Fundamentals of Computer Science

Logic Puzzles Workbook

**These puzzles aim to develop your Computational Thinking skills.
Enjoy solving them!**

Reproduced from material prepared by Paul Curzon, Professor of Computer Science at Queen Mary University of London.

Introduction

These puzzles are aimed at developing Computational Thinking (CT) skills, that are highly regarded in problem-solving. Every module that you do in your BSc/MEng Computer Science, BSc. Computing or BSc. Business Information Technology uses CT in solving problems.

However the Fundamentals of Computer Science module makes it clear to you what it actually means. Aim to practise solving the following puzzles, and along the way develop your CT skills. Enjoy solving them, at the meantime bearing in mind that you will find similar questions in the Week 10: Phase Test assessment.

All the best!

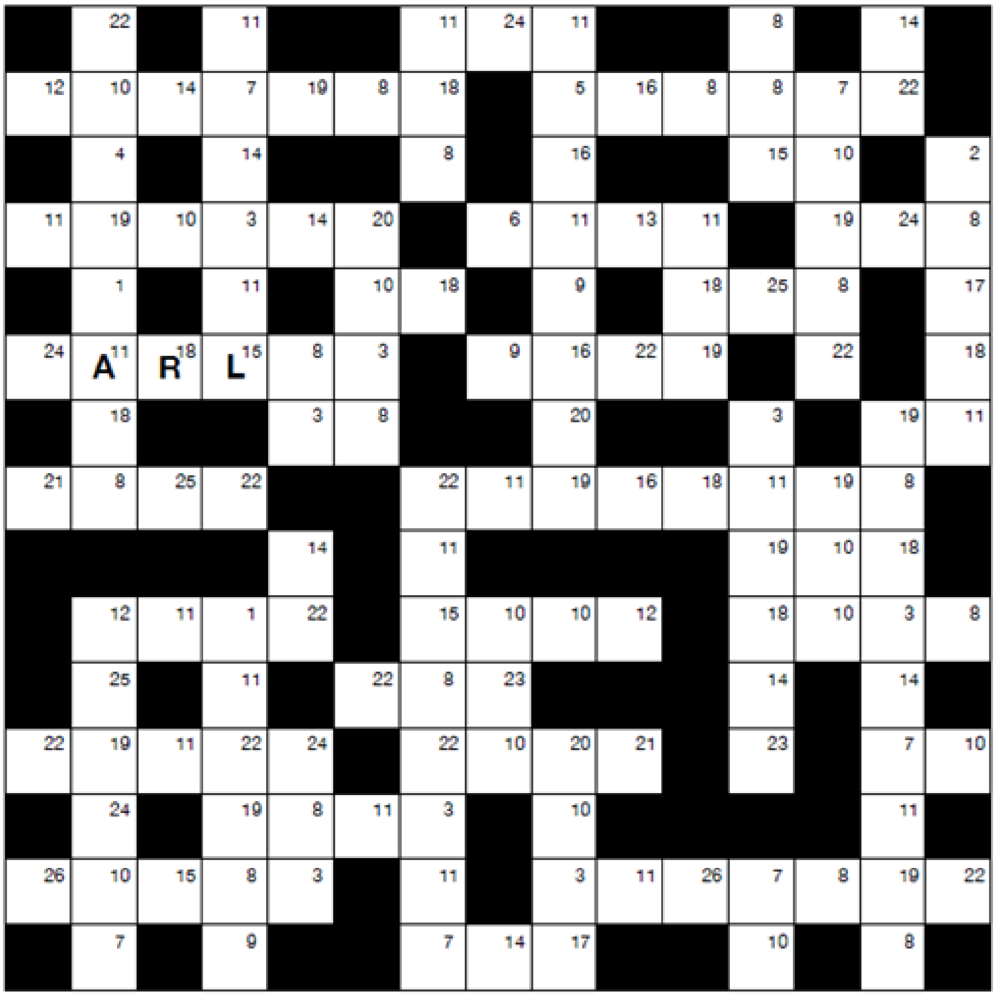
Meg, Lesley and Kiran

The FCS teaching team

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 Here, letters have been replaced by numbers.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 have given you four 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 M	4	5	6	7	8	9	10	11 A	12	13
14	15 L	16	17	18 R	19	20	21	22	23	24	25	26



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.

