Gerald's Column
by Gerald Fitton

Thanks for your letters and emails. It is early days yet and Christmas is upon us so the correspondence which I've received about last month's article is somewhat sparse. I shall deal briefly with that correspondence before continuing with my theme, how mathematicians have influenced the development of the computer.

Mathematics, Invention or Discovery

Nobody has questioned my statement that Mathematicians were responsible for the early development of the computer but the general opinion amongst my correspondents is that Man invented Mathematics rather than discovering its existence. Well, I did say that it is widely believed, by non mathematicians in particular, that mathematics would not exist if there were no mathematicians to appreciate it. Many great mathematicians, as well as lesser ones such as I, believe that the fundamental nature of the physical universe can be understood only in terms familiar to Pure Mathematicians. I would go further and say that if the Physical Universe has a Creator then one of the many specialities of that Creator was Pure Mathematics. Creations are built in the image of their creator; by this I mean that every artifact contains some part of its creator. Our mathematical talent may be just a clouded image of the exceptional mathematical talent of our Creator.

The Future of Computers

To my surprise a couple of you wrote to say that you've read not only "Design for a Brain" by W Ross Ashby but also his companion volume "An Introduction to Cybernetics". The latter does contain more mathematics than the former but my correspondents point out that the mathematical parts contain nothing too exotic for them.

I have been reminded by others that computers do best when all relevant facts are available and when there is only one 'True' answer to a problem. Some have asked me whether I believe that the computer can be made to appear more intelligent by using a statistical approach (sometimes called 'Fuzzy Logic') when the problem requires more data than is currently available. I don't want to answer that question in any detail at the moment. Instead I shall draw your attention to the fact that children learn how to ride a bicycle without knowing anything about dynamics. Indeed they learn to ride with 'no hands' without knowing anything about the gyroscopic and castor action of the front wheel! They have learnt from what is commonly known as experience; in electronic and engineering terms this is called negative feedback. Children's brains have an aptitude for solving problems by using negative feedback. Try putting a young child in front of a computer and letting them get on with it. Unless you have done something similar before (or read the book!) you will be amazed how far they get. (Don't help them at all unless they ask for help and, if they do, give them a hint rather than a solution.)

Counting

$$\mathbb{N} = \{0, 1, 2, 3, 4, 5, 6, \dots\} \qquad = \text{Natural (counting) numbers}$$
$$\mathbb{Z} = \mathbb{N} + \{0, -1, -2, -3, \dots\} = \text{Integers (whole numbers)}$$
$$\mathbb{Q} = \mathbb{Z} + \{¼, ½, ¾, \dots\} \qquad = \text{Rational numbers (fractions)}$$
$$\mathbb{R} = \mathbb{Q} + \{\pm\sqrt{2}, \pi, e, \dots\} \qquad = \text{Real numbers (smooth continuum)}$$
$$\mathbb{C} = \mathbb{R} + \{\sqrt{-1}, (1 + \sqrt{-1}), \dots\} = \text{Complex (and imaginary) numbers}$$

To return to the different types of number which I introduced briefly last month. You will remember the table. The Natural Numbers, $\mathbb{N}$, can be used for counting. Not only that, if all you want to do is to count then you don't need any other sort of number.

	A	B	C
1			
2		Number	1
3		Number	2
4		Text	Fred
5		Name	VAT
6		Number	5
7		Text	Bill
8		Text	Charlie
9		Blank	
10		Number	3
11			
12		count(C1C11)	4
13		dcount(C1C11,C1C11<4)	3
14		dcounta(C1C11,C1C11<4)	3
15		dcount(C1C11,C1C11<"Fred")	4
16		dcounta(C1C11,C1C11<"Fred")	6
17			

Window title: m: scsi::HD4.$.PipeLine1.Archive.1998.9802.Files.C

PipeDream and Fireworkz have three counting functions namely count(), dcount(,) and dcounta(,) which I shall describe with the aid of the screenshot above.

In the block C2C10 of the PipeDream spreadsheet [Count01] I have included a mixture of numbers and text together with one blank space. The function count(C1C11) counts the number of numbers in the block; this is 4. The function dcount(C1C11,C1C11<4) counts

the number of numbers less than 4; this is 3. Jumping to dcounta(C1C11,C1C11<"Fred"); this function returns 6. The function dcounta(array,condition) counts not only all the cells containing numbers but also counts those cells containing text objects which start with letters before F in the alphabet. Bill and Charlie are counted as well as the four numbers.

The function count() is fast but dcounta(,) is very slow. I have not found a simple way of speeding up dcounta(,).

The function dcount(,) is rarely used with a textual condition but, when it is, it counts only numbers and not cells containing text objects.

