The Magic of Computer Science:
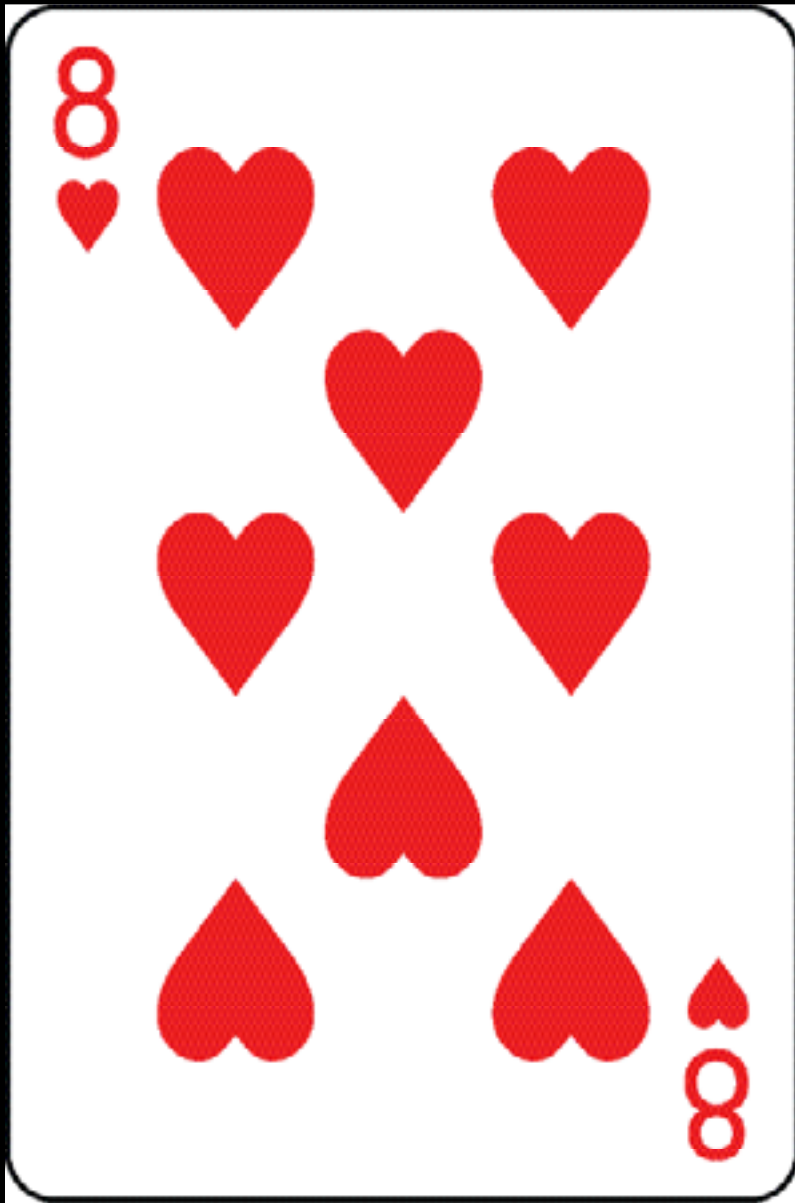
Card Tricks Special

or
A plethora of
pasteboard
paradoxes
purporting the
principles of
Computer Science
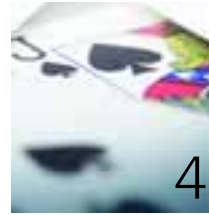
Presented by
Peter McOwan and Paul Curzon
of the Department of Computer Science,
Queen Mary, University of London
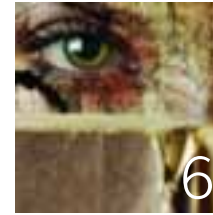with support from **www.cs4fn.org**

Queen Mary
University of London

This is your chosen card
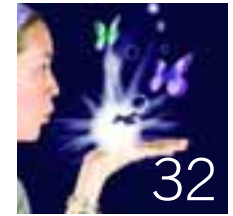
Contents

Magic and Computer Science

"Pick a card, any card!" How often have you heard magicians say that? The normal routine is that you pick a card, the magician shuffles the deck, and abracadabra, reveals your chosen card. But behind this magic often lies some interesting maths and ideas used in computer science, and as we shall see, magicians' shuffles have actually led to the development of new ways for computers to work. It's hardly surprising then that some of the great Magicians have also been Computer Scientists or Mathematicians.

Burn the witches!

If, 500 years ago, you had claimed to be able to communicate instantaneously with someone on a different continent you would have probably been burnt at the stake as a witch. Nowadays we all can do it anytime, anywhere – using our mobile phones. Magic has become reality. Of course the technology does not need the imagined psychic powers of the Mystics.

Magicians, Penn and Teller demonstrated the principle in what may have been the most expensive trick of its time when done in 1990s. The televised trick took place live on the streets of London, in Piccadilly Circus. They asked a passer-by to choose a card from a normal pack of playing cards. The person took care not to show the card to the Magicians or the camera which were in any case behind Penn. The Magician shuffled then fanned the cards. He could immediately say which card was chosen. Psychic powers? No. High technology? Yes.

Science Fiction writer Arthur C Clarke summed it up with his now famous quote: *"Any sufficiently advanced technology is indistinguishable from magic."*

How did they do it? The camera couldn't see the chosen card. It could see the rest of them though. Vision software analysed the picture of the fanned out cards. It quickly identified all the cards that *were* present using state-of-the art image recognition software and so determined the one that was missing. The computer then connected to the neon billboards in Piccadilly Circus and swapped the advert for a giant message naming the missing card. As this was behind the back of the volunteer, they had no idea. It seemed amazing. Not magic though, technology. It now really is everyday technology, used on the streets of London, recognising car number plates in the congestion charging zone!

How to use this book

Magic

This booklet contains a series of card tricks. Each is presented in two parts. First we present the magic in enough detail that with practice you should be able to do the trick yourself. This first part comes in three sections. First we describe the effect that you are aiming to create. We then describe the detailed mechanics of the trick – actually what you do that ensures the trick works. Finally, each trick comes with a showmanship section. It suggests alternative ways to present the trick that may give that all-important extra 'wow' factor. You can also experiment with your own variations once you know the core secret that makes the trick work.

Computer Science

All the tricks have a link to Computer Science, though not in the obvious Penn and Teller way of using clever technology to pull off the trick. The computer science link is to something deeper than today's technology, something fundamental in the subject. The second section of each trick describes this Computer Science link. We hope you will find the science and maths as fascinating as the tricks.

Keep the Magician's Code

Some of these effects are actually in the shows of professional magicians. We present them here for educational and entertainment purposes. If you do perform them later for friends then don't break the magicians code. Never reveal the secrets in your tricks to your audience.

Keep the secrets!

It's in the presentation

Keep the secrets, yes, but also remember that in magic, presentation is just as important as the secret. Ultimately it is the final effect on your audience that matters. With a slightly better presentation a trick that last week had a mediocre reception can suddenly cause gasps from your audience.

As we will see, computer programming is very similar to magic in this way. Programs combine the code to do the job (like the secret) with a user interface (like the magician's presentation) through which the human user interacts with the program. Just as with a trick, the same program can be transformed from unusable junk to a best seller by getting the interface right. iPods are one obvious example of the effect a revolutionary interface can have on sales. Revolutionary magic can just as easily be down to the presentation too.

The 21-card trick: the one where you read minds

The magicical effect

A volunteer shuffles a pack of cards. You deal out single cards, left to right into three piles of seven cards, all face up and visible. Your volunteer mentally selects one of the cards. You read their mind and tell them the card they are thinking of...

Mind reading of course is not that easy (unless your volunteer is a very clear thinker with a thin skull), so you may need a bit of help.

They mustn't tell you which card it is, but get them to tell you the pile it is in. You collect up the cards, and deal them out a card at a time left to right into three piles once more. Again they tell you the pile their card is in, you collect the cards once more, saying you're struggling to "read their mind". Deal the cards out across the table in the three piles again in the same way. Your friend indicates the pile their card is in. Collect the cards again and deal them into the three piles one last time. You immediately announce their card and magically it is in the very middle position of the pack.

The mechanics

Let's look at the 'mechanics' of the trick: how do you make it work? It involves several deals, each apparently shuffling the order of the cards, but doing so in a rather cunning way. In fact it's really rather simple.

All you have to do is make sure you always put the pile your volunteer selects carefully between the other two piles and deal the pack as above. Do that and after the fourth deal the middle card of the middle pile is the chosen card, which you can reveal as you see fit. If you are having trouble getting it to work, see our more detailed instructions with pictures at www.cs4fn.org/mathemagic/magicshuffles/ There is even a computer program there that can do the trick itself (and so read your mind over the Internet)!

Laying out the 21-card trick

The 21-card trick: the one where you read minds

The showmanship

Showmanship is important for a good trick. You need some patter to make things more fun and also distract attention from what is really happening. You can come up with your own ideas but here is a version we do.

After first dealing out the cards, stare into the person's eyes as you try and read their mind. Tell them they shouldn't giggle as giggles bubbling up get in the way of the thoughts. (They probably will then struggle not to giggle). Say you need to try again as there were too many giggles. On the second deal try it from the back of the head. After all (you explain) the front of the skull is the thickest part as it is important to protect your brain. Remind them not to giggle… complain it's not working as all they are thinking about is not giggling instead of the card! You will need to deal again. Try this time through their ears – stare hard and you will probably get the colour at least. One more deal and you will have it. Double check through the other ear to make sure it looks the same and you have it! Gradually turn over the ones they weren't thinking of, a few at a time (maybe make a mistake turning over the middle column then correct yourself). Finally their card is the one left face up.

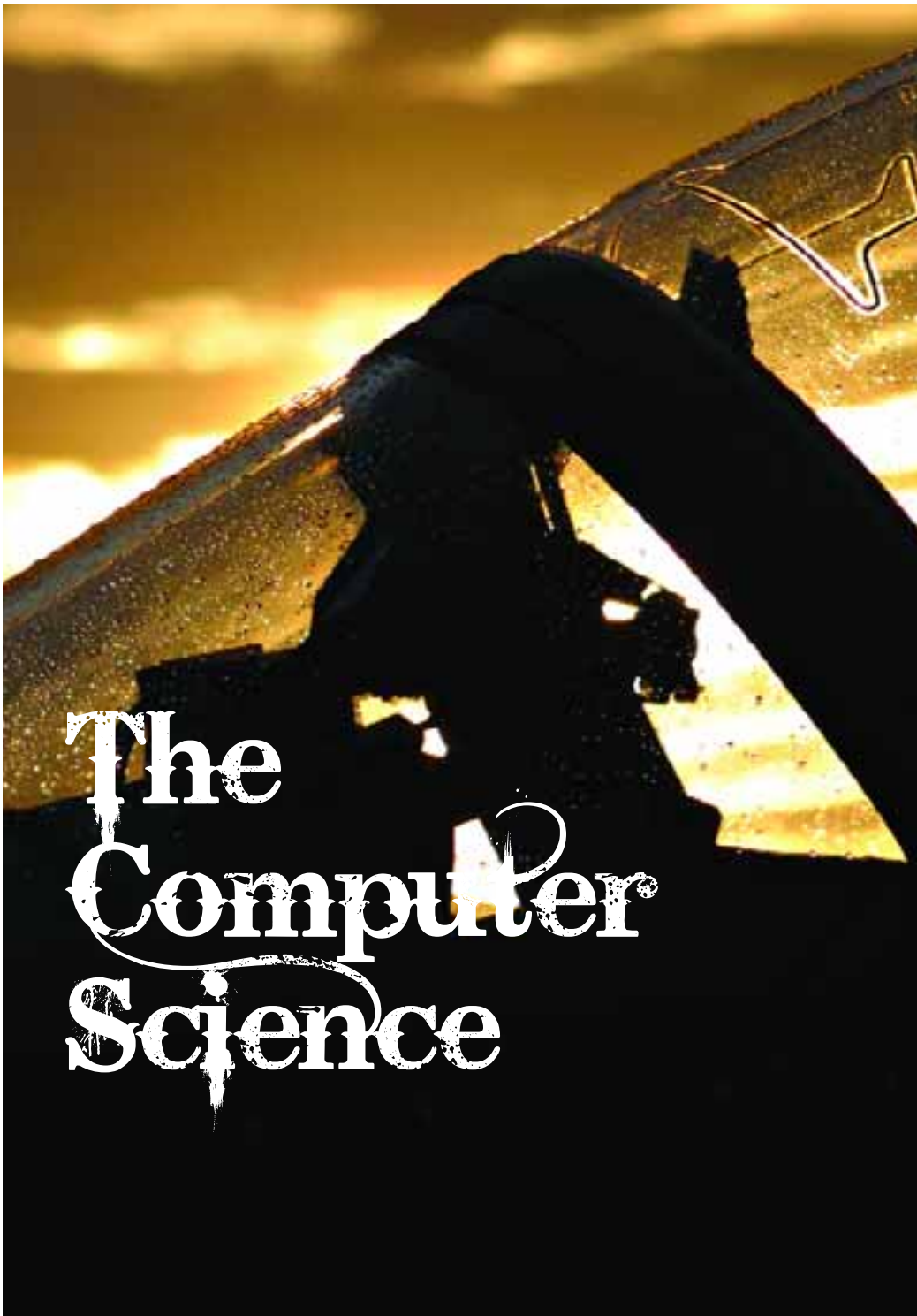## Magic and computers – developing your own algorithms

Once you understand the mechanics of a trick and why it works you can play with some ideas. The order of the chosen pile must not be changed, but the two other piles could for example be shuffled before being put together. As long as the chosen pile goes undisturbed between the two other piles of seven cards the order of the other cards doesn't matter. You might want to try and come up with your own additional twists and ways to build them into your presentation now you know how it's done.