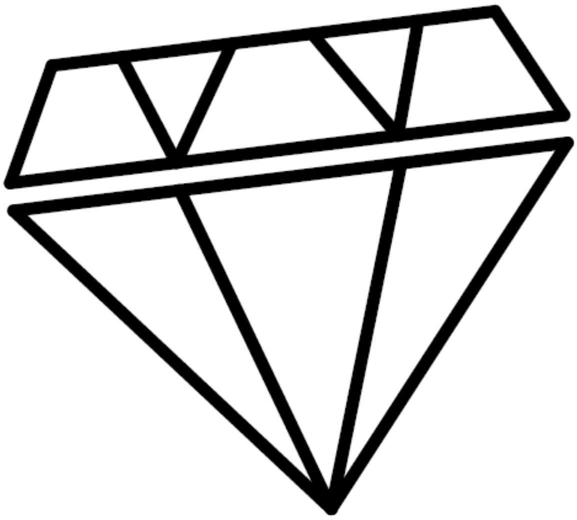
Sherlock syllogisms

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.

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

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?

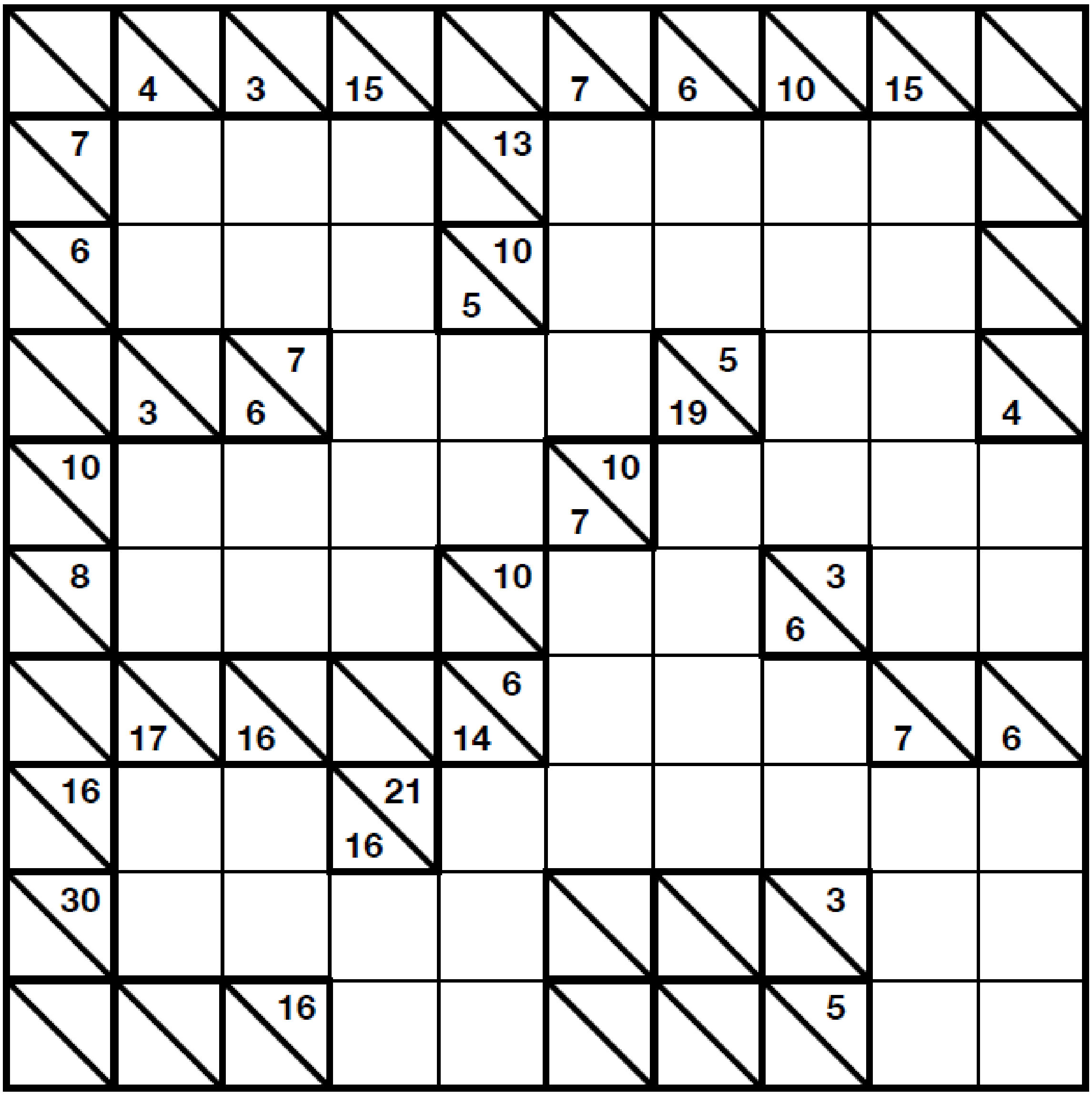
```
# This program adds two numbers provided by the user
# Store input numbers
num1 = input('Enter first number: ')
num2 = input('Enter second number: ')

# 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.



A bit of computational thinking wisdom

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

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)?

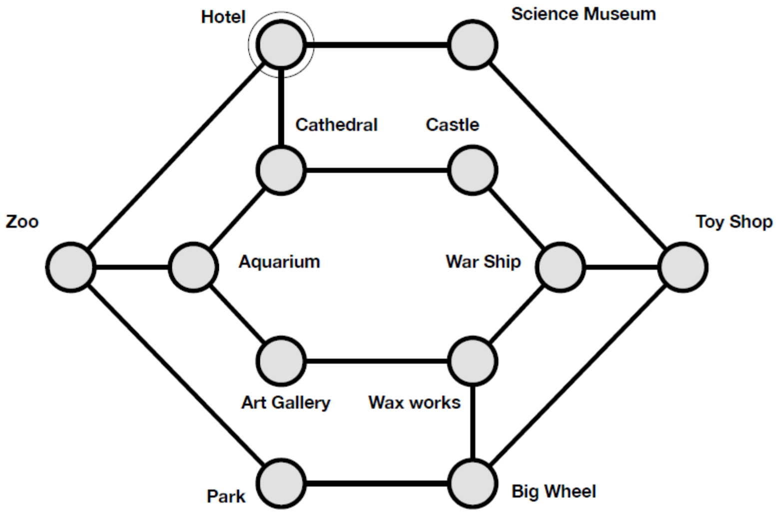
```
public static Sort(int{}a; int m)
{
	for (int p=0, p < n-1, p++)
	{
		For (int i; i < n-q; i++)
		{
		if (a[i] < a[i+1];
		{
			swap(a, i, i+l)
		}
	}
}
```

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.

The tour guide

You are a hotel tour guide. Tourists staying in your hotel expect to be taken on a tour visiting all the city's attractions. You have been given an underground map that shows all the locations of the attractions and how you can get from one to another using the underground network. You must work out a route that starts from the hotel and takes your tour group to every tourist site. The tourists will be unhappy if they pass through the same place twice. They also want to end up back at their hotel that evening.

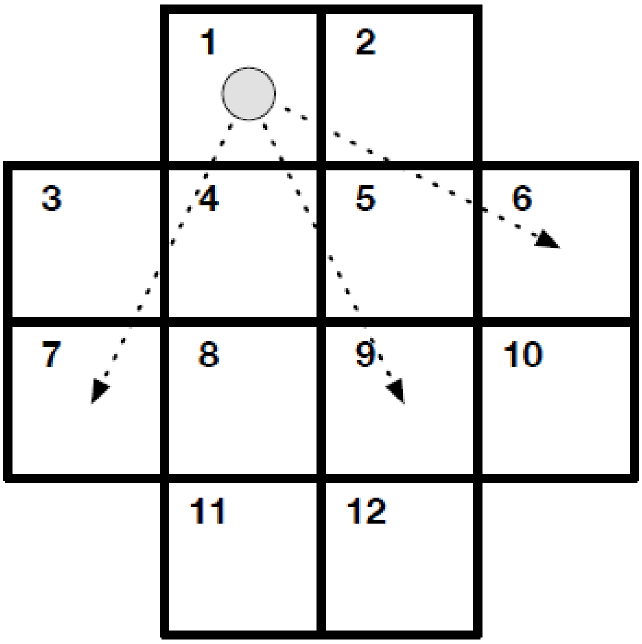


A bit of computational thinking wisdom

