Computational Thinking Puzzles Answer Booklet

www.cs4fn.org/puzzles

This answer booklet and its linked puzzle book were written by Paul Curzon and Peter McOwan of Queen Mary University of London. It is possible due to support from Google's CS4HS programme with additional support from the Department for Education, Mayor of London and EPSRC through the CHI+MED research project.

1. Word Search

										1	1	1	1	1	1	1	1	1	1
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	N	**K**	-	-	-	S	-	-	-	-	**T**	-	-	A	L	U	S	R	**U**
1	A	A	-	F	R	E	**C**	-	-	N	I	-	-	-	-	N	H	O	**J**
2	M	R	-	-	-	K	-	-	I	-	M	-	-	**N**	-	**F**	**B**	-	-
3	H	E	**A**	**D**	A	**L**	O	**V**	E	L	A	C	E	I	-	R	R	**S**	**P**
4	C	N	N	I	**W**	I	R	T	H	-	-	-	-	K	R	A	I	E	H
5	L	**S**	I	J	-	**W**	E	N	D	Y	**H**	**A**	L	L	E	N	N	R	I
6	E	P	T	K	-	-	N	V	O	K	S	I	**L**	A	Y	-	-	G	L
7	**W**	A	A	S	-	-	**M**	A	U	R	I	S	E	U	A	-	-	E	I
8	N	R	G	T	-	-	E	E	**L**	-	R	-	**M**	S	M	R	-	Y	P
9	O	C	G	R	O	**B**	G	-	-	**A**	**J**	**E**	A	N	N	E	T	T	**E**
10	D	K	Y	A	A	N	-	-	G	-	R	D	R	-	A	D	I	**M**	M
11	R	J	H	-	I	**C**	O	D	D	A	-	S	T	-	S	L	M	O	E
12	O	O	T	**W**	-	-	**E**	S	O	-	-	G	I	-	S	A	Z	H	A
13	**G**	N	O	-	-	-	I	**H**	-	-	-	E	N	-	I	**C**	I	A	G
14	R	E	R	-	-	R	Y	E	O	**B**	A	R	B	A	R	A	R	M	W
15	A	S	O	-	H	N	C	-	-	P	-	-	G	-	A	-	A	M	A
16	C	-	**D**	**C**	O	I	-	-	-	-	P	-	N	Y	**M**	-	W	E	L
17	E	-	**S**	**T**	R	A	C	H	E	Y	-	E	I	-	R	-	H	D	I
18	-	-	-	U	**D**	E	N	N	I	N	G	-	R	-	-	E	K	-	-
19	-	-	A	-	S	R	E	N	R	E	**B**	**M**	U	F	F	Y	L	-	-
20	-	**M**	**V**	O	N	N	E	U	M	A	N	N	**T**	**D**	A	N	**A**	**U**	-

The first letter of each answer is in bold. Solutions are given in the format:
answer word: (row, column, direction)

ADA (3,2,E) LOVELACE (3,5,E) ANITA (3,2,S) BORG (9,5,W), BARBARA (14,9,E) LISKOV (6,12,W), DANA (20,13,E) ULERY (20,17,NW) DOROTHY (16,2,N) DENNING (18,4,E) FRAN (2,15,S) ALLEN (5,11,E) GRACE (13,0,S) HOPPER (13,7,SE) JEANNETTE (9,10,E) WING (12,3,NE) KAREN (0,1,S) SPARCK-JONES (5,1,S) MARISSA (16,14,N) MAYER (8,14,N) MUFFY (19,11,E) CALDER (13,15,N) URSULA (0,18,W) MARTIN (8,12,S) WENDY (5,5,E) HALL (5,10,E)

ALAN (9,9,NW) TURING (20,12,N), CHRIS (16,3,NE) STRACHEY (17,2,E) EDGAR (12,6,NE) CODD (11,5,E) , EDSGER (9,11,S) DIJKSTRA (3,3,S) JOHN (1,18,W) VON NEUMANN (20,2,E) MAURICE (20,1,NE) WILKES (5,5,N) MOHAMMED (10,17,S) AL-KHWARIZMI (20,16,N), NIKLAUS (2,13,S) WIRTH (4,4,E), PHILIP (3,18,S) EMEAGWALI(9,18,S) SERGEY (3,17,S) BRIN (2,16,S) TIM (0,10,S) BERNERS (19,10,W) LEE(8,8,W),TONY (17,3,NE) HOARE (13,7,NE), VINT (3,7,NE) CERF (1,6,W)

