Excursions in Computing Science: Week iii. Bases and Polynomials

T. H. Merrett*
McGill University, Montreal, Canada

August 2, 2024

I. Prefatory Notes

1. Base 10. Teacher, invite your grade scholar to talk about the following picture.

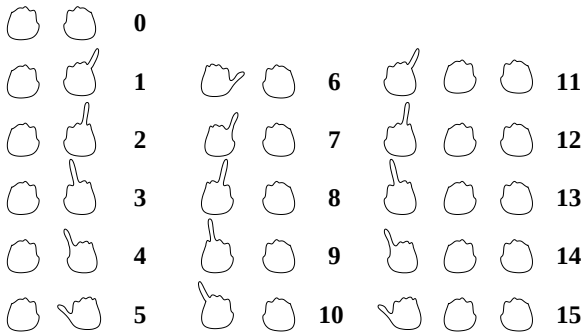
Why is the right thumb not used? Which numbering systems have exactly ten different symbols? Which number appears in these systems that does not in the others? In the other numbering systems how are the first ten different symbols used differently from the systems with exactly ten? Help your grade scholar include the Roman numbering system and discuss that. What is the largest number that can be represented in each of the systems? How many symbol positions are needed in each system to represent the number 9999?

We use ten different symbols to count because our two hands have ten fingers. But, as the picture shows, the way to represent the symbols by fingers is not completely obvious. The great invention of the Hindu mathematicians around the year 600 of the present era was the symbol for zero. They discovered that zero could be used as a place-holder for each position in their numbering system, so that they could express really large numbers, which their philosophy of time required. The Hindus passed this idea on to the Islamic mathematicians who passed it to the rest of the world via the Arabic numbering system which is now universally used.

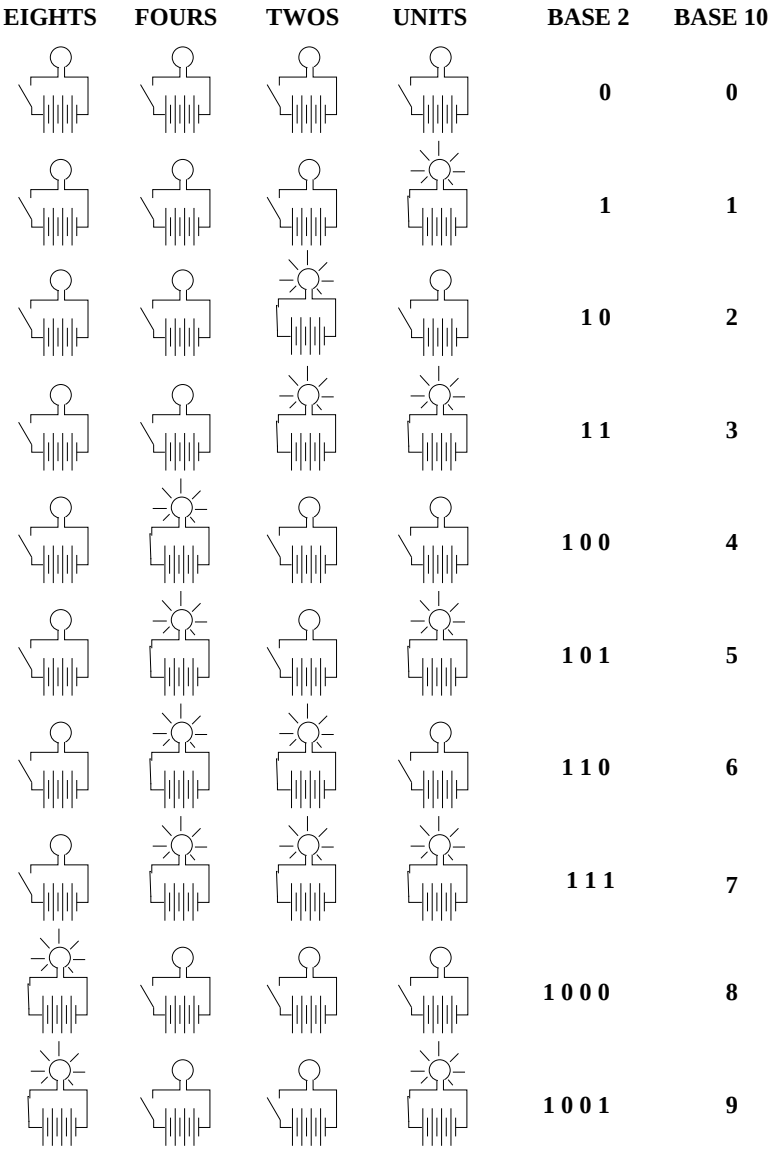
 The author gratefully acknowledges support from the taxpayers of Québec and of Canada who have paid his salary and research grants while this work was developed at McGill University, and from his students and their funding agencies.