A Bit about Magicians

Persi Diaconis was a professional magician, but his passion to debunk crooked casino games pulled him into advanced mathematics. He is now a Stanford professor of Mathematics and Statistics studying the randomness in events such as coin flipping and shuffling playing cards.

He and fellow mathematician David Bayer have shown that you need to give a pack of cards seven dovetail shuffles before the cards are really in a random order.

Films We Loved

The Prestige is a great Oscar™ -nominated film about the rivalry of Professional Magicians, Science and perhaps(?) supernatural powers.

Learn more at **www.cs4fn.org/mathemagic/**

Learn more at **www.cs4fn.org/mathemagic/**

The Computer Science

The 21 card-trick: The Computer Science

Step by step

You want to be sure a magic trick always works. After all, it may work 99 per cent of the time but could you be sure that the one time you're trying to impress a friend or in front of a big audience it would not be the one per cent it didn't work? I know what my luck is like!

Some tricks need your skill at sleight of hand to work. The ones we prefer always work. Computer Scientist's call them **'algorithmic'**. An algorithm is just a clear set of actions to be taken in a given order that achieve some task. Guaranteed!

The steps that you go through to get the 21-card trick to work are like this. They are also similar to the way that a computer steps through its instructions in a software program. All that computers do, in fact, is follow instructions. They follow algorithms that programmers work out for them. The idea is that if they follow the algorithm then they will always complete their task, whether it is playing chess, sending your emails or flying a plane. Every program you have ever used is working the same way as an oversized magic trick.

The point about an algorithm is that if you follow its instructions exactly, you are guaranteed to achieve what you are trying to do…if the algorithm is correct. What if it isn't? Are we really sure our trick always works, whatever?

Testing times

How could we make sure our algorithm is correct and our trick does work? Well we could do the trick lots of times and check it works every time. Computer Scientists call that **'testing'**. It's the main way programmers make sure their programs are correct. They run the program lots of times with different data. Would that be enough to be sure, though?

How many times would we need to do the magic trick to be safe? To be really certain it looks like we would have to try it out with every possible set of 21 cards, in all possible starting positions, checking for every card the person might have thought of.

Try it… How many orders did you do before you got bored? It's a lot of combinations… there are far too many to test them all. It would take an impossibly long time. Similarly testing programs exhaustively like this is not practical. Most programs are far more complicated than this simple trick after all. Instead, as many combinations as possible are tested given the time available. If it works each time then the programmer assumes it works in the cases they didn't try too (and hope!)

That is why there are so often bugs in programs – too much hope, not enough testing!

The 21 card-trick: The Computer Science

There must be a better way!

Perhaps we can be a bit cleverer than that though and work out a shorter set of tests that still give us the guarantee that our trick always works. With a bit of thought it's obvious it doesn't actually matter what any of the cards are. All that matters are the 21 start positions. If a card in the first position ends up in the centre when we test it, we can reason that every time, if a person thought of the card in that position at the start, it will end up in the middle. With this little bit of reasoning we have reduced our testing problem to only 21 tests: one for each starting position. Programmers use similar kinds of reasoning, based on their knowledge of the structure of the program to reduce how many tests they do too.

Prove it!

In fact we can go further and do some more reasoning to prove the trick always works. If the proof has no flaws then it proves the trick (or program) works whatever the combination …and you don't need to test any of them. It might be a good idea to still do some testing though. After all, you could have made a mistake in your proof!

It boils down to the fact that putting the chosen pile (column) in the middle of the other two piles and re-dealing the cards in effect limits where the chosen card can go. Let's work through it.

After Deal Number 1: After the first deal of the cards into three piles, the seven-card pile holding the chosen card is put in the middle of the other two. There are now only seven places it could be.

After Deal Number 2
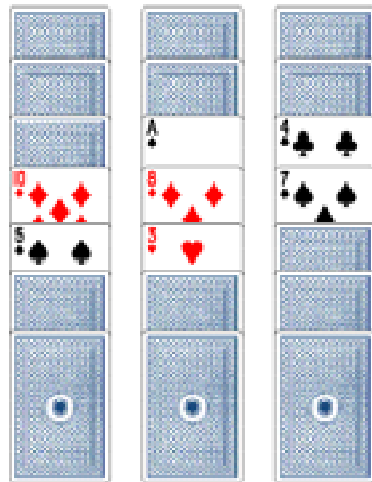
You deal the cards into three new piles. Where do those seven cards from the middle pile go? Anywhere? No. The seven possible places are: the fourth or fifth card of the first pile; the third, fourth or fifth card of the middle pile, or the third or fourth card of the last pile. They are just the middle cards of each pile (as above). The volunteer tells you which pile again, and you again put that pile between the other two. The chosen card must be in the third, fourth or fifth position of the middle pile now. Only 3 possible places are left.

After Deal Number 3
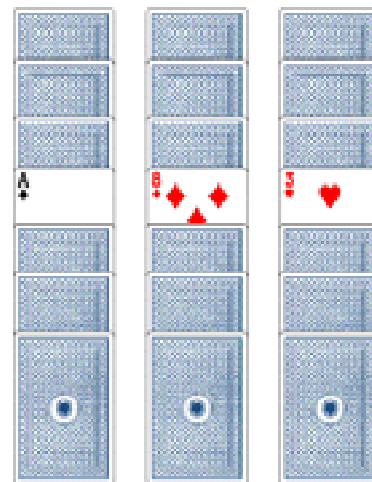
You deal again. This time, the card has to be the fourth card – the middle card – of the first, middle or last pile. Why? There were only three possible places and they each get moved to the middle of their pile as they are dealt out again. In fact more than 40 per cent of the time, it will be in the middle pile (can you see why?), so that's a good pile for you to guess if you want. Once your friend tells you which of the three piles has their card, you know exactly where their card is.

After Deal Number 4
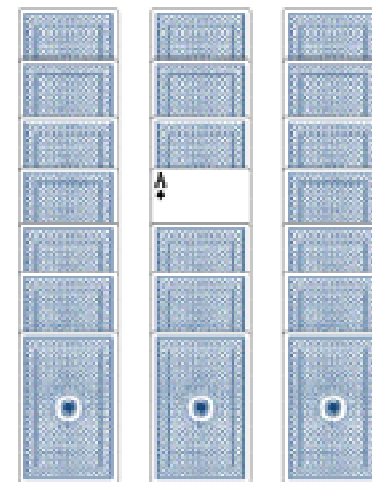
The fourth deal moves the chosen card to the middle of the middle pile… just for effect.

The correctness of algorithms

What we have just done is give a convincing (we hope) argument that the trick or algorithm always works. That is all that mathematical proofs are: convincing arguments where there is no room for doubt if you follow the detail. Here we were just proving that a trick works, but as we saw the instructions of the trick are an algorithm – just like a computer program. It's very important that programs always work too. We can therefore similarly do proofs about the algorithms behind programs. Proofs are just one of the ways computer scientists have developed to help find bugs in programs, and it's useful for finding them in computer hardware too.

A Perfect Shuffle:
the one where you magically shuffle a card to a position of your choice

The magical effect

The magicians' art of shuffling in special ways to make tricks, like the 21-card trick, work can also help us build computers. Magicians want to move cards around efficiently; computers want to move data around in their memory efficiently.

In a perfect shuffle, the magician cuts the cards exactly in half and perfectly interlaces them, alternating one card from each half. It takes years of practice to do but looks massively impressive. There are two kinds of perfect shuffles. With an 'out-shuffle' the top card of the deck stays on top. With an 'in-shuffle' the top card moves to the second position of the deck. Magicians know that eight perfect out-shuffles restore the deck to its original order! It looks like the deck has been really mixed up, but it hasn't.

A Perfect Shuffle:
The Computer Science

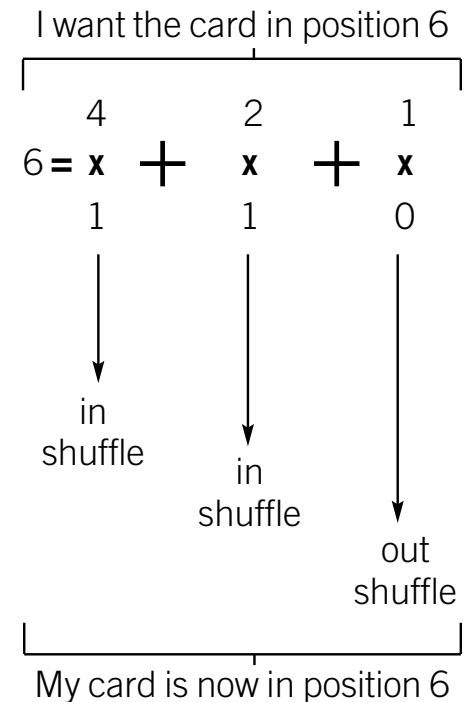## Brent Morris: Magician and Computer Scientist

Computer scientist Brent Morris was fascinated by magic. In particular he became interested in the 'perfect shuffle' in high school and has pursued its mathematics for more than 30 years with some amazing results. He earned his Doctorate in Maths from Duke University, and a Masters in Computer Science from Johns Hopkins University in the United States. He is believed to have the only doctorate in the world in card shuffling. He also holds two US patents on computers designed with shuffles, and has written a book on the subject called *Magic Tricks, Card Shuffling, and Dynamic Computer Memories…* but why so much interest in perfect shuffles?

Binary shifts – as if by magic

You can use perfect shuffles to move the top card to any position in the pack, using a little bit of the maths behind computers: binary numbers. Suppose you want the top card (let's call that position 0) to go to position 6. Write 6 in base 2 (binary), giving 110 (1x4+1x2+0x1). Now read the 0s and 1s from left to right: 1:1:0. Then, working through the 1s and 0s, you perform an out-shuffle for a 0 and an in-shuffle for a 1. In our case that means:

1: an in-shuffle, first

1: another in-shuffle,

0: and finally, an out-shuffle

As if by magic (if you are capable of doing perfect shuffles) the top card will have moved to position 6. Of course it works whatever the number, not just 6. What does this have to do with the design of computers? You can use exactly the same ideas to move data efficiently around computer memory, which is what Brent Morris discovered and patented.

I want the card in position 6

$$6 = \mathbf{x}\begin{matrix}4\\1\end{matrix} + \mathbf{x}\begin{matrix}2\\1\end{matrix} + \mathbf{x}\begin{matrix}1\\0\end{matrix}$$

in shuffle

in shuffle

out shuffle

My card is now in position 6

The remote control brain experiment: the one where you control the cards by thought alone

The magical effect

Get a deck of cards and give them a good shuffle. Spread the cards on the table face down. Now think of the colour RED and select any eight cards, then think of the colour BLACK and select another seven cards at random. Now think of RED again, select another six random cards, then finally BLACK again and select five cards.

Shuffle the cards you chose, and then turn the pile face-up. Take the remaining cards, shuffle them and spread them face down.

Now the remote control starts. Concentrate. You are going to separate the cards you selected (and that are now in your face-up pile) into two piles: a RED pile and a BLACK pile, in the following way.

Go through your face-up cards one at a time. If the next card is RED put it in the RED pile. For each RED card you put in your RED pile think RED and select a random card from the face down cards on the table without looking at it. Put this random card in a pile face down in front of your RED pile.

Similarly if the next card is a BLACK card put it face up on your BLACK pile, think BLACK and select a random face down card. Put this face down card in a pile in front of your BLACK pile. Go through this procedure until you run out of face-up cards.

The experiment so far

You now have the following: a RED pile and in front of that a pile containing the same number of face down cards you selected while thinking RED. You also have a BLACK pile in front of which

is a pile of random cards you selected while thinking BLACK.

Interestingly your thoughts have influenced your choice of random cards! Don't believe me? Look at the pile of random cards you chose and put in front of your RED pile. Count the number of RED cards in this pile. Now look at the random cards in front of your BLACK pile, and count the number of BLACK cards you selected. You selected the same number of RED and BLACK cards totally at random!

One card out and it wouldn't have worked! It's a final proof that your sub-conscious mind can make you choose random cards to balance those numbers! ... Or is it?