Additional name
GORDON (13,0,N) WELCHMAN (7,0,N)

2. Cypher breaking grid

W 1	Z 2	M 3	F 4	Q 5	J 6	N 7	E 8	D 9	O 10	A 11	P 12	V 13
I 14	L 15	U 16	B 17	R 18	T 19	C 20	K 21	S 22	X 23	H 24	Y 25	G 26

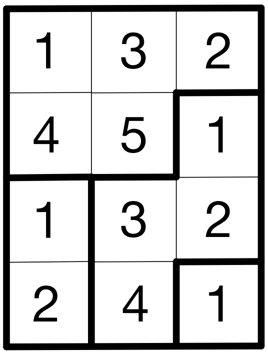
	S 22		A 11			A 11	H 24	A 11			E 8		I 14	
P 12	O 10	I 14	N 7	T 19	E 8	R 18		Q 5	U 16	E 8	E 8	N 7	S 22	
	F 4		I 14			E 8		U 16			L 15	O 10		Z 2
A 11	T 19	O 10	M 3	I 14	C 20		J 6	A 11	V 13	A 11		T 19	H 24	E 8
	W 1		A 11		O 10	R 18		D 9		R 18	Y 25	E 8		B 17
H 24	A 11	R 18	L 15	E 8	M 3		D 9	U 16	S 22	T 19		S 22		R 18
	R 18			M 3	E 8			C 20			M 3		T 19	A 11
K 21	E 8	Y 25	S 22			S 22	A 11	T 19	U 16	R 18	A 11	T 19	E 8	
				I 14		A 11					T 19	O 10	R 18	
	P 12	A 11	W 1	S 22		L 15	O 10	O 10	P 12		R 18	O 10	M 3	E 8
	Y 25		A 11		S 22	E 8	X 23				I 14		I 14	
S 22	T 19	A 11	S 22	H 24		S 22	O 10	C 20	K 21		X 23		N 7	O 10
	H 24		T 19	E 8	A 11	M 3		O 10					A 11	
G 26	O 10	L 15	E 8	M 3		A 11		M 3	A 11	G 26	N 7	E 8	T 19	S 22
	N 7		D 9			N 7	I 14	B 17			O 10		E 8	

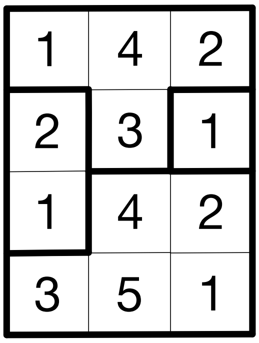
I 14 N 7   S 22 P 12 A 11 C 20 E 8   N 7 O 10   O 10 N 7 E 8

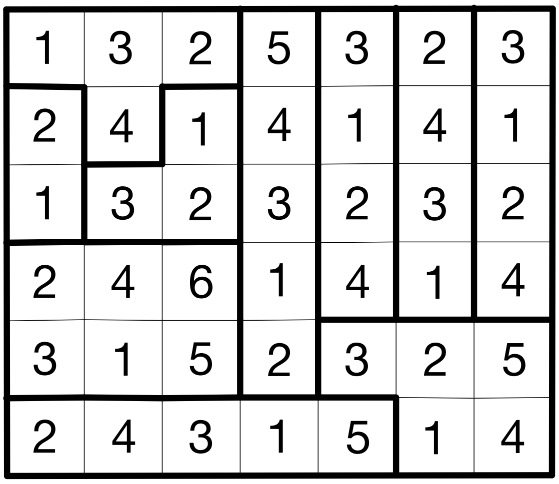
C 20 A 11 N 7   H 24 E 8 A 11 R 18   Y 25 O 10 U 16

S 22 C 20 R 18 E 8 A 11 M 3

3. Cut block puzzles







Download our booklet about solving Cut Block puzzles from www.cs4fn.org/puzzles/

4. Sherlock Syllogism

i) **b**: all rubies in the game are expensive in-game purchases.
(a) and (c) are also true though give less information

ii) **d**. None of the above.

iii) **b**. Some Rounding Errors are Poor Computer Software.

iv) **a**. Some websites are not educational

5. Word Ladder

Here is one solution:
LISP
LIMP
LAMP
LAMA
LAVA
JAVA

6. Bit Ladder

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

7. Debugging spot the difference

```
sum = float(num)
```
should be
```
sum = float(num1)
```

There is a missing quote in the last line:
```
'The sum of {0} and {1} is {2}
```
should be
```
'The sum of {0} and {1} is {2}’
```

There is a missing) at the very end of the last line.