HUNDREDS	TENS	UNITS		INDIA <~600P.E.	MODERN CHINA/JAPAN	MODERN SANSCRIT	MODERN ARAB
			0			०	٠
			1	𑁒	一	१	١
			2	𑁓	二	२	٢
			3	𑁔	三	३	٣
			4	𑁕	四	४	٤
			5	𑁖	五	५	٥
			6	𑁗	六	६	٦
			7	𑁘	七	७	٧
			8	𑁙	八	८	٨
			9	𑁚	九	९	٩
			10	𑁛	十	१०	١٠
			11	𑁛𑁒	十一	११	١١
			12	𑁛𑁓	十二	१२	١٢
			13	𑁛𑁔	十三	१३	١٣
			14	𑁛𑁕	十四	१४	١٤
			15	𑁛𑁖	十五	१५	١٥
			16	𑁛𑁗	十六	१६	١٦
			17	𑁛𑁘	十七	१७	١٧
			18	𑁛𑁙	十八	१८	١٨
			19	𑁛𑁚	十九	१९	١٩
			20	𑁜	二十	२०	٢٠
			99	𑁜𑁚	九十九	९९	٩٩
			100	𑁝	百	१००	١٠٠

Here is a not so good way to count on ten or more fingers.



2. Base 2. Computers have only two "fingers": *off* (0) and *on* (1).



We can count as high as we like with zeros and ones only, following the same structure.

Here is a way to convert from a base-10 ("decimal") number to its base-2 ("binary") equivalent.

(The notation $\lfloor \frac{25}{2} \rfloor$ gives the integer immediately below $\frac{25}{2}$, i.e., 12.) You can see the general idea from the example of 25.

First, keep dividing by 2 and finding the remainder.

$$\begin{aligned}
25 &= 1 + 2\lfloor \frac{25}{2} \rfloor \\
&= 1 + 2(12) \\
&= 1 + 2(0 + 2\lfloor \frac{12}{2} \rfloor) \\
&= 1 + 2(0 + 2(6)) \\
&= 1 + 2(0 + 2(0 + 2\lfloor \frac{6}{2} \rfloor)) \\
&= 1 + 2(0 + 2(0 + (3))) \\
&= 1 + 2(0 + 2(0 + (1 + 2\lfloor \frac{3}{2} \rfloor))) \\
&= 1 + 2(0 + 2(0 + (1 + 2(1))))
\end{aligned}$$

Next, remove all the 2s and write the remainders down *backwards*.

$$25_{10} = 11001_2$$

The subscripts, $_{10}$ and $_2$, give the two bases.

Here is a way to go back from binary to decimal, using the same example.

First, reverse the order of symbols and insert additions and factors of 2.

$$11001 \rightarrow 1 + 2(0 + 2(0 + 2(1 + 2(1))))$$

Next, work this out.

$$\begin{aligned}
1 + 2(0 + 2(0 + 2(1 + 2(1)))) &= 1 + 2(0 + 2(0 + 2(1 + 2))) \\
&= 1 + 2(0 + 2(0 + 2(3))) \\
&= 1 + 2(0 + 2(0 + 6)) \\
&= 1 + 2(0 + 2(6)) \\
&= 1 + 2(0 + 12) \\
&= 1 + 2(12) \\
&= 1 + 24 \\
&= 25
\end{aligned}$$

So why are the symbols written such that they must be reversed for both conversions? Because, as with base 10, the convention makes the leftmost symbol the most important one. For example, in 3241_{10} we can drop the latter symbols and approximate with $3 \times 10^3 = 3000_{10}$. Similarly, 101101110_2 is a little larger than $1 \times 2^8 = 100000000_2$.

3. Base 3. Any base is possible. Teddy bears have no fingers but they can use their two arms. Here is Beru counting with the three different symbols, 0, 1, 2, which hey represents respectively by no arms up, one arm up, two arms up.