Is mind control a reality? Do you now believe in hocus-pocus? Or are you instead looking for an explanation of why it always works?
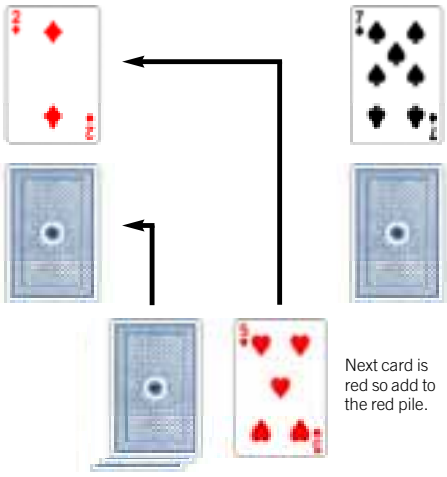
Next card is red so add to the red pile.
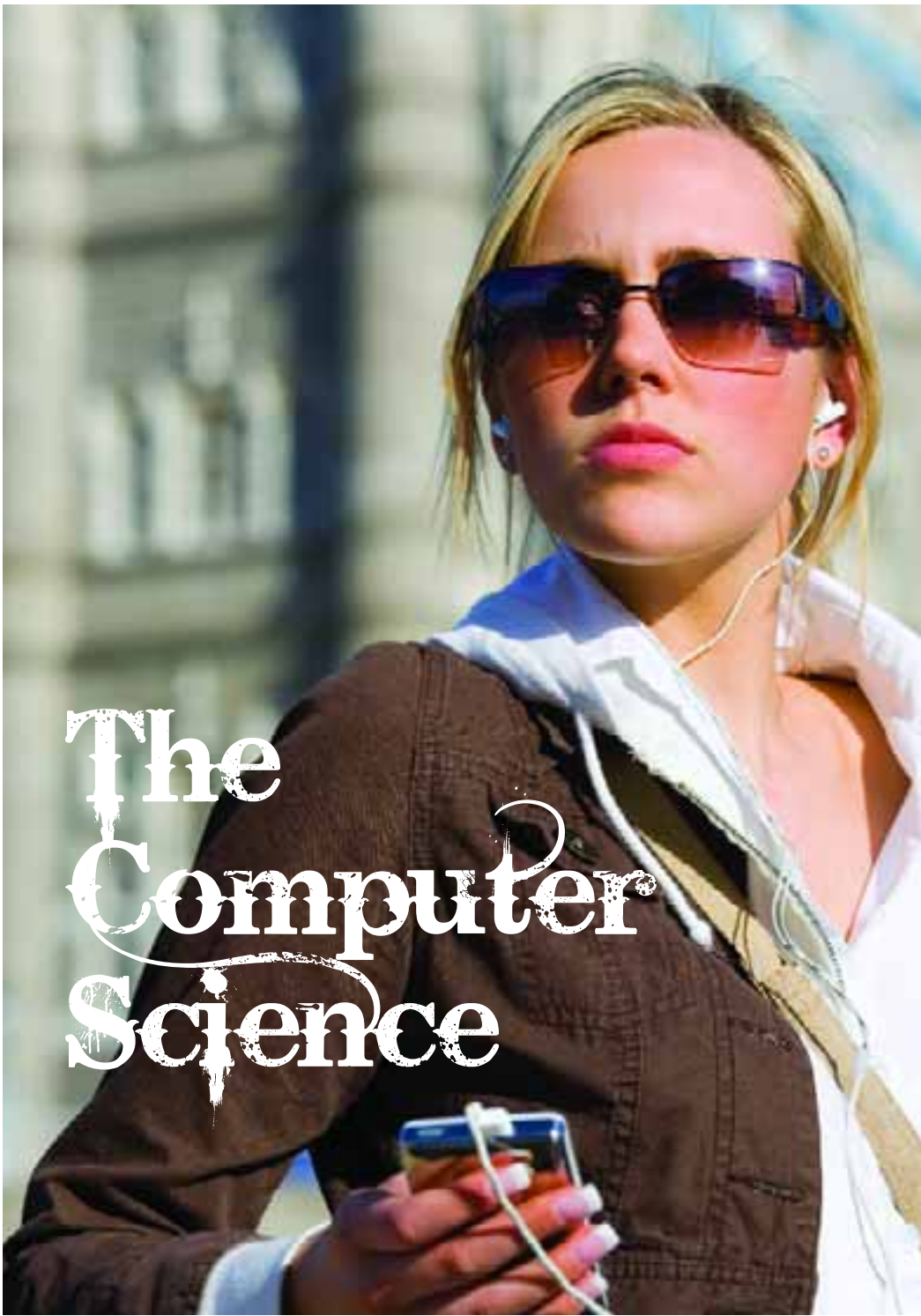
# The remote control brain experiment: The Computer Science

Of course it's not mind control. It's mathematics, but you knew that didn't you? I thought you would. But how does this mind reading miracle work? Well it's all down to Abracadabra algebra. Algebra is an area of Maths that matters a lot to Computer Scientists.

The set up – let's get abstract and do some algebra

Pile 1 (RED)   Pile 2 (BLACK)

R1   **B2**

B3
R3   **B4**
     R4

Pile 3   Pile 4
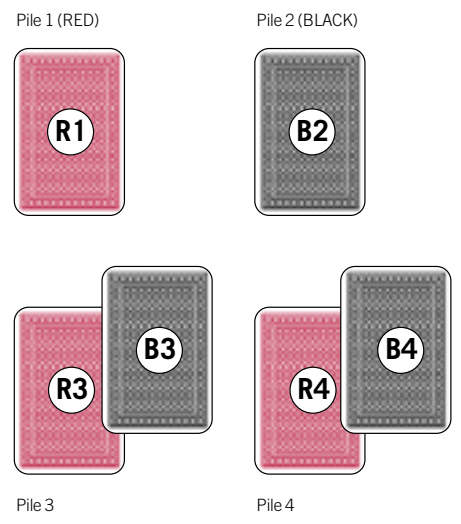
Pile 1 has R1 red cards and nothing else.
Pile 2 has B2 black cards and nothing else.
Pile 3 has R3 red cards and B3 black cards.
Pile 4 has R4 red cards and B4 black cards

Let's call the number of cards in the two piles you dealt R1 for the red pile (pile 1) and B2 for the black pile (pile 2) – see the diagram. The two other piles in front of these contain a random mixture of red and black, so let's say that the pile in front of R1 (pile 3) contains R3 reds and B3 blacks, and the pile in front of B2 (pile 4) contains R4 reds and B4 blacks.

So what do we know?

The first task is to work out what we actually know and turn it into the mathematical equations of the trick.

We actually asked you, in the first part of the experiment, to divide the pack in half. You may have missed that but 8+7+6+5=26.

Now we also know that, for a full pack of 52 cards half (26) are red, and the other half are black so all the red cards add up to 26 and similarly the blacks. We can write that as an equation using the names R1, R3 and R4 for the different sets of red cards and similarly for the black cards. We have to use names because we don't know the actual numbers.

$$R1 + R3 + R4 = 26$$
Call this equation (1)

$$B2 + B3 + B4 = 26$$
Call this equation (2)

The remote control brain experiment: The Computer Science

We also know the number of cards in the RED pile 1 (R1) is the same as the number of face down cards placed in front of it in pile 3 (made up of R3 red cards and B3 black cards) so together R3+B3 must add up to R1. Similar reasoning holds for the cards in front of the BLACK pile (pile 2 with pile 4). So we know two more equations:

R1 = R3 + B3
Call this equation (3)

B2 = R4 + B4
Call this equation (4)

Now we can start combining these equations by swapping things for their equals. For starters, we know R1 is exactly the same as R3+B3 from equation (3) so if we replace R1 in equation (1) by R3+B3 we get the same thing:

(R3 + B3) + R3 + R4 = 26
Call this equation (5)

Similarly if we substitute equation (4) in equation (2) eliminating B2 we get

(R4 + B4) + B3 + B4 = 26
Call this equation (6)

Combining equations (5) and (6) as both add up to 26, we get

(R3 + B3) + R3 + R4 = 26 = (R4 + B4) + B3 + B4

We can simplify this by grouping the same things together

2xR3 + B3 + R4 = R4 + 2xB4 + B3

We can also subtract R4 and B3 from each side leaving the sides still equal (we did the same to both). That leaves:

2 x R3 = 2 x B4

Finally, we can divide both sides by 2, giving:

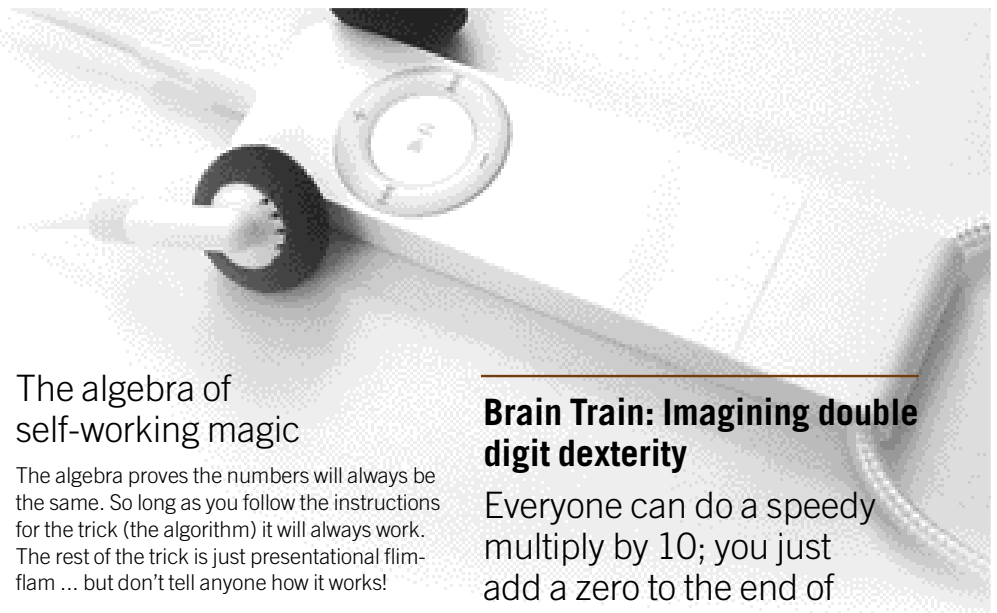**R3 = B4**

Back to reality

Now what did we say R3 and B4 stood for? They are just numbers of cards of particular colours in the face down piles.

The maths shows that the number of RED cards (R3) in pile 3 which is in front of the RED pile is ALWAYS equal to the number of BLACK cards (B4) in pile 4 which is in front of the BLACK pile.

That is how the magic works. Maths.

The algebra of self-working magic

The algebra proves the numbers will always be the same. So long as you follow the instructions for the trick (the algorithm) it will always work. The rest of the trick is just presentational flim-flam ... but don't tell anyone how it works!

Algebra is another way that we can prove computer programs will always do what we want them to, by taking the problem and turning it into an **'abstraction'**. As we have done here abstraction uses general quantities such as R1 rather than the actual number of cards, say 12. The use of various kinds of abstraction in programming languages also helps make it easier to write programs in the first place.

Anyway, using proof, this time algebraic proof, we can be sure that our trick will be self-working without having to try every single set of possible cards, just as we did with the 21-card trick. Remember we need the trick to work 100 per cent of the time if we aren't going to be embarrassed, not 99 per cent of the time.

Now, what if you were talking about, instead of a magic trick, a computer program that was controlling the landing gear on your plane. You would want to be sure that worked 100 per cent of the time as well: that every time the program followed the instructions the right thing happened. Or how about your MP3 player? It is just a computer controlled by programs. It's no good if that only works 99 per cent of the time.

Brain Train: Imagining double digit dexterity

Everyone can do a speedy multiply by 10; you just add a zero to the end of the number. But you can prove your superior mental superpowers by speedy multiplication of a two-digit number by 11. Stretch your imagination and learn how to train your brain's double-digit dexterity by visiting **www.cs4fn.org/mathemagic/** and then challenge your friends.

Would you be happy if every 100th track failed to play? Using similar kinds of abstraction and algebra we can prove programs work correctly too. Mathematical proof is at the core of computer science, and will be increasingly important in the future, helping create safer computer systems, systems you can trust.

The magical effect

You are blindfolded and lean against the wall at the back of the room with your back to the proceedings. Your spirit leaves your body and flies up to the ceiling so you can watch from above.

Meanwhile, your assistant shuffles a pack of cards. Volunteers then select cards and place them at random either face-up or face-down in a 4 by 4 grid. Your assistant adds more to make it even harder. Your spirit now has a target to watch. A further volunteer then chooses any card from the grid and flips it over. No-one speaks. You are still blindfolded. You can only know which one was flipped if your spirit really is floating above, watching.

You are told to return to your body, which you do. A little dazed, you go straight to the cards and point to the one that was flipped over!

The out of-body experience
The one where you float out of your body to watch events

The out-of-body experience: The one where you float out of your body to watch events

The mechanics

This trick is a flamboyant variation of one invented by New Zealand computer scientist, Tim Bell. Have a look at the set-up in the diagrams below. You might just catch the workings of the trick.

The assistant adds an extra row and column of cards but it isn't in fact random. It also isn't actually making things harder, but easier. What they do is look at the number of face-down cards in the row (the number of card backs showing) and if that number is odd, they put the new card face-down. This means that with the added card there is an even number of card backs in the row.