8. Kakuru

\	4\	3\	15\	\	7\	6\	10\	15\	\
\7	1	2	4	\13	1	4	3	5	\
\6	3	1	2	5\10	4	2	1	3	\
\	3\	6\7	1	4	2	19\5	4	1	4\
\10	2	4	3	1	7\10	1	2	4	3
\8	1	2	5	\10	1	9	6\3	2	1
\	17\	16\	\	14\6	2	3	1	7\	6\
\16	9	7	16\21	1	4	6	5	2	3
\30	8	9	7	6	\	\	\3	1	2
\	\	\16	9	7	\	\	\5	4	1

9 Bakuro

			5 0101	15 1111		
	11 1011	3 0011 13 1101	1 0001	2 0010		
15 1111	2 0010	8 1000	4 0100	1 0001	15 1111	
5 0101	1 0001	4 0100	12 1100	4 0100	8 1000	
9 1001	8 1000	1 0001	9 1001	8 1000	1 0001	9 1001
				3 0011	2 0010	1 0001
				12 1100	4 0100	8 1000

10. Extreme logical thinking: Eating at Quonk

That answer is 3: Babs and Yabu eat at Quonk
Why?
We know Alice and Zach eat together at Quonk. That means that EITHER

- the woman who likes to eat there is Alice OR
- the man who likes to eat there is Zach.

It is not BOTH Alice and Zach who like to eat there as then they would have eaten somewhere else.
This means that the woman and man in question is EITHER

- Alice and Yabu OR
- it is Babs and Zach.

We cannot tell which though. Luckily, that does not matter, we have enough information. In both cases we get the same conclusion.
If it is Alice and Yabu that like Quonk, then:

- Babs and Yabu would eat there because of Yabu.
- Alice and Yabu wouldn't because both are there.
- Babs and Zach wouldn't because neither Babs nor Zach eat there.

If it is Babs and Zach that like Quonk then:

- Babs and Yabu would eat there because of Babs.
- Alice and Yabu wouldn't because neither Babs or Zach are there.
- Babs and Zach wouldn't because both are there.

So whichever the pair is:

- Babs and Yabu would eat there.
- Alice and Yabu wouldn't eat there.
- Babs and Zach wouldn't eat there.

11. Extreme logical thinking: A Trip to Market

Here is one possible answer, there is another efficient solution that is similar.

1) Farmer travels across with the hen (Dog left with Corn but that is okay).
2) Farmer returns.
3) Farmer travels across with Dog.
4) Farmer returns with Hen (as otherwise dog will eat it)
5) Farmer travels across with corn.
6) Farmer returns (leaving dog and corn again)
7) Farmer travels across with the hen.
8) ***Farmer sends the seat back to the other bank.***

It is very common to get this puzzle wrong by forgetting to return the seat at the end returning the seat. The aim is to get everything across so the farmer can carry on home. However, once that was done you had to put the seat back to the side it came from. If you made the mistake you cannot claim you did not know. It was there in the instructions, and you were even told why it was important. Making this kind of mistake where there is a last tidying up task to do is so common it has a special name - it is called a post-completion error. It appears in lots of situations like forgetting a credit card at a supermarket checkout or self service machine. When designing software you have to design the way the task is done too and avoid last steps that are easily forgotten like this.

12. Compression Codes

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

b)
Great fleas have lesser fleas,
Upon their backs to bite 'em,
And lesser fleas have lesser fleas,
and so, ad infinitum.

And those great fleas, themselves, in turn
Have greater fleas to go on;
While those again have greater still,
And greater still, and so on.

13. Debugging spot the difference

Line 1
The word void is missing.
{} should be []
The ; should be a ,
int m should be int n