Schema supports only count(); Eureka supports both count() and counta(,). These behave differently from their PipeDream equivalents (for example Eureka counts blank cells) so you need to check exactly what the function does before you use it.

Merging Sets

When you have counted the number of objects in a set (by handing out labels and looking at the last label) the number of objects counted this way is a Cardinal Number. As I said last month you can do sums with Cardinal Numbers. You can merge together sets containing similar things; three people in one set plus two people in another set equals five people in the merged set.

Demerging sets

You can split up sets. Of the five people in a room three are male and two are female. You can split the set if you have a condition which can by applied. In PipeDream (and many other spreadsheets) you create an extra column and enter an if(,,) function to do this. For example a function which looks at the content of a cell and executes if(male,M,F) will return either an M or F in the extra column. Sort on the extra column and split the database in two. I use such a technique to create a list of late payers so that I can send them a "hurry up and pay" note.

Creating New Numbers

You may have seen stickers in the back windows of cars which read "Wind surfers do it standing up" or "Scuba divers do it under water". I have never been sure what it is that is done but when I have suggested to my A Level mathematics students that "Mathematicians do it backwards" the conversation soon gets around to the most recently seen stickers – some of which have even more risque undertones than the two I have quoted. At least the risque nature of the slogans we discussed ensured that my students remembered the principles which I tried to teach them – I know this to be true because, some years afterwards, they told me that the 'do it backwards' lectures were in the 'more memorable' class along with the 'go anywhere triangle', 'the bouncing ball' and a few others.

Using as their starting point the Natural numbers, $\mathbb{N}$, mathematicians use this 'do it backwards' strategy to discover a new set of numbers, the Integers, $\mathbb{Z}$; I'll show you how.

Addition

Addition is a binary operation. If you take a couple of Natural numbers from the set ℕ and add them together the answer is always another Natural number from the same set. For example start with three people and add two more to create a set of five people. Five is a member of the set of Natural numbers, ℕ. There is no pair of Natural numbers which, when added together produce a number which is other than a Natural number. Addition does not expand the range of numbers from ℕ to ℤ.

Mathematicians insist on being able to 'do it backwards'. By this I mean we consider the following question: We know we started with three, we know we finished with five so how many more people came in? In mathematical jargon $3 + ? = 5$. If you prefer the problem set out in more formal mathematical notation it would read "Find a number n such that the equation $3 + n = 5$". Much time is spent by pupils learning how to convert implicit equations such as $3 + n = 5$ into an equation which starts $n = \ldots$ The process is called "Changing the subject of an equation".

After spending time at school 'Changing the subject' at university the initial emphasis was on so called Existence Theorems. A good question is: "Does n exist at all?" I shall not answer that but press on in the hope that your faith in me is sufficient for you to accept that the existence of n can be proved.

Subtraction

In the example we have the two numbers 5 and 3. We invent a new operation called Subtraction which reverses the process of Addition. In the appropriate notation we write $5 - 3 = ?$. The relationship between addition and subtraction is between an operation and its inverse. The same relationship exists between multiplication & division, the functions sin(x) & arcsin(x), differentiation & integration, and between implicit & explicit equations. The general principle at work here is that if you perform an operation and then perform the inverse of that operation, generally you get back to where you started from. Sometimes there are problems because you might be able to start from two (or more) different places and get to the same end point. I don't want to deal with this complication yet.

Subtraction is the inverse of addition. If we start with the Natural number five and subtract the Natural number three the result is the Natural number two. No problem!

As I have said, Addition of two numbers from the set ℕ always results in a Natural number from the same set. Let's ask the same question about our newly discovered operation, Subtraction. "Is the result of subtracting one Natural number from another always a Natural number?" Because you are better educated than most Roman mathematicians you will give the answer "No!" to that question.

The Integers

If you start with three and subtract five then the answer is not a Natural number. The answer is what secondary school pupils learn to call "Negative two". This new type of number is from a set larger than ℕ, it is from the set of Integers, ℤ. We have discovered

the Integers.

There are many people who just can't get to grips with the meaning of negative Integers. They have no conception of the nature and importance of Existence Theorems and it is highly likely that they would not believe in them even if they had. You certainly can't count "Negative two" people in a room. Nevertheless, negative Integers are useful to Engineers, Physicists and Bank Managers as well as being of interest to Mathematicians.

Unlike a few of my erstwhile students you will not be uneasy about the existence of negative Integers. Let me rephrase that. I guess that any unease felt by you, my readers, about there being "Negative two hundred pounds" in your bank account is related more to the difficulties of its repayment than the nature and meaning of its existence.

What I have tried to demonstrate is that, starting from the more easily understood Natural numbers and inverting the process of Addition, we have discovered a new set of numbers, $\mathbb{Z}$, the Integers. Later in this series we shall use the process of inversion to discover numbers which are even more bizarre than negative Integers.

