
- Output of Pr1 with l=63 and w=51.
- Output of Pr2 with A=6 and B=15^2
- Sinclair ZX81 Computer Launched
- X*Y*Z, where
 - X=(The number of times Pr6 prints 'Hello')
 - Y=(the value of variable J at the end of running Pr3)
 - Z=(output from Pr2 with A=223 B=-100)
- Alan Turing died in this year

Number search grid

1	3	4	5	6	1	9	8	1	2
9	1	7	1	0	2	4	9	5	2
7	1	8	0	3	1	8	6	9	1
2	3	2	2	2	2	7	1	3	9
1	3	8	6	3	4	8	3	5	5
1	1	3	2	7	6	1	2	8	4
5	9	1	2	0	2	9	4	7	3
1	3	1	7	7	6	3	9	9	8
1	8	1	5	9	5	3	0	0	1
0	1	8	5	2	1	2	6	5	2

Bakuro / Revelations

Fill the empty cells of the grid with the numbers 1,2,4 and 8 (i.e., only powers of 2). The numbers in each block in a column or row must add up to the number given in the clue above or to the left, respectively. No number can be used twice within any sum. The clues are given in both binary and decimal. The answers must also be written in both binary and decimal.

	9 1001	6 0110		7 0111	1 0001	15 1111	
5 0101			7 0111				3 0011
15 1111						3 0011	
		6 0110			10 1010		
	12 1100			3 0011			
	12 1100			3 0011			
12 1100			10 1010			10 1010	
9 1001			7 0111				
				9 1001			

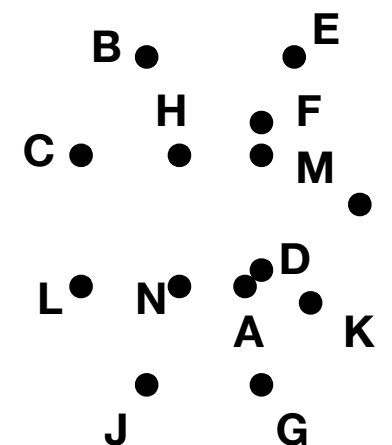
A bit of computational thinking wisdom

Binary numbers are based on powers of 2. You may have noticed that, because of this, all the binary versions of the numbers you are writing in the grid have only a single 1 in them. The position of the 1s in the numbers being added correspond to the number in the clue they add to. So, for example, to make 13 (1101) you need to use the numbers 8 (1000) + 4 (0100) + 1 (0001). The binary tells you which numbers you need to use!

History timeline dot to dot

Place the following events in computing history in to the order they happened. Then join the dots in that order to find a sign first used by medieval monks, but that gained a new computing life in the 1970s.

- Karen Spärck Jones invents IDF weighting, the algorithm behind most search engines, making it practical to search the web.
- Francis Bacon invents the Bacon cipher, a way of encoding letters in binary for use in secret writing.
- Mary, Queen of Scots is executed as a result of Walsingham decoding her secret messages plotting to kill Elizabeth I using frequency analysis.
- Ada Lovelace's notes on Charles Babbage's proposed analytical engine lead to her being hailed as the first programmer.
- Gottfried Leibniz discovers the modern binary number system.
- Jeanette Wing popularises the idea of Computational Thinking as the core, generally useful, skill set of the computer scientist.
- The idea of steganography, the practice of hiding messages, is recorded for the first time in the book 'Histories' by Herodotus.
- Grace Hopper invents the first programming language compiler, making code far easier to write.
- Jacques de Vaucanson builds The Flute Player, the first biomechanical automaton. It is a life-size figure that can play 12 different songs.
- Julius Caesar uses what becomes known as the Caesar cipher to encrypt his messages.
- The Jacquard loom is first demonstrated. It revolutionises weaving and its punch card system that controls the pattern is a foundation for the idea of stored computer programs and data storage.
- Al-Khwarizmi writes his book 'On the Calculation with Hindu Numerals': the translation of his name gives us the word algorithm.
- Alan Turing invents the Universal Turing Machine, a mathematical precursor to the stored program computer.
- ALGOL, the first block structured programming language is invented influencing virtually all languages that follow.



Compression codes

Repeatedly replace the characters in the following compressed message by those from the codebook (the characters between the square brackets) to reveal a computing joke.

Message

C
C
46A7B5DF
7GHE9.

Codebook

0 [little]
1 [code]
2 [in the]
3 [ninety-nine]
4 [one]
5 [compile]
6 [bug]
7 [and]
8 [s hiding]
9 [06821]
A [is fixed ...]
B [we]
C [39,]
D [it all]
E [hundred]
F [again,]
G [there are]
H [a]
I [fixed]



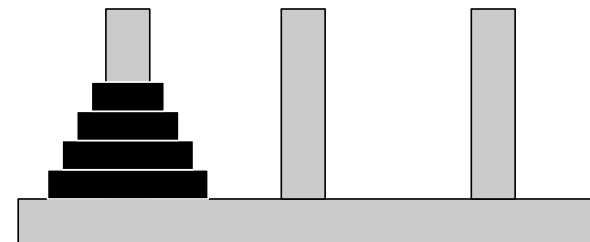
Troubling Communication

You are a Doctor in a hospital, where a new patient needs help. They have had a stroke that has left them with 'Locked-in syndrome'. This means they are almost totally paralysed, unable to move, or even speak, though they are fully conscious and still intelligent. They can see and hear everything around them, but just can't communicate back. The one movement that they have is that they can blink one eyelid, though that is quite an effort to do. You cannot cure them, but perhaps you can help them to communicate.