Line 3
The , should be ; (twice)
< should be <=
p=0 should be p=1

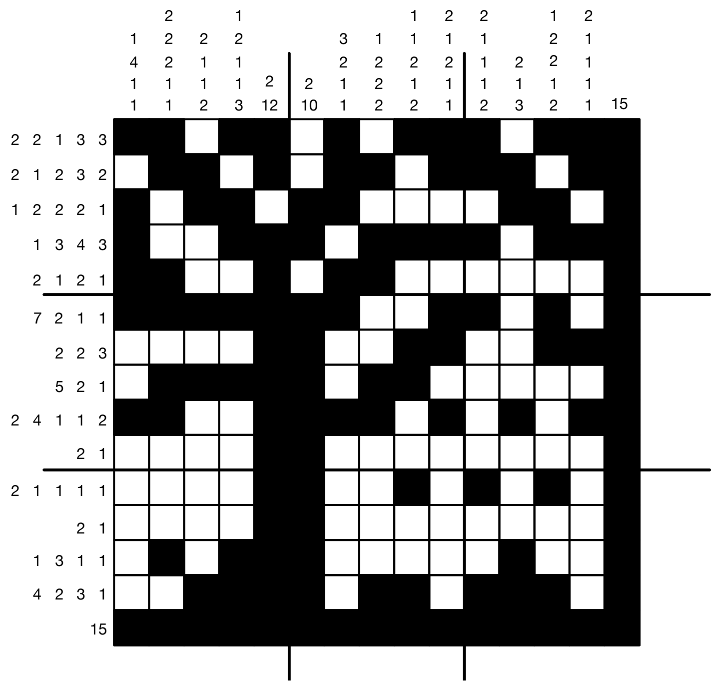
Line 5
int i should be int i=0
q should be p
The F in for shouldn't be capitalised.

Line 7
Remove the ; from the end
< should be >
Missing) at end

Line 9
S in Swap shouldn't be in capitals.
i+l should be i+1 (the number 1, not the letter l)
Missing ; from end

Line 13
Missing }

14. Compressed Pixel Puzzles



15. The Tour Guide

There are many solutions. One solution is:
Hotel – Science Museum – Toy Shop – Big Wheel – Park – Zoo – Aquarium – Art Gallery – Wax Works – War Ship – Castle – Cathedral – Hotel

16. The Knight's Tour

There are many solutions. One solution is:
1-9-3-11-5-7-12-4-10-2-8-6-1
In fact, this puzzle is actually the same as the Tour Guide puzzle. If you draw a map showing which squares can be jumped between you get a picture that looks just like the Tour Guide map.

This puzzle was adapted from an idea by Maciej Syslo and Anna Beata Kwiatkowska, Nicolaus Copernicus University.

17. The Bridges of Königsberg

There is no solution. It is impossible to create such a tour without going back over the same bridge twice. Any suitable route must visit every land area (islands and the two banks). It must also involve every bridge but only once. Let's suppose there is such a route and we draw a red line over the map to show it. All the bridges must be on the route so should be coloured red. Now think about a land area on that route. It must have a red line in to it for every red line out from it. Otherwise, the route will get stuck when it arrives on that extra bridge. There will be no way out without going back over a bridge already crossed. The same reasoning applies to every land area. That means all land areas must have an even number of bridges connected to them if there is such a route. All the land areas on the Königsberg map have an odd number of bridges, so there is no such tour possible.

Download our booklet about the above three puzzles (15,16,17) from www.cs4fn.org/puzzles/

18. The Swap Puzzle

The following are efficient solutions for each board.

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**

The level 3 puzzle can be solved in 15 moves as follows:

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

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

Step 3: **Square 4** GETS THE PIECE FROM **Square 5**

Step 4: **Square 5** GETS THE PIECE FROM **Square 3**

Step 5: **Square 3** GETS THE PIECE FROM **Square 1**

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

Step 7: **Square 0** GETS THE PIECE FROM **Square 2**

Step 8: **Square 2** GETS THE PIECE FROM **Square 4**

Step 9: **Square 4** GETS THE PIECE FROM **Square 6**

Step 10: **Square 6** GETS THE PIECE FROM **Square 5**

Step 11: **Square 5** GETS THE PIECE FROM **Square 3**

Step 12: **Square 3** GETS THE PIECE FROM **Square 1**

Step 13: **Square 1** GETS THE PIECE FROM **Square 2**

Step 14: **Square 2** GETS THE PIECE FROM **Square 4**

Step 15: **Square 4** GETS THE PIECE FROM **Square 3**

19. Word search

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	S	Y	N	T	A	X	E	N	G	I	N	E	R	A	B
1	A	C		H	C	R	A	E	S	P	A	M	E	S	
2	V	F	A	S	H	I	F	T			S	T	T	E	S
3	E		R	N				W			O	O	U	C	O
4				U	N			O	S		F	H	P	U	C
5	T				S	E		R	C		T	S	M	R	I
6	O						R	K	R		W	P	O	I	A
7	H	S	U	T	A	T	S	I	I		A	A	C	T	L
8	S	P	Y	W	A	R	E	N	P		R	N	R	Y	R
9	N	S	H	E	L	L	R	G	T	Y	E	S	E		E
10	E				L		V		E			C	P		M
11	E			O			E	K				R	U		M
12	R		R	S	H	A	R	E	W	A	R	E	S		A
13	C	C		S	P	R	E	A	D	S	H	E	E	T	P
14	S	T	O	R	A	G	E					N	A	C	S