NINES	THREES	UN ITS	base 3	base 10
			**0**	**0**
			**1**	**1**
			**2**	**2**
			**10**	**3**
			**11**	**4**
			**12**	**5**
			**20**	**6**
			**21**	**7**
			**22**	**8**
			**100**	**9**
			**101**	**10**
			**102**	**11**
			**110**	**12**
			**111**	**13**
			**112**	**14**
			**120**	**15**

NINES	THREES	UN ITS	base 3	base 10
			**121**	**16**
			**122**	**17**
			**200**	**18**
			**201**	**19**
			**202**	**20**
			**210**	**21**
			**211**	**22**
			**212**	**23**
			**220**	**24**
			**221**	**25**
			**222**	**26**
			1000	**27**
			1001	**28**
			1002	**29**
			1010	**30**
			1011	**31**

Conversion from "ternary" to decimal and back again follows a similar procedure.

$$\begin{aligned} 25 &= 1 + 3\lfloor \frac{24}{3} \rfloor \\ &= 1 + 3(8) \\ &= 1 + 3(2 + 3\lfloor \frac{8}{3} \rfloor) \\ &= 1 + 3(2 + 3(2)) \\ 25_{10} &= 221_3 \end{aligned}$$

Back again

$$221_3 \rightarrow 1 + 3(2 + 3(2))$$

$$\begin{aligned} 2 + 3(0 + 3(2)) &= 2 + 3(2 + 6) \\ &= 1 + 3(8) \\ &= 1 + 24 \\ &= 25 \end{aligned}$$

4. Bases and powers. At heart, a number system is constructed on powers of its base. Here is 25 again for bases 2, 3 and 10, this time more explicitly.

$$\begin{aligned} 25 &= 2^4 + 2^3 + 2^0 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 11001_2 \\ 25 &= 2 \times 3^2 + 2 \times 3^1 + 1 \times 3^0 = 221_3 \\ 25 &= 2 \times 10^1 + 5 \times 10^0 = 25_{10} \end{aligned}$$

Conversely, logarithms tell us how many places are needed to represent an integer.

Since 25 lies between 2^4 and 2^5, it requires 5 binary digits ("bits").

Since 25 lies between 3^2 and 3^3, it requires 3 ternary digits (which I guess could be called "trits").

Since 25 lies between 10^1 and 10^2, it requires 2 decimal digits (simply "digits").

The way to write this is $\lceil \log_b(n) \rceil$, the next integer above the logarithm to base b of n, e.g.,

$$\begin{aligned} \lceil \log_2(25) \rceil &= 5 \\ \lceil \log_3(25) \rceil &= 3 \\ \lceil \log_{10}(25) \rceil &= 2 \end{aligned}$$

How many bits would be needed to store a 5-digit integer? That would be an integer between 10^5 and 10^6. Check out both extremes.

$$\lfloor \log_2(10^5) \rfloor = \lfloor 5 \log_2(10) \rfloor = \lfloor 5 / \log_{10}(2) \rfloor = \lfloor 5/0.30103 \rfloor = \lfloor 5 \times 3.3219 \rfloor = \lfloor 16.61 \rfloor = 16$$

$$\lceil \log_2(10^6) \rceil = \lceil 6 \log_2(10) \rceil = \lceil 6 / \log_{10}(2) \rceil = \lceil 6/0.30103 \rceil = \lceil 6 \times 3.3219 \rceil = \lceil 19.93 \rceil = 20$$

(Why use floor ($\lfloor \rfloor$) of the minimum and ceiling ($\lceil \rceil$) of the maximum?)

So a 5-digit integer will lie between $2^{16} = 65536$ and $2^{20} = 1\ 024\ 000$.

Note that multiplying the number of digits by lg 10 = 1/0.30103 to get the number of bits is a refinement over multiplying by 10/3, which we used in Week ii.

For large quantities of data, computer people use "bytes" rather than bits. A byte is 8 bits, and working with bytes is working with base $2^8 = 256$. To find the number of bytes needed to store B bits, use $\lceil B/8 \rceil$:

16 bits needs 2 bytes
21 bits needs 3 bytes
24 bits needs 3 bytes

5. Rationals < 1. Not only integers may be represented in different bases. The fraction 1/2 is 0.5 in decimal (base 10). It is also 0.1 in binary (base 2). Here's how.

We write the fraction as a series of unknown coefficients divided by powers of the base (or multiplied by negative powers of the base). Then we systematically figure out each unknown in turn until the pattern of unknowns repeats or we hit zero exactly. We revert to the forms of the series we used in Notes 1–3 rather than using the powers explicitly.

The secret in figuring out the unknowns is that each must be a one-symbol integer (one digit 0–9, one bit 0–1, etc., depending on the base).

In base 10,

$$\frac{1}{2} = \frac{1}{10}(p_{-1} + \frac{1}{10}(p_{-2} + \frac{1}{10}(p_{-3} + ..)))$$

and so we can multiply both sides by 10 to get

$$5 = p_{-1} + \frac{1}{10}(p_{-2} + \frac{1}{10}(p_{-3} + \frac{1}{10}(p_{-4} + ..)))$$

This means $p_{-1} = 5$, the nearest integer.