- a) Devise a way, that you can explain to them, that will allow them to talk to you. As all they can do is blink, they will need to use blinking to communicate letters, and so words, to you. What do you need to do, and what do they need to do to make it work?
- b) How might you adapt the method you have suggested if blinking is easier for them to make it quicker for them to communicate.

Program search

The following 7 line Python program solves the ancient Towers of Hanoi puzzle. To solve the puzzle you must come up with a series of steps to move a stack of different sized rings from one pole to another. They must be moved one at a time, using only a single intermediate pole, so that no ring is ever placed on top of a smaller ring.



```
def thanoi (pieces, movefrom, moveto, other):
    if pieces == 1:
        print ("move ring from", movefrom, "-", moveto)
    else:
        thanoi (pieces-1, movefrom, other, moveto)
        thanoi (1, movefrom, moveto, other)
        thanoi (pieces-1, other, moveto, movefrom)
```

We have hidden the program in the following word search grid vertically and horizontally. Punctuation is included at the start or end of words. Spaces separate words. Repeated words appear multiple times. Can you find the whole program.

i f o t h e r (: 1 = =) v e f m
p r = t n i r p i n = : o) h r o
i o t h a n o i " - " , t m a o v
e n h a (p i e c e s , e o n - e
c " a n (" o c r i n) v r , l t
e , n o m o t e l s e : o f m m o
s v o i m d m s g n i r m e o o)
(p i e c e s - 1 , o v o v r v o
d e m t h f p l , " t p v o f e :
e v o m " (1 , d t h a e m e f)
m o v e f r o m , f e f t s v r r
, r e h t o o t h e r) o = o o e
m o t h a n o i r i , t , = m m h
o m o v e f r o m , f r o m " , t
m r , m o v e t o , t h a n o n o

Solutions to a selection of the puzzles

We have included answers to a few simple puzzles as tasters. The answer booklet containing solutions to all the puzzles in this puzzle book can be found at:

www.cs4fn.org/puzzles/solutions/

Puzzle 3: the simple cut block puzzles

1	3	2
4	5	1
1	3	2
2	4	1

1	4	2
2	3	1
1	4	2
3	5	1

Puzzle 4: Sherlock syllogisms

- (i)
- b: All rubies in the game are expensive in-game purchases. c and d are also true but give less complete information

Puzzle 5: word ladder

Here is one solution:

L I S P
L I M P
L A M P
L A M A
L A V A
J A V A

Puzzle 6: bit ladder

Here is one solution:

0 0 0
0 0 1
0 1 1
0 1 0
1 1 0
1 1 1
1 0 1
1 0 0

Puzzle 12: compression code

Ready, fire, aim (the fast approach to software development).

Ready, aim, aim, aim, aim ... (the slow approach to software development).

Puzzle 18: the swap puzzle

The level 1 puzzle can be solved in 3 moves as follows:

Step 1: **Square 1** GETS THE PIECE FROM **Square 0**

Step 2: **Square 0** GETS THE PIECE FROM **Square 2**

Step 3: **Square 2** GETS THE PIECE FROM **Square 1**

The level 2 puzzle can be solved in 8 moves as follows:

Step 1: **Square 2** GETS THE PIECE FROM **Square 1**

Step 2: **Square 1** GETS THE PIECE FROM **Square 3**

Step 3: **Square 3** GETS THE PIECE FROM **Square 4**

Step 4: **Square 4** GETS THE PIECE FROM **Square 2**

Step 5: **Square 2** GETS THE PIECE FROM **Square 0**

Step 6: **Square 0** GETS THE PIECE FROM **Square 1**

Step 7: **Square 1** GETS THE PIECE FROM **Square 3**

Step 8: **Square 3** GETS THE PIECE FROM **Square 2**

Puzzle 21: Lovelace's life ladder

1815

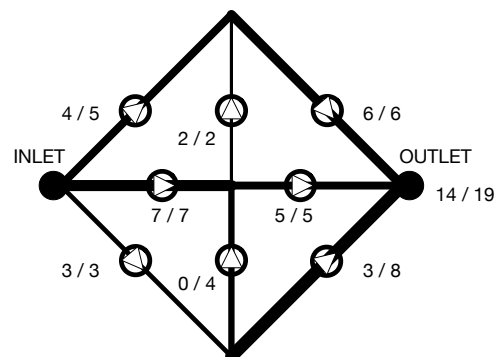
1812

1852

Puzzle 28: Pinky's pipe pickle

The maximum flow is 14.

Here is one solution that gives that maximum flow.



Teachers

pdf copies of many of these individual puzzles for photocopying for class use can be downloaded from www.teachinglondoncomputing.org/puzzles/ Other complementary resources can be downloaded too including in-depth computational thinking booklets and activity sheets linked to these puzzles, and linked programs to run.

This puzzle book was written by Paul Curzon and Peter McOwan of Queen Mary University of London. Spring 2016. Hamet Soyel helped test the puzzles. This book was made possible due to support from Google's CS4HS programme with additional support from the Department for Education and Mayor of London.



Attribution NonCommercial ShareAlike - "CC BY-NC-SA"

This license lets others remix, tweak, and build upon a work non-commercially, as long as they credit the original author and license their new creations under the identical terms. Others can download and redistribute this work just like the by-nc-nd license, but they can also translate, make remixes, and produce new stories based on the work. All new work based on the original will carry the same license, so any derivatives will also be non-commercial in nature.



Queen Mary
University of London

For more computational thinking
puzzle fun go to
www.cs4fn.org/puzzles

Supported by the Google CS4HS programme



Department
for Education

SUPPORTED BY

MAYOR OF LONDON

COMPUTING AT SCHOOL
EDUCATE • ENGAGE • ENCOURAGE