The answer words and their positions in the grid are given in the following format:
answer (row,column,direction)

```
BAR(0,14,W)
ENGINE(0,6,E)
KEY(11,7,NE)
NETWORKING(0,7,S)
SAVE(0,0,S)
SCAN(14,14,W)
SCANNER(0,0,SE)
SCREEN(9,11,S)
SCREENSHOT(14,0,N)
SCRIPT(4,8,S)
SCROLL(14,0,NE)
SEARCH(1,8,W)
SECURITY(1,13,S)
SERVER(7,6,S)
SHAREWARE(12,3,E)
SHELL(9,1,E)
SHIFT(2,3,E)
SNAPSHOT(9,11,N)
SOCIAL(2,14,S)
SOFTWARE(2,10,S)
SPAM(1,8,E)
SPAMMER(14,14,N)
SPREADSHEET(13,3,E)
SPYWARE(8,0,E)
STATUS(7,6,W)
STORAGE(14,0,E)
SUPERCOMPUTER(12,12,N)
SURF(5,4,NW)
SYNTAX(0,0,E)
```

20. Word Ladder
Here is one solution:

ALICE
SLICE
SLIME
SLIMS
SLITS
FLITS
FLATS
FEATS
TEATS
TENTS
VENTS
VENUS
GENUS
GENES

21. Lovelace's Life Ladder

1815
1812
1852

Download a special issue of cs4fn magazine on Ada Lovelace from www.cs4fn.org/puzzles/

22. Bit Ladder

Here is one solution:

```
0 0 0 0
0 0 0 1
0 0 1 1
0 0 1 0
0 1 1 0
0 1 1 1
0 1 0 1
0 1 0 0
1 1 0 0
1 1 0 1
1 1 1 1
1 1 1 0
1 0 1 0
1 0 1 1
1 0 0 1
1 0 0 0
```

23. Napier's Life Ladder (from birth to bones)
1550
1650
1657
1617

24. Dave's data dilemma

This magic trick is based on a mathematical foundation called a forcing matrix. You can easily create one of these yourself for any force number you want, a force number is the number that will always turn up at the end, here we used 58 as an example. The principal works for any square matrix size 4 by 4, 6 by 6 and so on you just need to think how the size of the matrix this fits into the story you're going to tell in the performance when creating your trick.

Here's the secret, for a 5x5 force matrix (we used the 5 weekdays in the trick), select any ten numbers (2x5), which we will call the seed numbers, that sum to your desired force number(in this case 58). For example, 6 + 1 + 4 + 7 + 12 + 7 + 3 + 4 + 11 + 3= 58.

Place these seed numbers in any order round a 5 x5 grid, this grid when complete will be your forcing matrix. Now fill in the grid number by number by adding the seed number in the row to the seed number in the column to get the grid number value for each position of the 25 numbers in the grid. For example the first number in the matrix 13 is made by adding 7 (the row seed) and 6 (the column seed) to give the value for that matrix entry of 6+7=13. If we are at the position 3, 3 in the grid, the row seed is 4 and the column seed is 4 so the number is 4+4=8.

	M	T	W	T	F
	<u>6</u>	**<u>1</u>**	**<u>4</u>**	**<u>7</u>**	**<u>12</u>**
A **<u>7</u>**	13	8	11	14	19
B **<u>3</u>**	9	4	7	10	15
C **<u>4</u>**	10	5	8	11	16
D **<u>11</u>**	17	12	15	18	23
E **<u>3</u>**	9	4	7	10	15

Your forcing matrix ready, now is the time to use it. You need to come up with a clever story in the performance that requires you audience to choose each of the column of numbers, here we used choosing a different day of the school week to make sure our audience select each of the columns, and this selection can be in any order. For each column (day M T W T F) selected the audience need to choose a different row position in the selected column each time, here we used friends names (call them A B C D E).

What does this mean mathematically? Ensuring we use every day (M-F) and ensuring a different name is chosen on each day (A-B) means all the chosen numbers we add at the end each contain the row and column seed numbers we used to create the forcing matrix in the first place. For example choosing Monday ensures we have a contribution from Monday column seed 6, and choosing friend B on Monday ensure we have a contribution from row seed 3, so both seeds are contained in the chosen number 9. Ensuring that the next selection is a different day, and a different friend will add another set of two column and row seeds onto the final total, and so on. The final sum total is therefore guaranteed to contain all the seeds hidden in the numbers selected, and will always add to the force number 58,

Magicians note

This mathematical magic trick of using a forcing matrix is believed to be first written about in a magic book by Maurice Kraitchik in 1942, but the idea has been used by other magicians with a range of different presentations including Walter Gibson, Mel Stover, Stewart James, Martin Gardner, Paul Hallas and Max Maven amongst others.