They continue with the next row. If there is an even number of face-down cards, they add a card face-up, so that the new row still has an even number of face-down cards. Of course if the row has an odd number of face-down cards, (ie one or three), they add a new card to make this total even: two or four. Repeat this for all the rows, then do the same for the columns. Add the extra card so that there is an even number of card backs in the column/row. The final card on the bottom right of the last row finishes the set.
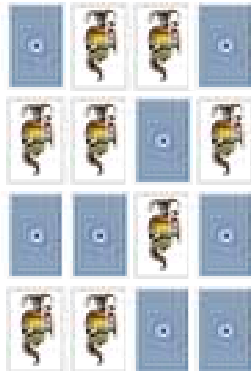
a) 4 by 4 grid of random cards are laid out by spectator
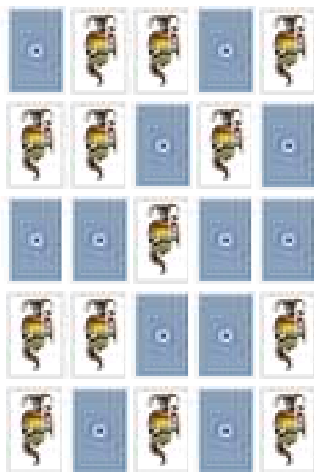
b) You add an extra row and column to make it harder

Detecting the change

Detecting the change doesn't now need any special mystical abilities. You just stand quietly at the back ignoring proceedings.

If one of the cards is turned over without you seeing, it's a simple process to find its location. It's shown in the diagram on the right. Look back at the cards. Start from the top, scanning down row-by-row looking for card backs. Remember you added the extra cards to ensure there was an EVEN number of backs in the row. There will be one row where there are now an ODD number of backs; one of the cards in this row was turned over, but which one?

Start to scan the columns now, again looking for the column where there is an ODD number of card backs showing. When you find it that's the column with the reversed card. So you now have the row position and the column position of the reversed card, and you can reveal this in any super memory sort of way you like.

You could do this with a larger number of cards of course, they just take longer to lay out and longer to scan through to find the changes in the line and columns.

Flipped card in this row

Parity row

Flipped card is in this column

Parity column
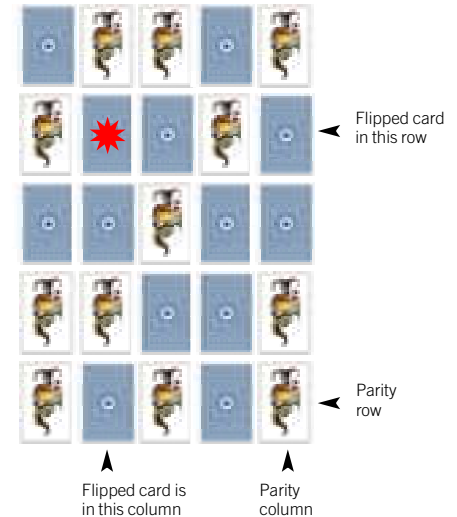
Detecting the flipped card using parity

The showmanship

You can have lots of fun with the presentation of this trick. Get someone to check the blindfold for hidden trapdoors, and so on. Get them then to stand guard over your body. As you return to your body carefully bang against the wall as though you re-entered too quickly. Clearly if you were upside down on the ceiling you will be a bit dizzy when you return so you can wobble about a bit. You will also presumably have trouble working out which way up the square was if you were upside down so pretend to struggle to work it out – turning your head to one side perhaps. The possibilities are endless.

The out-of-body experience: The Computer Science

The Computer Science

Finding mistakes in data – parity

What does this trick have to do with computer science? In the figure the extra row and column you add have a technical name: the **'parity'** row and the parity column. (Parity means equal). Instead of thinking about face-up and face-down cards, think about binary 1 and 0. You can see that your block of cards could just as easily represent a segment of computer data, with the data encoded in 1's and 0's. (These are called **'binary bits'**).

Data sent over a computer network is just a series of 1s and 0s (each called a **'bit'**) packaged into blocks. Trouble is the real world is a 'noisy' place. Signals can be corrupted in all sorts of ways: cosmic rays, radio signals, nearby power lines and the like can all zap bits. It's easy for them to be flipped as they pass over a network. One change can destroy the whole meaning of the message.

The out-of-body experience: The Computer Science

To ensure that, when you send data over a computer network, all the data does make it to the other end without getting scrambled, computer scientists and engineers came up with the idea of adding parity bits to each block of data. It is no different to the way you added the extra cards.

Suppose you want to send a message over a network consisting of the numbers 6, 13, 2 and 12. They can be converted into binary using a special code where each number has its own sequence of 1s and 0s to represent it (see page 17). Our numbers are converted to the four sets of digits: 0110 1101 0010 1100. Rather than send those digits though we add the parity bits to make them five digits long with an extra block at the end for the column parity:

01101 11010 00100 11001 10101

We have used the parity bits to give an even number of zeros here.

Now when the data arrives the receiving computer can see if one of the bits (cards) has an error e.g. it's 1 when it should be 0 or vice versa.

Suppose the computer at the other end actually receives the following message:

01101 10010 00100 11001 10101

By lining the separate groups back into a rectangle, we can see where the parity has been broken in row 2 and column 2 as they both now have three 0s whereas everything else is still even:
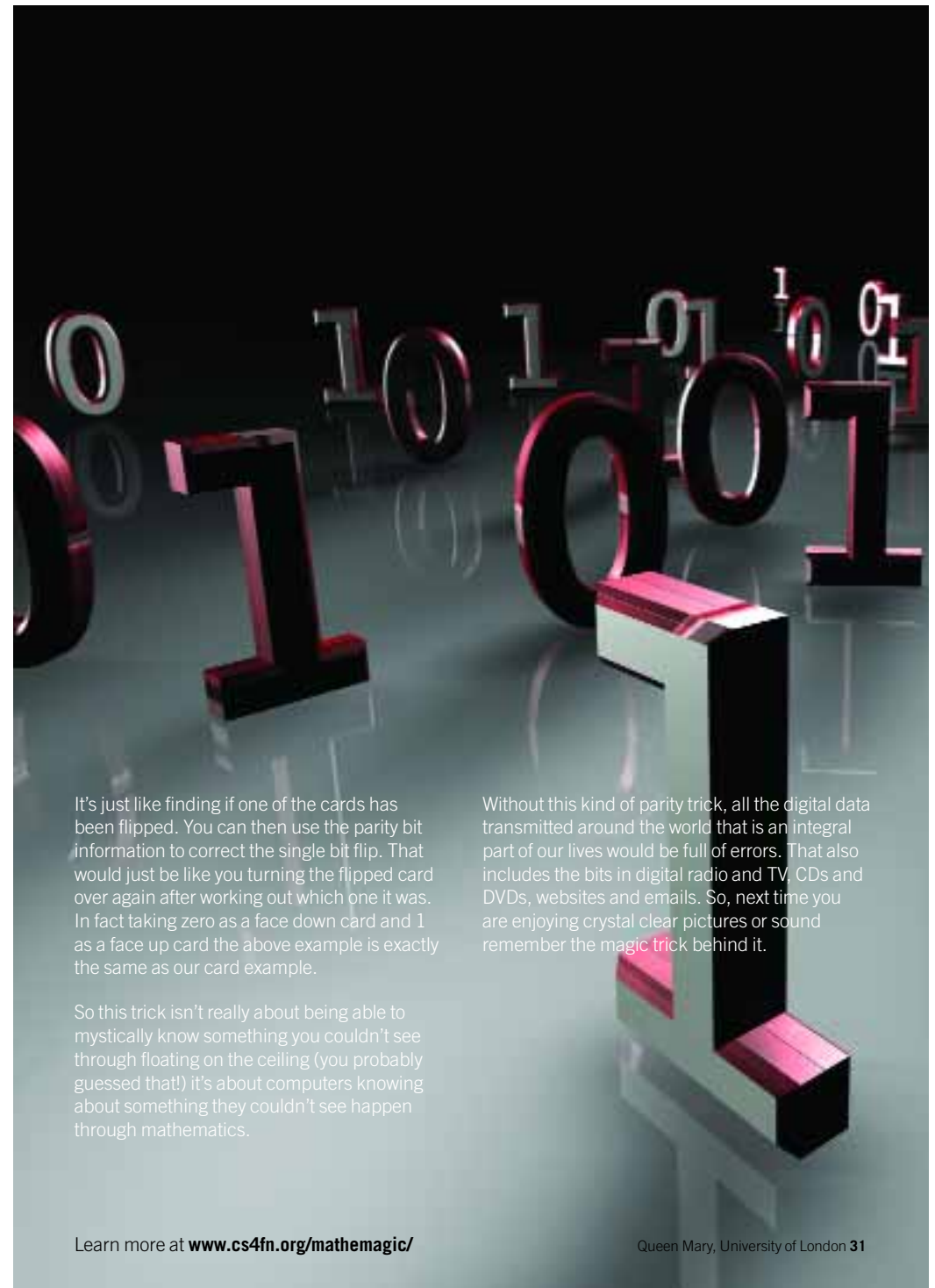
01101

10010

00100

11001

10101

Just a quick one: a way with words

In this experiment we need a random word, a word even you could not have guessed in advance. To start choose any word in the first sentence of the 'showmanship' section on page 47. Count the letters in the word, and use this number to count along the page to a new word. Again count the letters in this new word, and use this number to count along to another new word. Repeat this 'count the letter, move to a new word' until you hit a word in the 2nd paragraph. This is your selected word. Remember you started anywhere you wanted, and chose random words and random numbers, then how could we know your chosen word would be a **'trick'**.

It's just like finding if one of the cards has been flipped. You can then use the parity bit information to correct the single bit flip. That would just be like you turning the flipped card over again after working out which one it was. In fact taking zero as a face down card and 1 as a face up card the above example is exactly the same as our card example.

So this trick isn't really about being able to mystically know something you couldn't see through floating on the ceiling (you probably guessed that!) it's about computers knowing about something they couldn't see happen through mathematics.

Without this kind of parity trick, all the digital data transmitted around the world that is an integral part of our lives would be full of errors. That also includes the bits in digital radio and TV, CDs and DVDs, websites and emails. So, next time you are enjoying crystal clear pictures or sound remember the magic trick behind it.

Carry on Conjuring: The one where you see into the future

The magical effect

You gather up the pile of cards from your last trick (perhaps the 21-card trick page 6) after triumphantly revealing the card the volunteer was thinking of. You now show that you can not only read minds, but also see into the future. First, you write a prediction on a piece of paper and seal it in an envelope so no one sees your prediction. You give it to a member of the audience to hold so that the sealed envelope remains in clear sight and cannot be tampered with.

Next you ask the spectator to cut about half the pack off the top. They decide how much: free choice. They are going to select a card from the top half of the pack that they just cut off but even they aren't going to know which one it will be. They deal the first card face-down on to the remnants of the pack, and the next card face-up

on the table, next card face-down on the remnants of the pack, next face-up on the table, and so on. Once they have finished with the cards in their hands they start again, picking up the face-up pack turning over and dealing the first face-down on the pack remnants and the next face-up, until all cards are dealt. Again they pick up the face-up cards and deal in the same way. They continue doing this until they have exhausted the cards in their hand and there is only one left face-up on the table. You recap for the audience: a free cut of the original pack, a fair deal to eliminate all but one from their original free choice, a sealed prediction written at the start.

Now you reveal your prediction from the envelope...you predicted the card that is now face-up on the table! Magical mind reading...or is it?

Fortune Telling?

Fortune tellers often seem to be able to know all about us. Psychic powers or the clever psychology of the Barnum effect? Read more at **www.cs4fn/mathemagic/**

Carry on Conjuring: The one where you see into the future

The mechanics

All you need to know for this trick is the value of the 16th card in the pack. Write that card's value down as your prediction.

Ask the spectator to cut the pack approximately in half. The important thing here is that they cut off more than 16 but less than 32 cards. If it looks like they haven't, ask them to replace them and cut less (or more). You can claim you don't want to take too long with the experiment (or for it to be too easy).

Now follow the instructions: first card face down on remnants of the pack, next card face up on a separate pile and so on. If you find it hard to remember whether the first card is face-down or face-up, it may help to know that Magicians call this part an 'Australian Shuffle'….because it is a 'Down-under' deal!

The final card left in their hand will be the 16th card from the top of the original deck (guaranteed) so will match your prediction. Remind them of their free cut, the shuffle and take your deserved applause.

A bit about magicians
This trick, of which there are many variations, was invented by Famous Magician Alex Elmsley. He was a graduate of Cambridge University where he studied Mathematics and Physics…before going on to work as a computer programmer.

The showmanship

It's important with this trick to add extra confusion over how easily it could have been a different card. One way to do that, before looking at their chosen card and comparing it with the prediction is to show the last cards discarded, saying "If you had chosen one card further …", reinforcing the idea that it could so easily have gone wrong. This kind of distraction from the true story is very important in making tricks seem mystical.

There are lots of ways you can get to know the 16th card other than just pre-preparing the pack. Here is one that works smoothly into your show if you are doing the 21-card trick too.