A good representation for problems involving 'places' and ways to get between them is to draw circles (or nodes) for the places and lines (called edges) between them to show the links. This is called a graph by computer scientists.

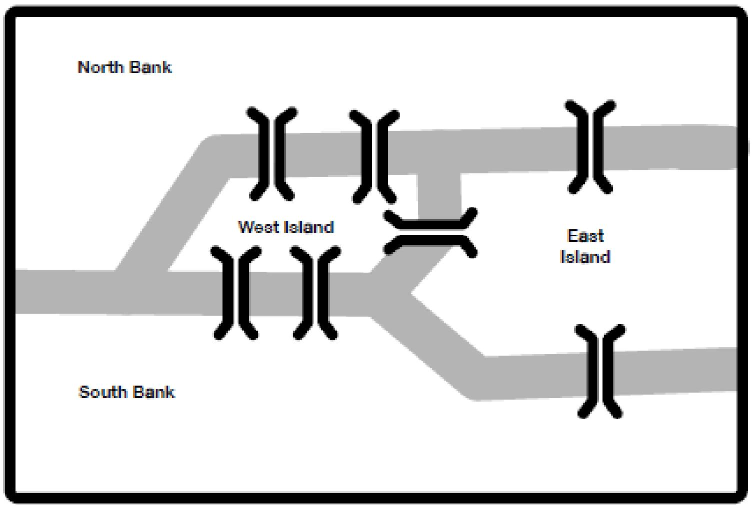
The Knight's tour

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.



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.



A bit of computational thinking wisdom

Harder problems can be made easier by tackling simpler version first, then generalising the solution.

Bit ladder

Convert the binary word 0000 into the binary word 1000 in exactly 15 steps. You must only change one bit of the word on each step. You may only use 1s and 0s. Your answer must involve every combination of 1s and 0s.

0 0 0 0

__ __ __ __

__ __ __ __

__ __ __ __

__ __ __ __

__ __ __ __

__ __ __ __

__ __ __ __

__ __ __ __

__ __ __ __

__ __ __ __

__ __ __ __

__ __ __ __

__ __ __ __

__ __ __ __

1 0 0 0

A bit of computational thinking wisdom
Once they've solved a problem and invented an algorithm to do it, computational thinkers generalise it so thev can use the solution in different ways.

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?

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?

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. **Use Google to find the dates!**

A) Karen Spärck Jones invents IDF weighting, the algorithm behind most search engines, making it practical to search the web.

B) Francis Bacon invents the Bacon cipher, a way of encoding letters in binary for use in secret writing.