25. Cryptogram

The message was:
```
THE YOUNG ADA DREAMT OF BEING ABLE TO FLY AND APPROACHED THE
PROBLEM OF HER EARTH BOUND STATE WITH TYPICAL GUSTO AND A
SCIENTIFIC EYE
```

The following key was used to encrypt it though not all letters appear in the message. Though not all letters ultimately appear so the full key can't be reconstructed from this message.

K 1	D 2	M 3	V 4	N 5	T 6	A 7	O 8	U 9	B 10	J 11	I 12	S 13
X 14	Z 15	L 16	C 17	W 18	F 19	Y 20	G 21	P 22	R 23	E 24	H 25	Q 26

26. Extreme logical thinking: Card sorting, well sort of

1) **You must turn over card A and card D**

Did you get this one wrong? This is called the Wason selection task and was devised by Peter Cathcart Wason in 1966. It may well have fooled you! Psychology researchers have shown only about 5% of the population gets it right. So well done if you did get it right. Though we think we act logically, often we don't. Most people think the answer is to turn over card A and card C which wouldn't prove

Why? The statement "every card with a vowel on one side has an even number on its opposite side" can only be shown to be false if there was an odd number on the opposite site of one known to be a vowel card (ie card A) and/or a vowel on the opposite side of one known to be an odd numbered (i.e. card D).

2) **You must turn over card B and card C**

Later in 1992 researchers Cosmides and Tooby found that people could do this task and select the correct cards, if they were given a version of the test that was relevant to a social situation, like the one here about whether someone is old enough to buy fireworks. Most people get this right even though they found the first hard and logically they are exactly the same problem.

We are better at logical thinking that is socially relevant - we are social creatures more than logical ones, which is why we need to train our skills in logic.

27. Girls and Boys

This is a problem in both logic and probability. We have two equally probable possibilities:

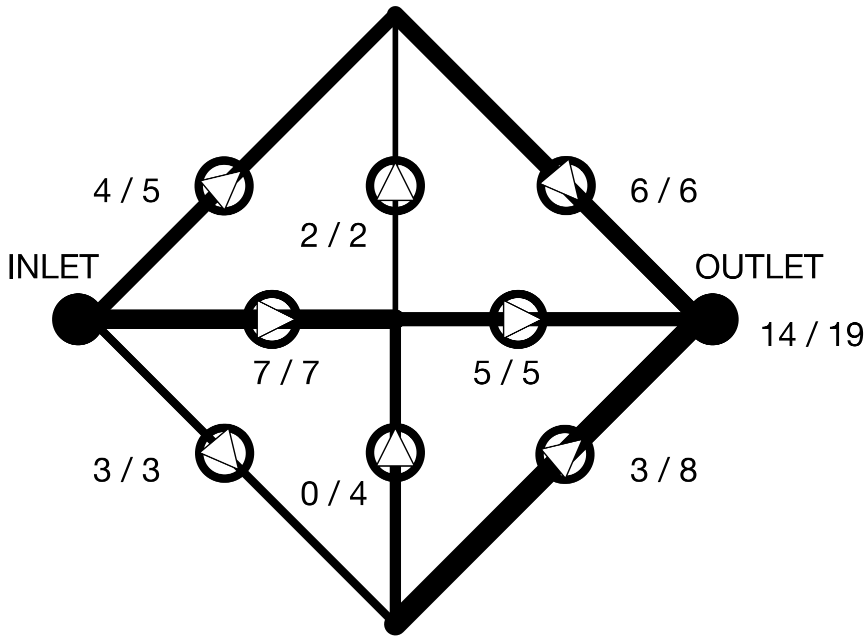
Possibility one - The younger child is a boy.