As you finish the 21-card trick you have 3 sets of 7 cards on the table. Two of these sets do not contain the spectators chosen card. As you move to revealing the chosen card that is in the middle of the middle set, place the other two piles of 7 cards face-up on the table to make a single discarded pile of 14 cards.

In the final set of cards you know the chosen card will be in the middle of the set (see the 21-card trick instructions). As you discard the other cards in the pile that aren't the chosen one place any two of them on the face-up pile of 14 to make a pile of 16. You now secretly know the value of the 16th card in the face-up pile (THE TOP ONE) - remember it!

Your prediction is…
Why not place the 8 of Hearts in the 16th position? Then get the person to look inside the cover of this book to see your prediction. It really is a magic book!

Once the 21-card trick is done, turn over and put this stack of 16 cards on top of the pack, putting the rest underneath, so you now secretly know the value of the 16th card from the top.

Your prediction is…

With the face-down pack in front of the spectator (you now do know the value of card 16 from the top), write this card's value as your prediction…and off you go.

Binary magic (sort of)

Why does it work? Well it's based on binary arithmetic and an algorithm for searching. The deals progressively eliminate every second card, and leave half of those that were previously left. Another way of saying that is that the remaining cards are related by 2n where n is the deal you are doing (n is 1 for the first deal, 2 for the second and so on). Let's look at this sieving process in detail.

Let's refer to the original set cut from the top of the pack with numbers from 1 upwards. You don't know how many cards there are but you have fixed this number of cards to be less than 32. Remember you know the value of card 16. It's your prediction. The first face down is discarded, the next kept, and so on. That means this first 'fair' deal actually just eliminates the odd position cards. Your volunteer is left with even cards in the face up pile.

Another way of saying that is they have cards from original positions, 2n for each n<16 (as you fixed it that 2n is less than 32 and 32=2x16).

Take this remaining set, turn them over and do the same deal again. Again you will remove every second card. You are left with cards from original positions 4n where n<8 now (as 32=4x8). The same deal again removes every second card

leaving cards from original positions 8n where n<4 (as 32=8x4). Finally, dealing again removes every second card leaving just the 16th. The deal selects cards with values 16n where n<2. Since n<2 means n must be 1, that shows we are left with card 16 alone ... and card 16 is your prediction.

Sort of important

Binary numbers are fundamental to computer science, in part because computers use binary to represent data. There are some more interesting links too. The ways that computers solve problems quickly are often based on binary properties. For example one of the most efficient ways to search for data works by discarding half of the data each time, always keeping the half where the searched-for thing resides, just as in our trick. It is called **'Binary search'**. An algorithm called **'Radix Sorting'** also works in a similar way to the trick to sort data into order: it was used on early computer punch card machines to sort punch cards. A variation of Radix Sorting was also used by early computers to pull out a particular punch card from a mixed up pack of cards – just as we did for card 16. Searching, in one form or another, is one of the main uses now of computers. Search engines, for example, using incredibly efficient search algorithms allow you to search the whole of the web in seconds. Abracadabra!

Each deal removes every second card

Start	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	27	28	29	30	31
Deal 1		2		4		6		8		10		12		14		16		18		20		22		24		26		28		30	
Deal 2				4				8				12				16				20				24				28			
Deal 3								8								16								24							
Deal 4																16															

Learn more at **www.cs4fn.org/mathemagic/**
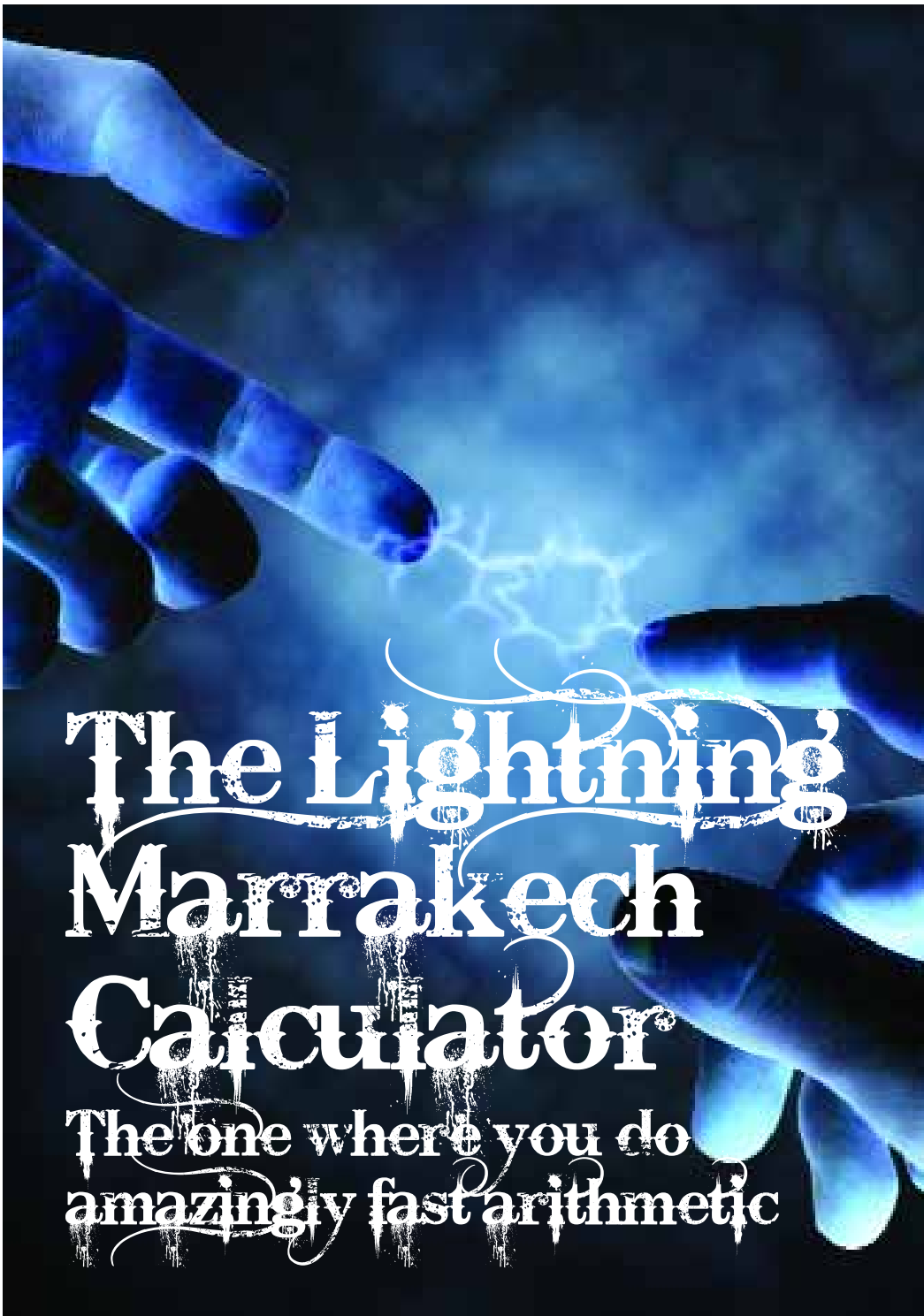
The Lightning Marrakech Calculator: The one where you do amazingly fast arithmetic

The magical effect

A volunteer takes you up on a challenge involving feats of lightening arithmetic: playing the game of 'Marrakech' (named after the ancient, magical Moroccan City of Gold) against the clock. Stunning your audience, in game after game you come out on top, apparently hardly needing to think at all to do the maths. Can you really do arithmetic that quickly in your head?

Both you and volunteer are given a clipboard, pen and paper to do your calculations. Cards 1 to 9 from one suit of a pack of cards are laid out face up on the table. Each player takes it in turn to select a card and place it in front of them. The aim is to hold any three cards that add up to 15 before the other player does. Play several games, taking it in turns to go first.

To win you need to be good at addition to be able to not only work out which cards you need to make 15 but which ones your opponent needs too. Don't you?

The mechanics

Whilst your opponent may be doing furious additions on their clip-board, you don't do any at all. Instead, draw out a Noughts and Crosses board and write the numbers into the squares in the following way.

4 9 2

3 5 7

8 1 6

Make sure you keep this hidden! Otherwise you will give the game away.

When you make a move, draw an X over the number chosen. When your opponent makes a move place a zero over their number. What is the secret? It is easy to spot numbers that add up to 15 because that is what all the rows, columns and diagonals do in the square. It is of course a 'Magic Square'. You aren't doing any arithmetic at all – just playing Noughts and Crosses. As long as you can play that well you will never lose. (Go to the cs4fn website www.cs4fn.org for instructions on how to play perfect Noughts and crosses – the secret is to go for the corners and fork your opponent!)

Why is this game named after Marrakech? Did you know that the main square in Marrakech is also called the Magic Square?

The showmanship

Rather than doing this with cards you can do it just with numbers written up on a white-board that you cross out – or pieces of card with bigger numbers on (it doesn't have to be 1-9). Just take the basic Magic Square and add the same large number to each of the numbers in the grid. Remember though: adding some number N to all the numbers means the 'target' is now $15+3N$. The bigger the numbers used the more amazing it will seem. Of course as you don't do any arithmetic it won't make any difference as long as you can write them into the Noughts and Crosses grid in the right order to make the magic square.

Learn more at **www.cs4fn.org/mathemagic/**
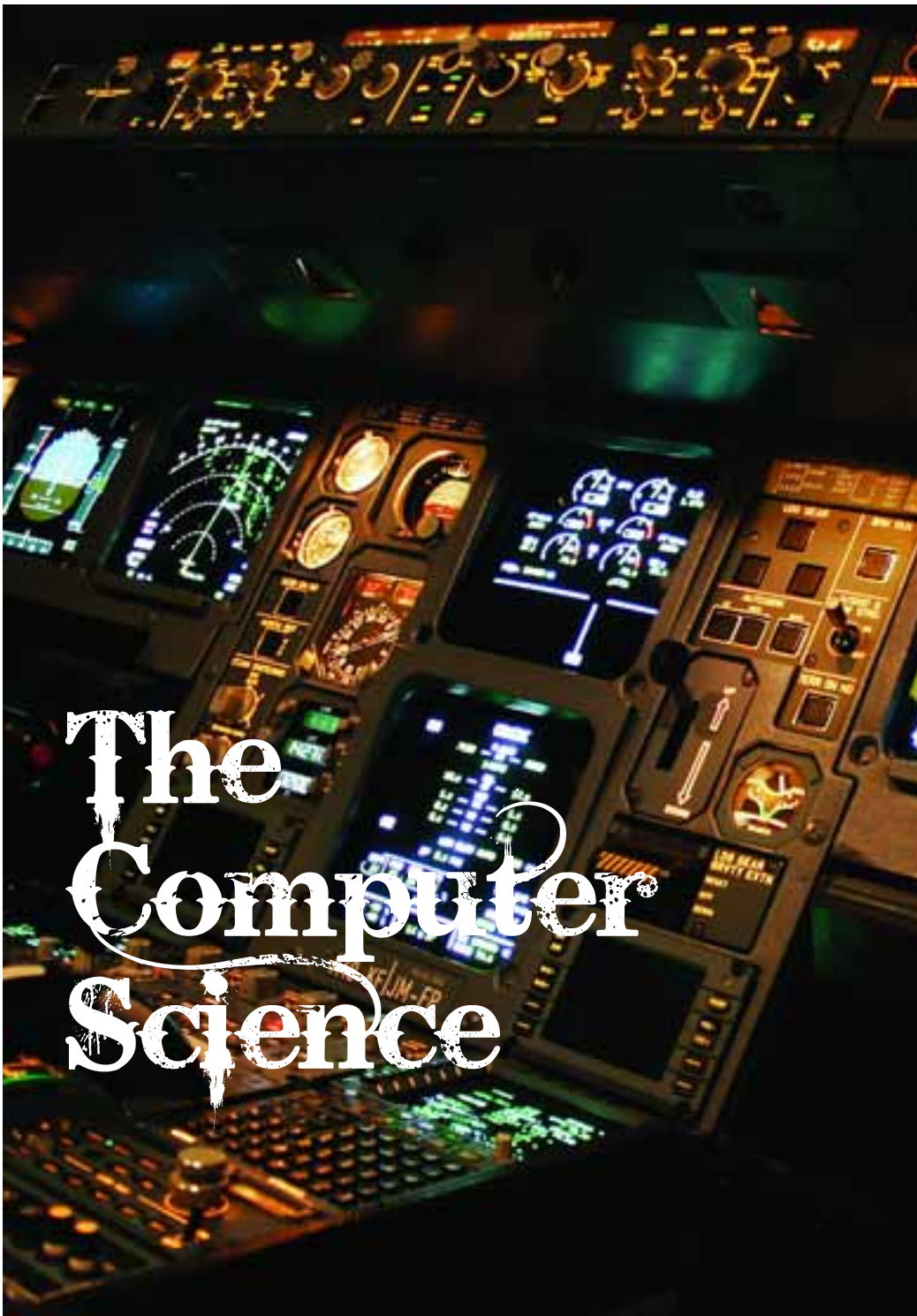
The Lightning Marrakech Calculator: The Computer Science

Here's one I prepared earlier

Noughts and Crosses and Marrakech are games that a mathematician would call **'isomorphic'**. All that means is that behind the presentation they are really exactly the same game. If you have a perfect strategy for playing one (say Noughts and Crosses) then you can also use it as a perfect strategy for playing the other game (Marrakech) too. All you do is translate from one to the other as we were doing in the trick.