Priorities

I have a BBC BASIC Guide for the Archimedes, Issue 1, published in 1988. My recollection is that it came with my old A440 machine when it ran under RISC OS 1 (the Arthur operating system). Is there a later version? On page 24 under the heading "Arithmetic Operators" a full list of arithmetic and logical operators is given. Each operator is given a priority. For example raising to a power using ^, as in 3^2, is Priority 2. By the way this 2 in "Priority 2" is an Ordinal number and not a Cardinal number. "Priority 2 + Priority 3" does not equal "Priority 5". Acorn could have labelled the priorities as A, B, C, etc with equal effect.

The binary operations of Addition (+) and Subtraction (−) have Priority 4 so that the sum 3^2+1 without brackets is evaluated as 3^2 = 9 followed by 9+1 = 10. If the priorities were reversed then the sum would be evaluated as 2+1 = 3 followed by 3^3 = 27. Brackets have Priority 1 so they can be used to force the evaluation of 3^(2+1) to make 27. As another by-the-way, multiplication (*) and division (/) have Priority 3. At the bottom of the list is Priority 7 containing the logical operators OR and EOR.

Many of my more mature readers may have learned the acronym BODMAS (Brackets, 'Of', Division, Multiplication, Addition, Subtraction) and used it to determine priorities when evaluating arithmetic expressions such as $2 + 3 \times 4 = ?$ As yet another by-the-way, even in my last year teaching at college some students were bringing in calculators which evaluated $2 + 3 \times 4$ as 20 and not the correct answer, 14.

The Unary Minus

The minus sign in the sum $3 - 5$ is not a unary minus. It is a minus sign having Priority 4 and its meaning is Subtract as in "Subtract the Natural number (plus) five from the Natural number three". This minus sign has nothing to do with the Natural number five which follows it.

The minus sign in the number which we've discovered, negative two, and written as – 2, is called a Unary Minus. What you mustn't do is try to unstick the unary minus from the two which follows it. If you do then one day you'll come unstuck with the minus sign. My better taught students refer to "Negative two" and not "Minus two" and do understand the difference between using Minus (an operator – a verb) to subtract a number and Negative (an adjective) which is part of the description of the number which follows "Negative".

So what priority is given in BBC BASIC to the unary minus? The unary minus has Priority 1 along with brackets, indirection operators (such as | which is a floating point indirection indicator), FN and the logical operator NOT! Both the unary minus and unary plus operators have Priority 1.

Finally, on most calculators it is difficult and on some it is impossible to enter a unary minus. With these latter machines it is necessary to resort to a trick such as typing in the expression (0 – 2) in order to enter negative two. Have a look at your calculator and see if you can make it evaluate (–2) * (–2). By the way, the answer is the Natural number (unary) positive four.

Unlike calculators, all the spreadsheets I know of, as well as BBC BASIC, have no problem whatsoever in handling the unary minus. In the formula line of any spreadsheet you can enter –2 * –2, as I have done here without brackets, and the correct answer, 4, will appear in the cell.

Storing Integers

I know I shall get many letters about my next remark. "Only Natural numbers from the set $\mathbb{N}$ can be stored in a computer!"

Your computer and mine resort to tricks in order to store any other type of number. I shall use BASIC V for my example. The range of integers which can be handled by BASIC V is from –2 147 483 648 to +2 147 483 647. These integers are stored in 4 bytes, 32 bits of pure binary. With 32 bits we can store 2^{32} = 4294967296 different numbers; the range of integers from –2 147 483 648 and +2 147 483 647 includes zero so this checks out OK.

In binary the number zero is stored as 0000 0000 0000 0000 0000 0000 0000 0000, 32 bits each having the value zero. For ease of writing what follows I shall convert this 32 bit number to the 8 digit hexadecimal number 0000 0000 but please bear in mind that each zero here is just a shorthand for the four bits 0000. As you might expect the Natural number one is stored (in this hex notation) as 0000 0001, fifteen as 0000 000F and sixteen as 0010. Similarly the Natural number +2 147 483 647 can be stored in 32 bit binary; in hex shorthand this is not FFFF FFFF but 7FFF FFFF; in 32 bit binary the first four bits are 0111.

Can you see where I'm going? Let me summarise. The range of Natural numbers from zero to +2 147 483 647 have used up the binary combinations corresponding to the range 0000 0000 to 7FFF FFFF in hex, exactly half the available range of binary combinations. We still have the range –2 147 483 648 to –1 to cover and we have the range 8000 0000 to FFFF FFFF available. Let me assure you that the Cardinality of these two sets (–2 147 483 648 to –1 and 8000 0000 to FFFF FFFF) is the same. What we need to do now is what mathematicians call mapping; we have to choose one binary combination for

each of the negative integers –2 147 483 648 to –1. You get one guess at how this is done and, unless you've thought about it I believe that you'll get it wrong.