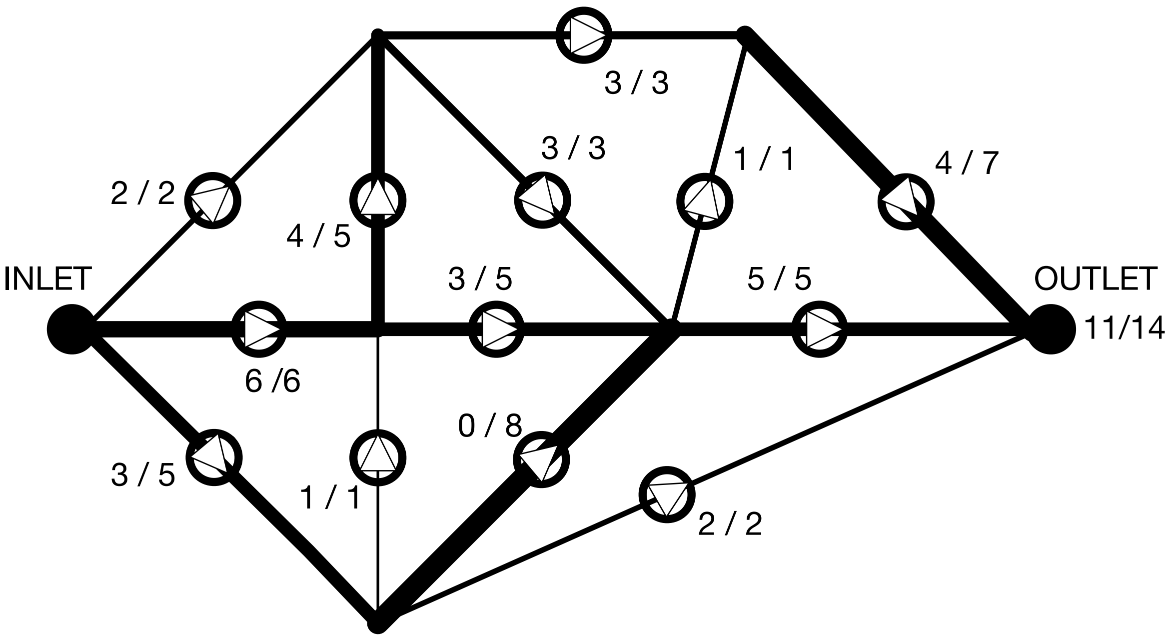
Possibility two - The younger child is a girl, so both children are girls.

Therefore using the information we have, the probability of the Professor having two girls is 1/2. Whatever the probability the Professor was a wonderful mother to both.

28. Pinky's Pipe Pickle

The maximum flow is 14. Here is one solution that gives that maximum flow.





29. Codeoku

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

30. Navigating the Numbers

The answers and their positions in the grid are given in the following format: answer (row,column,direction)

(a) 2015 (3,3,N) (b) 217 (3,0,NE) (c) 9367 (7,7,W) (d) 1815 (8,0,E) (e) 1024 (1,3,E) (f) 1791 (6,4, SE)

(g) 1972 (0,0,S) (h) 3456 (0,1,E) (i) 1852 (9,1,E) (j) 2652 (9,6,E) (k) 1003 (8,9,W)

(l) 8192 (4,6,NE) (m) 138 (4,0, E) (n) 2567 (0,9,SW) (o) 34835 (4,4,E) (p) 3587 (3,8,S)

(q) 5912 (6,0,E) (r) 3213 (4,4,SW) (s) 2311 (5,3,W) (t) 1981 (0,5,E) (u) 4920 (6,7,W) (v) 1954 (2,9,S)

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

31. Bakuro

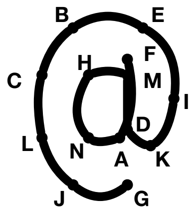
	9 1001	6 0110		7 0111	1 0001	15 1111	
5 0101	1 0001	4 0100	7 0111 14 1110	2 0010	1 0001	4 0100	3 0011
15 1111	8 1000	2 0010	4 0100	1 0001	3 0011	2 0010	1 0001
		6 0110 13 1101	2 0010	4 0100	10 1010 15 1111	8 1000	2 0010
	12 1100 12 1100	4 0100	8 1000	3 0011 3 0011	2 0010	1 0001	
12 1100	4 0100	8 1000	10 1010	2 0010	8 1000	10 1010	
9 1001	8 1000	1 0001	7 0111	1 0001	4 0100	2 0010	
				9 1001	1 0001	8 1000	

32. History Timeline Dot to Dot

The order to join the dots in the order:

G J L C B E I K D M H N A F

@



1. G (440BC) The idea of steganography, the practice of hiding messages, is recorded for the first time in the book ‘Histories’ by Herodotus.
2. J (~60BC) Julius Caesar uses what becomes known as the Caesar cipher to encrypt his messages.
3. L (~825 AD) Al-Khwarizmi writes his book ‘On the Calculation with Hindu Numerals’: the translation of which gives the word algorithm
4. C (1587) Mary, Queen of Scots is executed as a result of Walsingham decoding her secret messages plotting to kill Elizabeth I using frequency analysis.
5. B (1605) Francis Bacon invents the Bacon cipher, a way of encoding letters in binary for use in secret writing.
6. E (1679) Gottfried Leibniz discovers the modern binary number system.
7. I (1737) Jacques de Vaucanson builds *The Flute Player,* the first biomechanical automaton. It is a life-size figure that can play 12 different songs.
8. K (1801) The Jacquard loom is first demonstrated. It revolutionises weaving and its punch card system that control the pattern is a foundation for the idea of stored computer programs and data storage.
9. D (1843) Ada Lovelace’s notes on Charles Babbage’s proposed analytical engine lead to her being hailed as the first programmer.
10. M (1936) Alan Turing invents the Universal Turing Machine, a mathematical precursor to the stored program computer.
11. H (1952) Grace Hopper invents the first programming language compiler, making code far easier to write.
12. N (1960s) ALGOL, the first block structured programming language is invented influencing virtually all languages that follow.
13. A (1972) Karen Spärck Jones invents IDF weighting, the algorithm behind most search engines, making it practical to search the web.
14. F (2006) Jeanette Wing popularises the idea of Computational Thinking as the core, generally useful, skill set of the computer scientist.

33. Compression Codes

ninety-nine little bugs hiding in the code,
ninety-nine little bugs hiding in the code,
one bug is fixed … and we compile it all again,
and there are a hundred little bugs hiding in the code.

34. Troubling communication

a) There are many ways this could be done. If blinking is hard you need a way to make sure the patient have to blink as few times as possible for each letter. Probably the simplest way is that the Doctor slowly says the alphabet out loud a letter at a time, and the patient blinks when the letter they are thinking of is said. The Doctor writes that down and then starts again.

b) If blinking is easy they can blink multiple times per letter. One way would be to use Morse code though that might be hard for both people to remember. Another possibility would be to hold up a series of cards showing groups of letters and asking if it was in that group. This might start with the letters in the first half of the alphabet. If it is one of those letters the patient blinks. If not they don't blink. Then depending on the answer you show a card showing half the letters left. You repeat this until there is only one letter left.

35. Program search

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	i	f						(	:	1	=	=	)				m
1	p			t	n	i	r	p					o	)			o
2	i		t	h	a	n	o	i	"	-	"	,	t	m			v
3	e		h	a	(	p	i	e	c	e	s	,	e	o			e
4	c		a	n				c					v	r	,		t
5	e		n	o				e	l	s	e	:	o	f	m	m	o
6	s		o	i		d		s	g	n	i	r	m	e	o	o	)
7	(	p	i	e	c	e	s	-	1	,	o		o	v	r	v	
8			m			f		1			t		v	o	f	e	:
9	e	v	o	m	"	(	1	,			h		e	m	e	f	)
10	m	o	v	e	f	r	o	m	,		e		t		v	r	r
11	,	r	e	h	t	o	o	t	h	e	r	)	o		o	o	e
12			t	h	a	n	o	i			,		,		m	m	h
13		m	o	v	e	f	r	o	m	,	f	r	o	m	"	,	t
14			,	m	o	v	e	t	o	,							o

The words including punctuation as they appear and their positions in the grid are given in the following format: answer (row,column,direction)

```
def          (6,5,S)
else:        (5,7,E)
from",       (13,10,E)
if           (0,0,E)
movefrom)    (9,13,N)
movefrom,    (12,14,N)
movefrom,    (10,0,E)
movefrom,    (13,1,E)
movefrom,    (5,15,S)
moveto)      (6,12,N)
moveto)      (0,16,S)
moveto,      (6,12,S)
moveto,      (14,3,E)
moveto,      (8,2,S)
other)       (11,6,E)
other):      (14,16,N)
other,       (11,5,W)
other,       (7,10,S)
pieces       (1,0,S)
print        (1,7,W)
ring         (6,11,W)
thanoi       (1,3,S)
thanoi       (2,2,S)
thanoi       (12,2,E)
thanoi       (2,2,E)
(pieces-1,   (0,7,S)
(pieces-1,   (7,0,E)
(pieces,     (3,4,E)
("move       (9,5,W)
(1,          (9,5,E)
==           (0,10,E)
1:           (0,9,W)
"-",         (2,8,E)
```