Computer scientists are really interested in situations like that. A lot of the subject is about solving problems so you can then produce algorithms (programs) that a computer can follow. Now if you can show two problems are the same then you can solve the second one in the same way as you solved the first. You don't have to start from scratch – just pull the ready-made solution out of the hat.

For example, suppose you have worked out the perfect strategy for playing Noughts and Crosses, and written a program to do it, you can use the same algorithm and so much of the same program to play Marrakech too. Essentially, all you have to reprogram is the interface that presents numbers instead of Os and Xs and some code to translate from one problem to the other.

It's not just in games that you can play that trick. It works in lots of problem areas including some that are known to be incredibly hard to solve well. A classic example is called the **'Travelling Salesperson'** problem. It's to do with plotting a fast route visiting each of a series of cities only once. It turns out if you could come up with a perfect solution to it then you would also have a solution to lots of apparently completely different problems. Trouble is no-one has come up with a perfect solution! Fairly good ways to do it (known as **'heuristics'**) have been invented that also work across all the problems though.

The Computer Science

The Lightning Marrakech Calculator: The Computer Science

User Interfaces

We've glossed over something important though. Why did switching to Noughts and Crosses make the game easier anyway?

The difference is just that the information has been presented in a way that makes it easier for our brains to process. Our brains are very good at seeing visual patterns – we do it with very little effort as we evolved to be visual creatures. We naturally spot visual patterns without much thought. That contrasts with doing arithmetic that is a learned skill, or even the alternative of remembering (or searching through lists if we wrote them all down) the triples of numbers that add up to 15.

That is another lesson for computer scientists to remember – if the idea of computers is to make things easier for people to do (it usually is) then the user interfaces we write should take into account the things we are good and bad at. If a program can take advantage of the things we are good at in the way it presents information then it will be much more effective.

For example, we are much better at recognizing things when we see them than remembering them from scratch. That is a reason why Graphical User Interfaces are an improvement on the old systems where you typed in commands.

It is much harder to remember the word 'find', say, to type in when you want the computer to find a document (are you sure it's not 'search' or 'retrieve', or maybe 'fetch'…) than to recognize it as the correct choice in a drop-down menu.

So if you want to be an expert problem-solver remember the Marrakech trick – look for a solution you already have to a different problem that is really the same, then make the interface work for the user. It may just make life a lot easier for you and your users alike.


Just a quick one: Time travel

You are going to travel in time and predict the future. You need to create a secret 'random' target year, so get a friend to secretly write down any three-digit number. To make it 'harder' you say the digits must all be different and the biggest digit must be at the front. You say it's still too easy. Now you want them to jumble things up a bit, get them to reverse their selected number and write it underneath and, finally to make it even harder still have them subtract the lower number from the higher. You now have a random year even your friend couldn't have predicted, but you can!

You concentrate. The future is getting a bit clearer…the answer has a '9' in the middle. Yes, you're right, but there is more to come. As you 'tune your mind' more, get them to challenge you further and reverse the digits in this answer then add the two numbers. Even before they have finished the addition your mind has been able to 'jump the time curves' and predict the total is now the year written into the crystal ball on page 59.

Learn more at **www.cs4fn.org/mathemagic/**
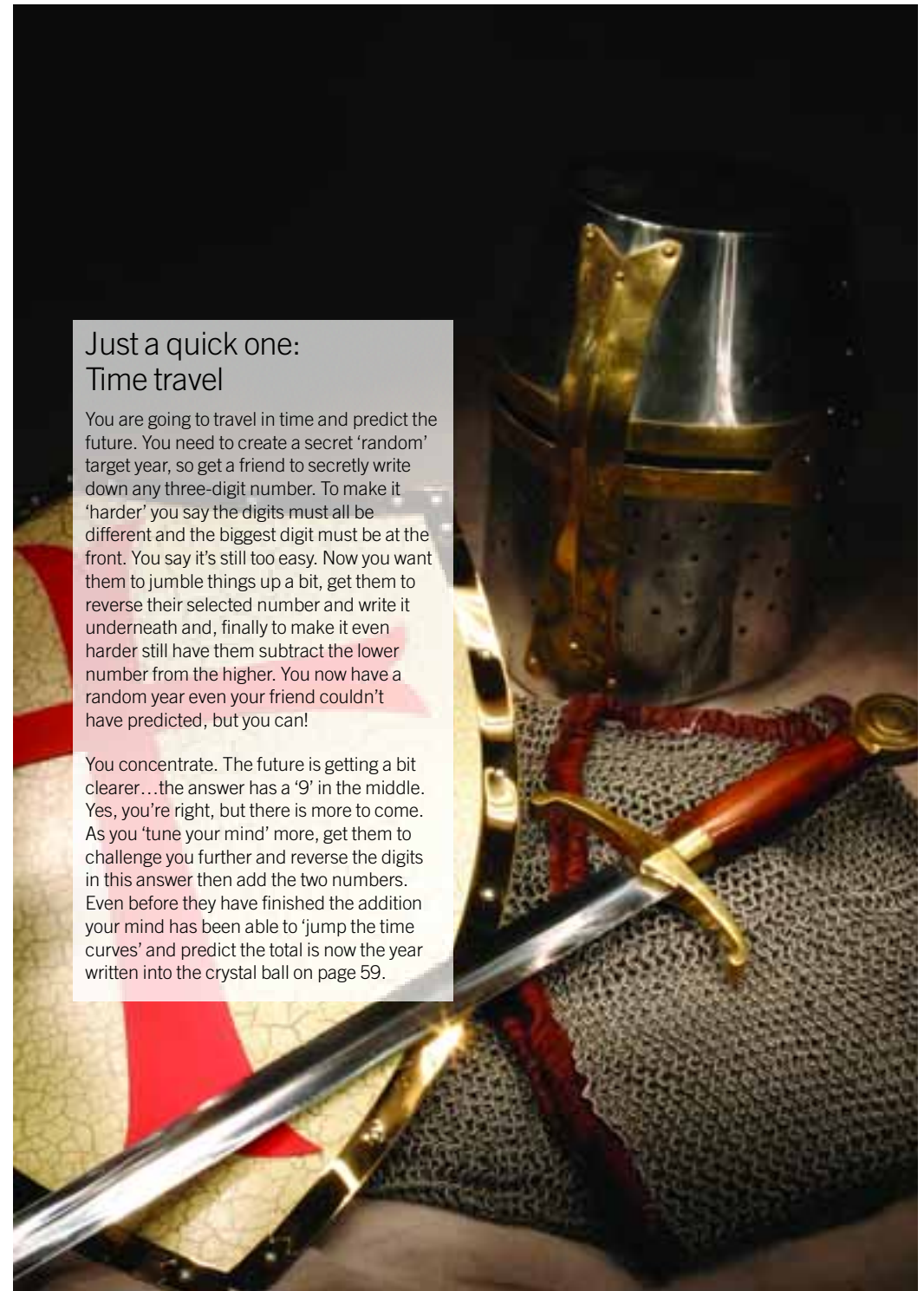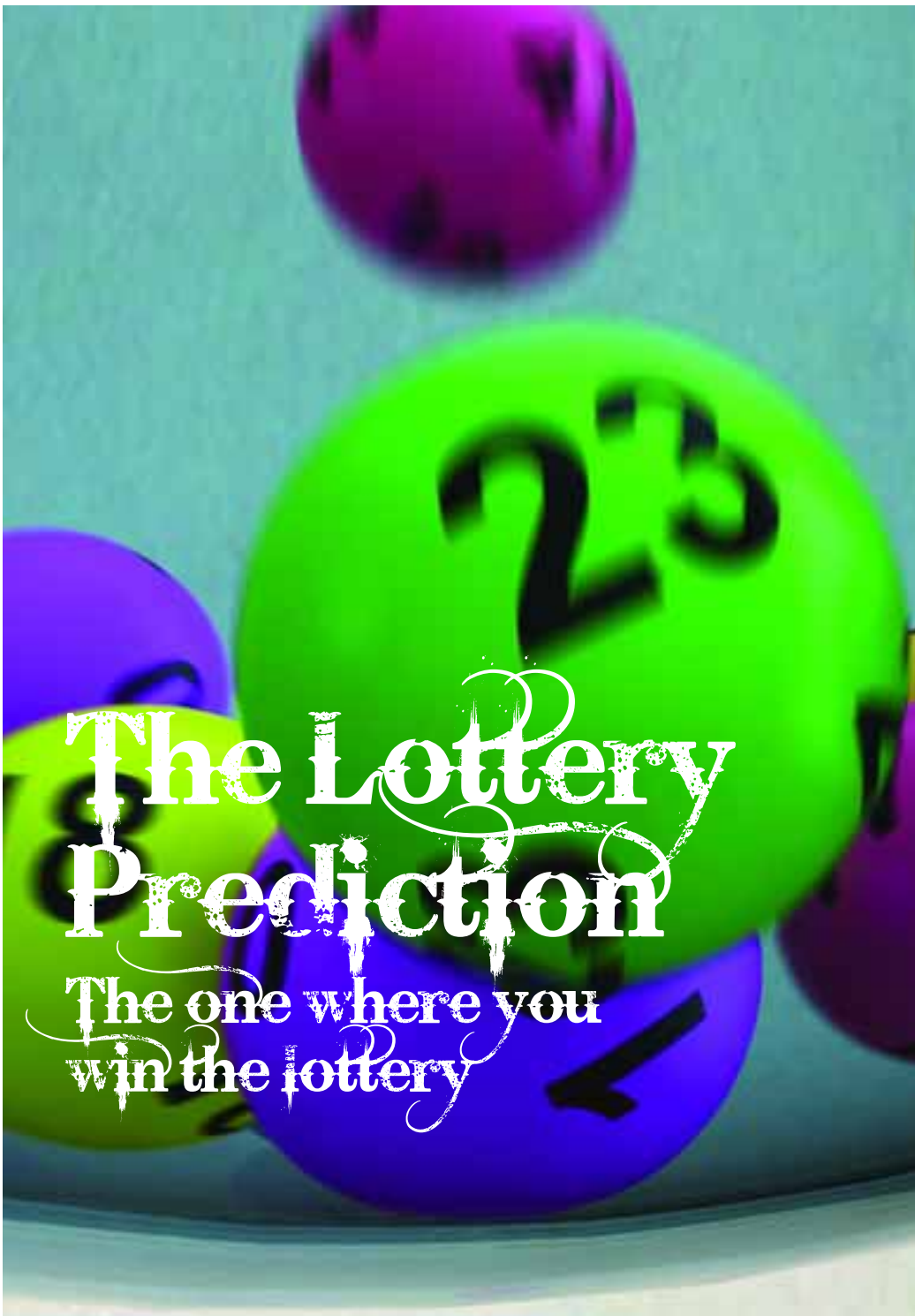
The Lottery Prediction:
The one where you win the lottery

The magical effect

Announce that you are going to get a volunteer to randomly pick a number to use as a lottery number. Every one writes down their chosen lucky 4-digit number.

Cards numbered from 1 (Ace) to 9 are then passed around the audience at random. A set of three numbers are chosen randomly by a volunteer by choosing people holding cards in turn. The person choosing doesn't know what the card chosen will be. Three are picked at a time to give a series of three digit numbers. These numbers are then added up to give a single four-digit number. That is the winning lottery number.

Find out if anyone in the room has the winning number (anyone who does gets a small prize). You then point out that you do not do the lottery. It would be unfair because you can see into the future. Get the volunteer to open the envelope they have been writing on from the start. Sealed inside is your lottery number…and amazingly it is the winning number – 1665!

The mechanics

The numbers chosen aren't completely random. The way they are chosen means they always add up to 1665. Here is how you do it. First you need to choose the right set of nine cards from three different suits as follows. Take the 3, 4 and 8 of diamonds, the Ace, 5 and 9 of Hearts and finally the 2, 6 and 7 of Spades. Notice that you have numbers 1-9 but in three suits. More to the point in each suit the numbers add up to 15.

Shuffle these 9 cards and pass them out into the audience so no one knows who has what.

Now get your volunteer to choose an order of the suits – say Hearts, Spades, Diamonds (It's up to them). Give them a clipboard with paper on it to write the order down so they don't forget it. In fact the 'paper' could be the envelope containing your prediction of 1665 prepared earlier. That way they will eventually discover that without realizing it they have guarded your winning lottery ticket all along!

Now, suppose they chose Hearts to be first. Get the three people holding Hearts to stand up and have the volunteer pick one at random. That is

A Bit about Magicians
Penn Jillette, half of the unconventional magical duo Penn & Teller, has a passion for computer technology and the web. He was a regular contributor to a Computing magazine in the early 1990s and wrote web articles for a search engine company.