So we subtract 5 and p_{-1} from their respective sides, giving

$$0 = \frac{1}{10}(p_{-2} + \frac{1}{10}(p_{-3} + \frac{1}{10}(p_{-4} + ..)))$$

and that tells us $p_{-2} = 0$.

The left-hand side is exactly zero, so we know that all further unknowns are 0 and we can stop: $1/2 = 5/10 = 0.5_{10}$

Here is the same process for base 2.

$$\begin{aligned} \frac{1}{2} &= \frac{1}{2}(p_{-1} + \frac{1}{2}(p_{-2} + \frac{1}{2}(p_{-3} + ..))) \\ 1 &= p_{-1} + \frac{1}{2}(p_{-2} + \frac{1}{2}(p_{-3} + \frac{1}{2}(p_{-4} + ..))) \\ 0 &= \frac{1}{2}(p_{-2} + \frac{1}{2}(p_{-3} + \frac{1}{2}(p_{-4} + ..))) \end{aligned}$$

From the second line, $p_{-1} = 0$ and from the third $p_{-2} = 0$ and so do all the rest: $1/2 = 0.1_2$

Here are ten fractions in six bases to work out. Note that the patterns either repeat or stop. (Stopping really just means that 0 is repeated.)

fraction	base 10	base 2	base 3	base 4	base 5	base 6
1/2	0.5	0.1	0.1111..	0.2	0.2222..	0.3
1/3	0.3333..	0.0101..	0.1	0.1111..	0.1313..	0.2
2/3	0.6666..	0.1010..	0.2	0.2222..	0.3131..	0.4
1/4	0.25	0.01	0.0202..	0.1	0.1111..	0.13
3/4	0.75	0.11	0.2020..	0.3	0.3333..	0.43
1/5	0.2	0.00110011..	0.01210121..	0.0303..	0.1	0.1111..
2/5	0.4	0.01100110..	0.10121012..	0.1212..	0.2	0.2222..
3/5	0.6	0.10011001..	0.12101210..	0.2121..	0.3	0.3333..
4/5	0.8	0.11001100..	0.21012101..	0.3030..	0.4	0.4444..
1/6	0.1666..	0.00101..	0.0111..	0.0222..	0.1111..	0.1
5/6	0.8333..	0.111	0.2111..	0.3111..	0.4111..	0.5
1/7? 1/8?	base 7?	base 8?	base 9?	base 11?	base 60?	

$$\frac{n}{d} = \frac{1}{b}\left\langle p_{-1} + \frac{1}{b}\left\langle p_{-2} + \frac{1}{b}\left\langle p_{-3} + .. \right. \right. \right. = 0.p_{-1}p_{-2}p_{-3}.. \text{ , with } 0 \le p_i < b \quad \text{for } n < d$$

At the bottom of this figure I have written the series using a novel opening bracket which does not need the tedious sequence of closing brackets all at the end of the series. (Note how the form of the bracket implies multiplication.)

I have also used negative indexes for the coefficients. This is not necessary, but these indexes give the powers of the base, and that is handy.

6. The Genetic Code. Our DNA is written in base 4. This is because DNA presents sequences of

four submolecules, the "purines" adenine and guanine, and the "pyrimidines" thymine and cytosine. These are represented by the letters `a, g, t` and `c`, respectively. (They are called "base pairs" because they pair up across the double strand of the DNA helix, `a` with `t` and `g` with `c`. In the following, we will think in terms of a single strand of DNA, so the pairing does not make any difference to this discussion.)

Genes are sequences of base pairs.

Each gene specifies a set of "amino acids", which are the proteins and the biochemical catalysts that the cell uses to build features of our bodies such as blue eyes or curly hair. There are 20 different amino acids, so they can be thought of as written in base 20.

Here are the twenty amino acid names, together with the usual abbreviations and a letter of the alphabet (`B, J, O, U, X` and `Z` are not used) that can represent them to base 20.