C) Mary, Queen of Scots is executed as a result of Walsingham decoding her secret messages plotting to kill Elizabeth I using frequency analysis.

D) Ada Lovelace's notes on Charles Babbage's proposed analytical engine lead to her being hailed as the first programmer.

E) Gottfried Leibniz discovers the modern binary number system.

F) Jeanette Wing popularises the idea of Computational Thinking as the core, generally useful, skill set of the computer scientist.

G) The idea of steganography, the practice of hiding messages, is recorded for the first time in the book 'Histories' by Herodotus.

H) Grace Hopper invents the first programming language compiler, making code far easier to write.

I) Jacques de Vaucanson builds The Flute Player, the first biomechanical automaton. It is a life-size figure that can play 12 different songs.

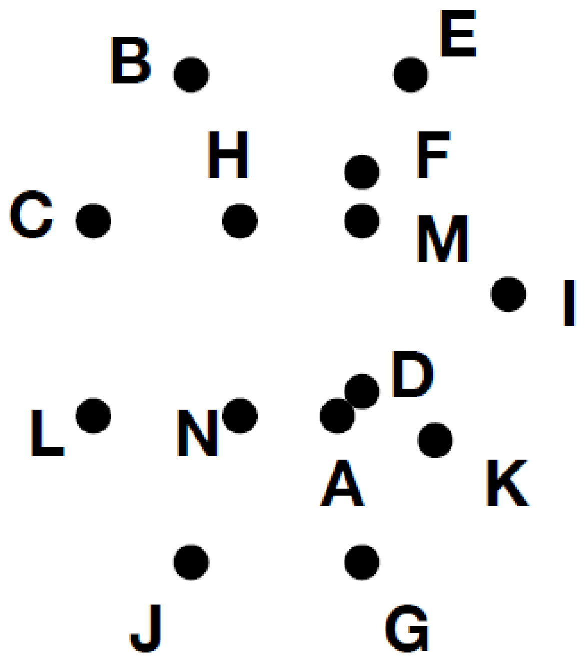
J) Julius Caesar uses what becomes known as the Caesar cipher to encrypt his messages.

K) 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.

L) Al-Khwarizmi writes his book 'On the Calculation with Hindu Numerals': the translation of his name gives us the word algorithm.

M) Alan Turing invents the Universal Turing Machine, a mathematical precursor to the stored program computer.

N) ALGOL, the first block structured programming language is invented influencing virtually all languages that follow.



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!

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]