The Lottery Prediction: The one where you win the lottery

the first digit of the first number. Write it up on the board for all to see. Now do the same with their second choice of suit. If it was Spades, then the three people holding Spades stand up and one is chosen. That is the second digit, written next to the first. Finally do the same with the last suit to get the final digit to make a three-digit number.

For example you may have ended up with numbers, 573.

Books we loved
Hiding the elephant by Jim Steinmeyer is a great book about the History of Magic. Find out how the same science and engineering keeps reappearing in different tricks.

Repeat this again using the same order of suits with the remaining cards to get a second 3-digit number written in columns under the first. For example, maybe the numbers this time were, 924. You have written:

5 7 3

9 2 4

Finally, do it again to get the final number: perhaps 1 6 8. You have written up in columns:

5 7 3

9 2 4

1 6 8

Remind everyone that the order of suits was a free choice and the order of numbers was free so we could have ended up with any numbers. What you don't say of course is that by keeping the order of suits the same each time you made sure, whatever the numbers were, the columns add up to 15 in the next step!

Now your volunteer adds up the three numbers to get the final number. They will get 1665.

Why does it work? You kept the suits in the same order. That means, because each column is just one of the suits, you ensured in the final addition the three numbers in each column added up to 15. The 6s creep in to the final answer because of the carries!
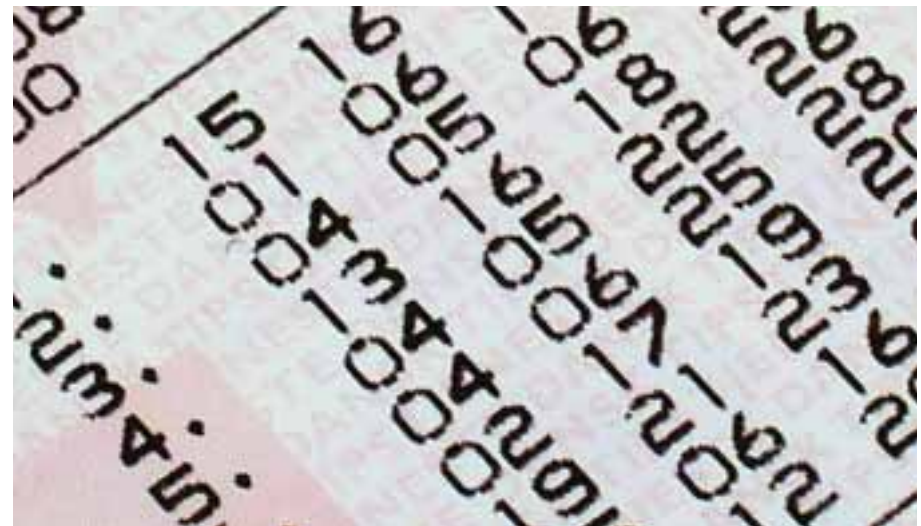
Since the order you add things makes no difference to the total, it doesn't matter which order the cards of a suit were chosen.

It wouldn't work with any set of numbers of course. We chose the numbers carefully. In fact they just come from the columns of a magic square – the same magic square we saw earlier in fact in the Marrakech game!

The showmanship

Rather than do this with cards you can write the numbers on pieces of paper with three different colours. Screw them up into balls one at a time and toss them into the audience. Then get the audience to toss them around the room randomly as though they were in the lottery ball machine, before getting to choose them using the colours in place of suits. Make sure the numbers of each colour add up to 15 of course!

If you are doing this trick in the same show as the others based on Magic Squares, then use a different total (a different Magic Square) so they don't spot the number 15 from before and start to think too hard about it. You will need to work out the new magic number to predict of course.

The Lottery Prediction: The Computer Science

Reuse it!

The first lesson here is one about reuse – we have actually taken the Magic square properties of numbers adding to 15 again and used it in a different way. Notice this isn't an isomorphism though (See The Lightning Marrakech Calculator, page 38) – it isn't the same trick just covered in different presentational flim-flam. We've taken a particular organization of our data (the cards) and found a new way to use the same property of the magic square.

Just a quick one: Street magic

Street magicians like David Blaine often use the following psychological trick. Ask a friend to quickly think of a two-digit number between 1 and 100, both digits odd and both digits different from each other. Concentrate, the answer is 37. Find out more at **www.cs4fn.org/mathemagic/**

Leave it alone!

Something we haven't come across so far is an important kind of property called an 'invariant'. Something is an invariant of an algorithm if it stays true even as the algorithm's instructions are carried out.

Think of it a bit like a paper chain cut from a newspaper – each copy is the same as the last so they don't change but they still make their way from one end of the table to the other. The last has made it to a different position from the first.

Invariants are useful in understanding why an algorithm works – and in proving that it does actually work.

Invariants are useful in understanding why an algorithm works – and in proving that it does actually work. That's because it turns out, in a weird sort of way, that understanding what property stays the same is the key to understanding how a computation changes things. It gives a way of writing a short argument of why even an enormously long computation works …provided the computation is repetitive in some way.

The Lottery Prediction: The Computer Science

Our trick is quite simple but it illustrates the way invariants are used. In the trick the total of the numbers in each suit is invariant – it is always 15. Taking into account how the cards are used and the columns of numbers constructed, another way to say this is:

The total of the numbers written in each column so far, added to the total of the numbers on the cards still to be picked of that suit, is 15.

We can write this to look more mathematical as an equation:

column + unpicked = 15
(We will call this equation, I, for Invariant)

We are just using column as an abbreviation for "The total of the numbers written in a column so far" and unpicked as an abbreviation for "The total of the numbers on the cards still to be picked of that suit".

Now consider the start of the trick. Nothing is written down in the columns. That means:

column = 0

On the other hand all of the cards have been passed into the audience but none picked yet, so we also know for each suit:

unpicked = 15

Our invariant, I, holds as

0+15 = 15

We have shown that our invariant does at least hold at the start.

The trick then basically consists of doing the same thing over and over: we pick a card from the audience and write the number on it in the corresponding column. Suppose that card has some value x (it doesn't matter what x is), then the new value of unpicked is unpicked – x. The value of column also changes though to unpicked + x. Our invariant property becomes:

(column – x) + (unpicked + x) = 15

This simplifies, canceling out the subtraction then addition of x, back to the original property.

column + unpicked = 15

That shows it really is an invariant. Even though we've moved things around and taken some cards out of the game, I still holds true. We've moved closer to the goal but stayed the same!

What have we shown so far? We have shown that at the start of the trick the invariant holds and also that whenever it holds, it continues to hold after the next step of the trick has completed. That means it will still hold, step after step, all the way to the end of the trick.

What do we know about the end of the trick? We stop when there are no more cards left to pick from the audience. That means when we finish,

unpicked = 0

holds for each suit. If we put that into our invariant property we get that

column + 0 = 15

In other words we also know that at the end of the trick

column = 15

Turning that back into English, it just means the sum of each of the columns add up to 15 at the end. That of course also means that the total will always be 1665.

Notice we proved this holds even without stepping through all the steps of the trick. We could generalize the argument too. Suppose there were 100 different 'suits' instead of just 3, with 100 different cards in each suit, each set chosen to add up to some number, total. Our proof above would still hold just with a different final total and with invariant

column + unpicked = total

Even though there are vastly many steps to follow instead of 9, and so the trick is far longer, the proof is just the same length.

A Bit about Magicians
Professional magician 'Fitch the Magician', or to give him his proper title, Dr William Fitch Cheney, Jr. earned the first mathematics PhD ever awarded by the prestigious American University, the Massachusetts Institute of Technology (MIT) in 1927.

As with the proofs we've seen earlier a similar approach can apply to programs. This use of invariants gives a way to reason about any program that repeats the same steps over and over to achieve some final result…that is, most programs.

The style of argument we have just given for our trick is based on a ground-breaking way of proving properties of programs called **'Hoare logic'**. It is a special kind of mathematical logic that makes the steps needed to complete a proof precise…which can then be used as the basis for computers to prove that new programs are correct. The logic is named after Professor Sir Tony Hoare who was made a Knight of the Realm for his contributions to computer science, including Hoare logic.
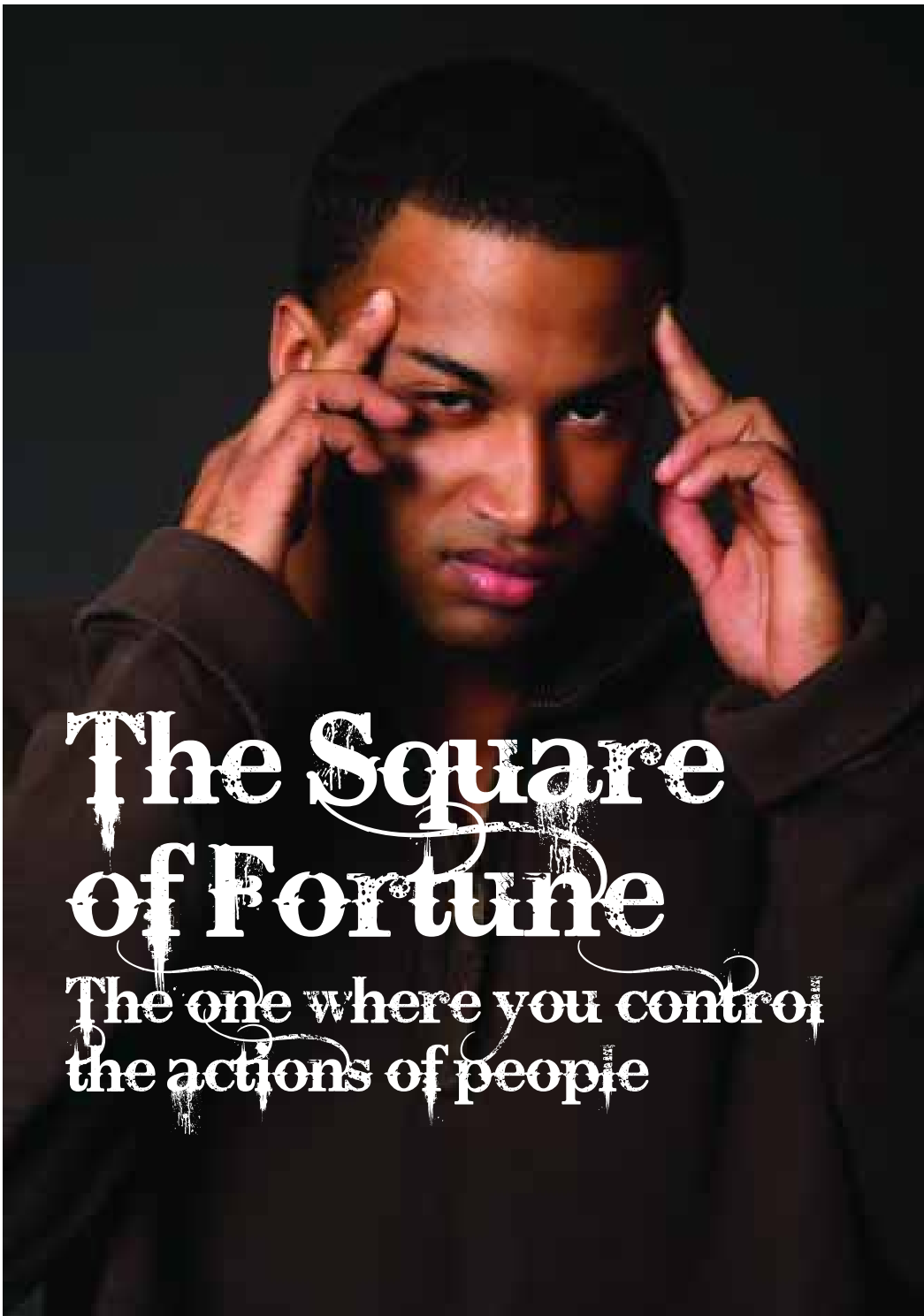
The Square of Fortune:
The one where you control the actions of people

The magical effect

You set out a square of cards and invite a series of people to come forward and choose a card. They take that card and remove the cards in the row and column it is in. Subsequent people do the same until all the cards are chosen or removed.

You add up the numbers on the cards chosen and miraculously you have controlled the choices so that the number is the prediction you sealed in an envelope at the start!
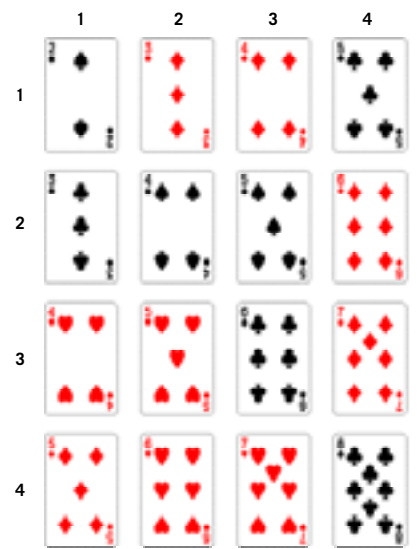
The mechanics

This just works! As long as the grid uses the cards shown here you will always get the answer 20.