Modular Arithmetic

The negative integer "Negative one", –1, is stored as the binary equivalent of FFFF FFFF, namely 1111 1111 1111 1111 1111 1111 1111 1111. The most negative number available in BBC BASIC V, –2 147 483 648 is stored as the binary equivalent of 8000 0000, namely 1000 0000 0000 0000 0000 0000 0000 0000. This number is 7FFF FFFF + 0001.

Perhaps the easiest way of understanding how this notation works is to consider not the usual number line which, close to zero, contains the Integers –3, –2, –1, 0, +1, +2, +3, but the odometer (sometimes called a mileometer) of a car. A three digit odometer when close to zero will read something like 997, 998, 999, 000, 001, 002, 003. The number "Negative 1" (miles) is represented by 999 and not –001; there is no 'space' in the three digit odometer for the unary minus sign.

Your computer does not store Negative Integers by storing a unary minus. What it does is to map the Natural numbers which have 1 as their first binary digit to the Negative Integers. It does so in a way which makes addition and subtraction of Integers, both negative and positive, much easier.

MOD in BASIC

I shall take as my example a decimal sum, (–2) + (+4) = (+2). In words this is "Start with Negative 2 and add to it Positive 4; the result is Positive 2" (I am using the words Negative and Positive as unary operators). Now translating this to three digit decimal (in which 1000 does not exist) I get 998 + 004 = 002. The type of arithmetic in which the 1 of the 1000 is discarded is well known to mathematicians; it is called 'Modular Arithmetic'. At the school my six year old grandson attends it is called "Clock Arithmetic" (22:00 plus 04:00 becomes 02:00 in the morning). You can simulate this sort of arithmetic in BASIC by using the MOD function. Try the BASIC instruction PRINT (998 + 004)MOD(1000) and you'll print the number 2. The mathematicians who designed BBC BASIC included MOD not only for modular arithmetic but for other reasons as well.

The Logical AND

In BASIC you will find that (1 AND 8) returns 0. If I ignore all the leading zeros then, in four digit binary, this is (0001 AND 1000). There are no matches between 0001 and 1000 so the answer to (1 AND 8) is 0.

Now try (–1 AND –8). What do you think it will return? Here's a clue. Without all the leading ones, in four digit binary, the expression is (1111 AND 1000).

I have included a couple of simple BASIC programs in the usual places so that you can evaluate both (998 + 4)MOD(1000) and (–1 AND –8) without struggling to learn how to write programs in BASIC.

Summary

The number two was probably the first number to be discovered. This was followed by the Natural numbers, ℕ, with zero being the last one found.

The discovery of Addition preceded the discovery of zero. Subtraction is the inverse of Addition. In spite of Subtraction having been around a long time negative Integers were denied an existence; indeed today many of my erstwhile students can't ascribe any meaning to these numbers.

For mathematicians the concept of Completeness is so important that they accept negative Integers as part of the Complete Set of Integers, ℤ. What Mathematical Completeness means is this: the operation of Addition together with its inverse, Subtraction, can be applied to any pair of Integers and the result is an Integer from the set ℤ. Unless a new operation is discovered then there are no more numbers to be discovered.

In my usual contentious manner I have suggested to you that only Natural numbers from the set ℕ are stored in our present day computers. Mathematicians resort to tricks such as mapping large Natural numbers to Negative Integers to represent other types of number. This trick has its advantages when it comes to executing Addition and Subtraction; these operations are executed in Modular ('Clock') Arithmetic. On the other hand the mapping of large Natural numbers to Negative Integers causes logical operators such as AND, OR and EOR to return answers which some people find strange when they are applied to negative Integers.

Next Month

Under the operations of Addition and Subtraction the set of Integers, ℕ, is complete in the mathematical sense. Next month we shall try out a new operation, Multiplication, and its inverse, Division. We shall apply these operations to the set ℤ and find out if it is sufficiently large to provide all possible answers.

Of course you know as well as I that it isn't. We shall discover a new type of number, ℚ, the Rational numbers. We shall discover that computers have problems storing such numbers accurately.

Finally

I have not forgotten that one of my objectives in this series of articles is to lead you to a vision of computers which rely less on logic and more on experience. Hence my whole set of articles about types of number is a massive digression – but then, my more mature students used to tell me that although they learned all they needed to know in order to pass, say, their accountancy exams, what would remain with them and be useful to them for the rest of their lives would be the well planned though apparently spontaneous digressions for which I became quite famous.

As usual your comments on this, or any other related subject from your experiences of negative numbers to the nature of the physical universe, will be most welcome.