So how does it work? Think about a grid like the one below with what we will call 'seed' numbers round the edges: 1 to 4 along the top and down the side.
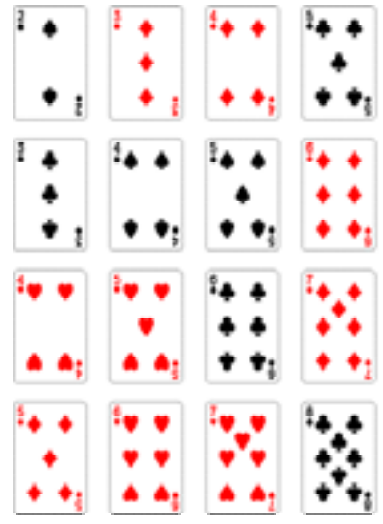
Laying out the square of fortune

The square of fortune with row and column seeds

The Square of Fortune

The number in each square is actually the sum of the seed number at the top of the column and the seed number for the row. For example, take the last entry, 8. Its column seed is 4 and its row seed is 4 too…and 4+4 = 8. All the numbers in the grid work like that – just the addition of those particular seed numbers.

Now think about what happens when the choices are made. When you pick a number you cross out everything in the same row and column. That means that every row will have just one number chosen from it. The same goes for the columns. When we add all the numbers together though, what do we get? The numbers in each position add in the seed for its row and the seed for its column. The 4 numbers chosen add in, between them, all the row seeds and all the column seeds and nothing else. What do we get when we add all the seeds?

Row seeds:	1+2+3+4	= 10
Column seeds:	1+2+3+4	= 10
Total:		= 20

The showmanship

You can use other numbers too if you want it to look more mysterious (see below for how to work a suitable set out). You can also do it as a "Bingo card" with a bigger (more impressive) 5 by 5 grid, say, of the numbers 1 to 25. The numbers 1 to 25 work really well: their ordinariness makes it clear there are actually no tricks in the choice of numbers! The seeds then are 1 to 5 and 0, 5, 10, 15 and 20 and the number 'forced' is 65 as that is what they add up to. Using numbers 1-25 also looks like a calendar, so you could base your presentation on that idea.

Grow your own

If you don't want to use the 1-25 grid, or want to force a particular number or a particular page in a book you can grow your own grid.

How do you construct a custom set of numbers? One way is to do the above backwards. First take down the number you want to force and break it down into a set of seed numbers. Start with a set of row seeds. Take them and spread them across the rows leaving that number in each position. Next on another grid take the column seeds and spread them down. Now add the two grids (i.e., add the values in the corresponding positions) to get your custom Bingo card.

	1	2	3	4	5
0	0	0	0	0	0
5	5	5	5	5	5
10	10	10	10	10	10
15	15	15	15	15	15
20	20	20	20	20	20

Spread the row seeds across the rows

+

	1	2	3	4	5
0	1	2	3	4	5
5	1	2	3	4	5
10	1	2	3	4	5
15	1	2	3	4	5
20	1	2	3	4	5

Spread the column seeds down the columns

=

	1	2	3	4	5
0	0+1	0+2	0+3	0+4	0+5
5	5+1	5+2	5+3	5+4	5+5
10	10+1	10+2	10+3	10+4	10+5
15	15+1	15+2	15+3	15+4	15+5
20	20+1	20+2	20+3	20+4	20+5

Add the two spread grids to get the final grid

=

	1	2	3	4	5
0	1	2	3	4	5
5	6	7	8	9	10
10	11	12	13	14	15
15	16	17	18	19	20
20	21	22	23	24	25

Creating a 'magic' Bingo card with the seeds 1 to 5 and 0,5,10,15,20. It gives a card with just the numbers 1 to 25. Different seeds give different grids and force different numbers.

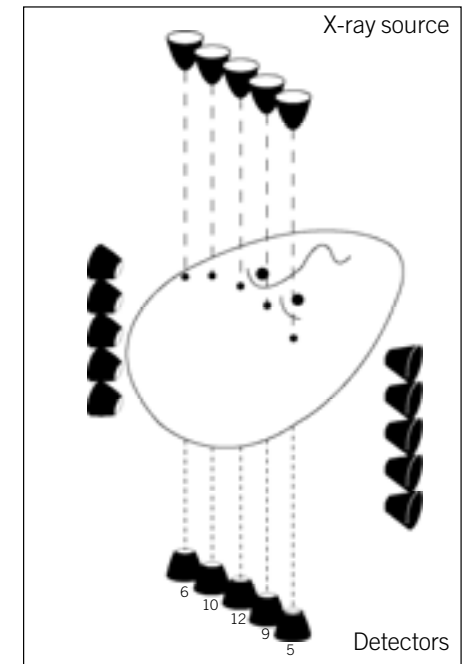# The square of fortune: The Computer Science

The link from this trick is actually to an amazing technology that we are starting to take for granted: computer tomography. Tomography is a kind of medical scanning that allows doctors to create a picture of a two dimensional (2D) slice through your insides. The pictures of the slices can then be put together to make a 3-dimensional (3D) picture. Tomography is used to help build up 3D brain scans, for example. It's a little like taking normal X-rays, but lots of them and from different directions.

The X-rays pass from one side of your brain and are measured by a line of detectors on the other side, so in effect you have a 1D (line) image of your brain at a particular angle.

This information alone doesn't give a 2D version though, just a series of 1D images. Worse than that each image is more like a shadow of what is there. The rays used passed all the way through the head but are blocked to a greater or lesser extent by the bone and brain stuff in the way. That makes the 1D image darker or lighter. The image you have has echoes of everything on the path the ray passed through, not just of one point somewhere in the middle.

A series of slices of a brain from a tomography scan

Tomography takes X rays at different points around the head getting images, very much like our seed numbers

The square of fortune: The Computer Science

The slice is obtained from the 1D pictures by a computational process called **'back projection'.** It's rather similar to the way we created our bingo grid.
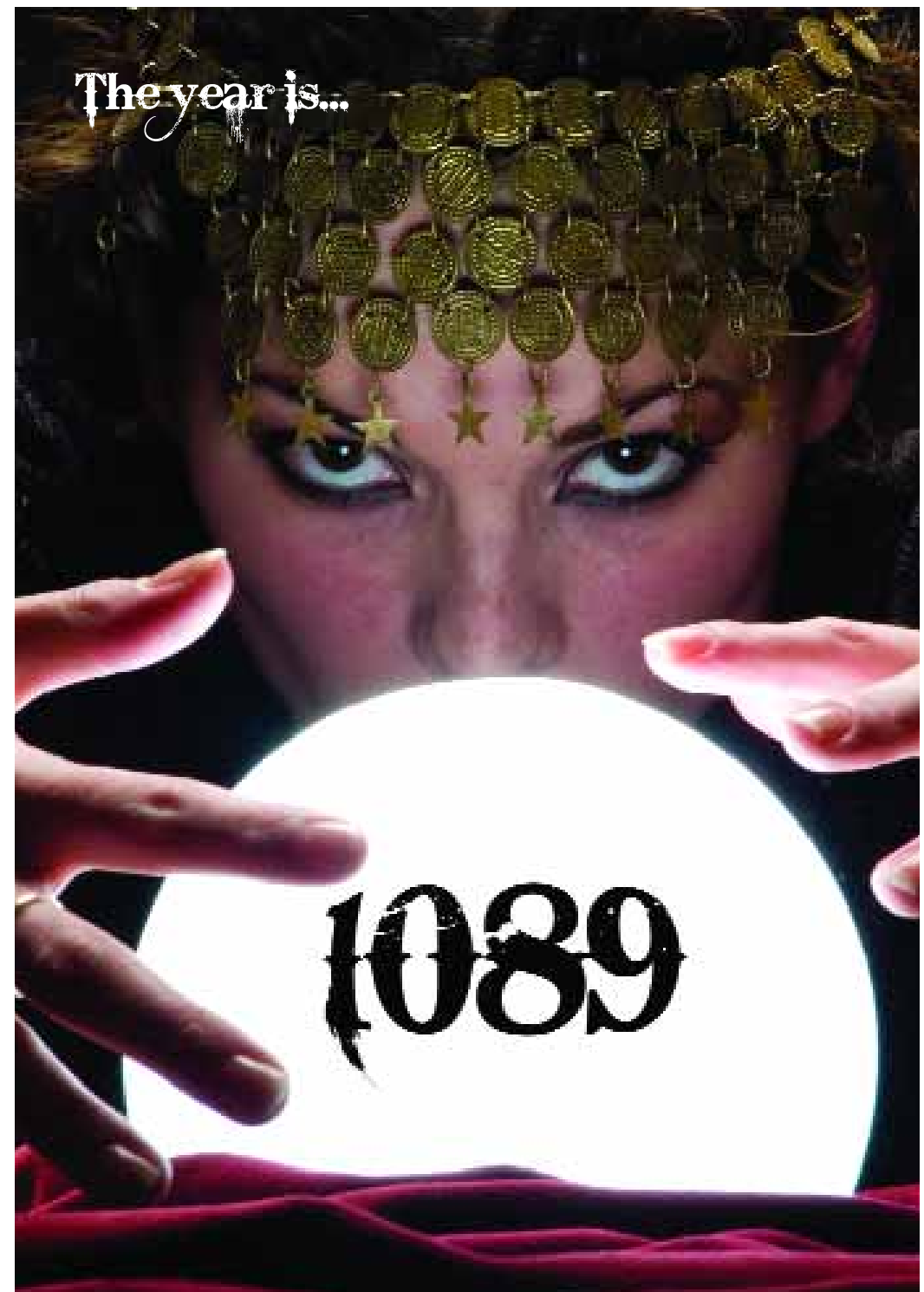
Think of creating the Bingo card as combining two of these 1D scans from 2 directions. Each measurement in a scan is from a ray passing, say, through your head, giving a number for the amount of stuff found along the way. Suppose we take 5 measurements in a line. That gives a line of 5 numbers, one for each position as the scanner scans across. Those numbers are just like our column seed numbers. They are not about what is at a single point but a mixed up combination of what is on each scan line.

Now we take 5 horizontal scans. That gives us 5 more numbers, but this time through your head in a different direction. Between them the 2 sets of five numbers cover the same slice of brain though. The new 5 numbers are like the 5 row seed numbers.

Now we want to reconstruct the actual amount of matter at each position in the square slice through the head at that position. We just spread the column numbers down into our grid and spread the vertical numbers across, then add the two at each position to get an image (the Bingo card) of what was actually at each location. This is what back projection means. To create a real 2D slice with high detail, the 1D scans from lots of angles are all back-projected and added together and the image is processed further to sharpen it up. This calculation gives a precise 2D image of the location of bone and brain materials in your head. To create a 3D scan you simply stack the 2D slices together as you move the person's head through the scanner.

Just a quick one: The fast fives
Five fingers, five toes, fives are all around us. Impress your friends with your ability to divide any number by 5 at super speed, with your answer correct to three decimal places! Find out how to divide and conquer the fast five calculations at **www.cs4fn.org/mathemagic/**

Learn more at **www.cs4fn.org/mathemagic/**

The year is...

1089

The Future

Today's magic, one way or another, is likely to turn into the reality of tomorrow as scientists and engineers develop new technologies to achieve the effects. They may not do it the way the magicians imagined of course, but as with real magic it's the effect that matters!

Let's have a quick look at what may lie in store for some of the effects we've looked at here. To find out about the computer science behind these technologies and more browse the cs4fn website (www.cs4fn.org).

Controlling actions at a distance: Professor Kevin Warwick had a chip implanted in the nerves of his arm. When moving his fingers the signals from his brain could be transmitted over the Internet and control a robot hand that did the same thing. He was on a different continent to the robot.

Seeing into the future: Actually that is what science is about whether predicting climate change, future changes in the financial markets or even spotting people acting suspiciously at railway stations and predicting they might do something bad next…the more that science uncovers the way reality works the better our applications are at predicting the future.

Winning the lottery: There have been a whole series of syndicates using technology to beat the odds at games of chance – even roulette. The roulette gang used secret cameras and computers to record and analyse the rotation of the wheel and work out where the ball was most likely to stop. They were successful enough that the gambling laws had to be changed to disallow it.

Out of body experiences: That is what virtual reality is all about! If a virtual reality environment is connected to sensors back in the real world, your virtual self could watch events elsewhere, even with heightened senses. There is also research on using nanotechnology to allow a solid version of your avatar to coalesce elsewhere making your virtual presence turn physical.

Reading minds: MRI scanners can already watch your thoughts in action. Brain-computer interfaces can even read your mind to allow you to control computers using simple yes/no thoughts. At the moment it's mainly used to help stroke victims to communicate, but who knows in the future?

That's the kind of magic we do. What kind of magic do you do?

Curtain call

We hope you have enjoyed this booklet. There are more fascinating activities and stories about magic, technology and computer science on the cs4fn website at **www.cs4fn.org/mathemagic/** We hope you will have a look and have fun. As you impress your friends with your tricks, coming up with your own performance ideas and are basking in that applause:

Remember the Magician's Code and never
reveal the workings of magic tricks to your audience!