Amino acid	*Abbrev.*	$b = 20$	*Amino acid*	*Abbrev.*	$b = 20$
alanine	Ala	`A`	arginine	Arg	`R`
aspartic acid	Asp	`D`	asparginine	Asn	`N`
cysteine	Cys	`C`	glutamic acid	Glu	`E`
glutamine	Gln	`Q`	glycine	Gly	`G`
hist[id]ine	His	`H`	isoleucine	Ile	`I`
leucine	Leu	`L`	lysine	Lys	`K`
methionine	Met	`M`	phenylalanine	Phe	`F`
proline	Pro	`P`	serine	Ser	`S`
threonine	Thr	`T`	tryptophan	Trp	`W`
tyrosine	Tyr	`Y`	valine	Val	`V`

How can four base-pair letters, `a, g, t` and `c` "code" for twenty amino acids? Single letters can distinguish only four different outcomes. Double letters only $4^2 = 16$. So it takes at least triples of letters to distinguish twenty. A triplet can in fact distinguish $4^3 = 64$ different outcomes, so triplet coding is sufficient and also highly redundant. Reduncancy is good, given the complexity of the biochemical machinery at work in cell division and reproduction, and given the possibility of errors in copying genes.

The base-pair triplets are called "codons".

Here, then, is the "genetic code", the actual mapping between base-pair triplets and the 20 amino acids. To be conventional, `u`, standing for the pyrimidine, uracil, appears instead of `t`, because uracil replaces thymine when the code is copied to RNA during the cell's process of making amino acids.

	`u`	`c`	`a`	`g`	
	Phe	Ser	Tyr	Cys	`u`
`u`	Phe	Ser	Tyr	Cys	`c`
	Leu	Ser	STP	STP	`a`
	Leu	Ser	STP	Trp	`g`
	Ile	Thr	Asn	Ser	`u`
`a`	Ile	Thr	Asn	Ser	`c`
	Ile	Thr	Lys	Arg	`a`
	Met	Thr	Lys	Arg	`g`

	`u`	`c`	`a`	`g`	
	Leu	Pro	His	Arg	`u`
`c`	Leu	Pro	His	Arg	`c`
	Leu	Pro	Gln	Arg	`a`
	Leu	Pro	Gln	Arg	`g`
	Val	Ala	Asp	Gly	`u`
`g`	Val	Ala	Asp	Gly	`c`
	Val	Ala	Glu	Gly	`a`
	Val	Ala	Glu	Gly	`g`

Note that three of the codons code not for an amino acid but for the end of a gene: STP (`uaa, uga` and `uag`). Genes start at `aug`, the codon that also codes for methionine.

While the average redundancy is $64/21 \approx 3$ for the 20 amino acids and STP, it actually ranges from 1 (methionine and tryptophan) to 6 (arginine, leucine and serine).

7. Polynomials. An integer written as a series in powers of its base has a form such as

$$25 = 1 \times b^4 + 1 \times b^3 + 0 \times b^2 + 0 \times b^1 + 1 \times b^0$$

where, in this case, the base $b = 2$.

Just as we can do arithmetic on integers, we can also do arithmetic on a construct which generalizes this representation by base (for which x is normally used instead of b).

$$1 \times x^4 + 1 \times x^3 + 0 \times x^2 + 0 \times x^1 + 1 \times x^0$$

or

$$2 \times x^5 + 6 \times x^4 + 2 \times x^3 + 1 \times x^2 + 4 \times x^1 + 4 \times x^0$$

or

$$1 \times x^6 + 9 \times x^5 + 9 \times x^4 + 6 \times x^3 + 0 \times x^2 + 8 \times x^1 + 0 \times x^0$$

(These can be more succinctly written, e.g., the latter: $x^6 + 9x^5 + 9x^4 + 6x^3 + 8x$)

The x is really only a placeholder and we can leave it out, too. This helps us in seeing how to add two polynomials. For example

			2	6	2	1	4	4
+		1	9	9	6	0	8	0
=		1	11	15	8	1	12	4

So $2x^5 + 6x^4 + 2x^3 + x^2 + 4x + 4$

$+ \quad x^6 + 9x^5 + 9x^4 + 6x^3 \quad + 8x$

$= \quad x^6 + 11x^5 + 15x^4 + 8x^3 + x^2 + 12x + 4$

Because addition of numbers is commutative and associative, clearly so is addition of polynomials. We can add a whole bunch of polynomials in any order.

				2	6	2	1	4	4
+			1	9	6	6	0	8	0
+		2	2	9	3	7	6	0	0
+		6	5	5	3	6	0	0	0
+	9	8	3	0	4	0	0	0	0
=	9	16	11	25	22	21	7	12	4

Subtraction of two polynomials is also easy.

	1	11	15	8	1	12	4
−	1	9	9	6	0	8	0
=		2	6	2	1	4	4

8. Multiplying polynomials. First, polynomials can be multiplied by numbers.

8×	3	2	7	6	8
=	24	16	56	48	64

So $3x^4 + 2x^3 + 7x^2 + 6x + 8$

$\times \quad 8$

$= \quad 24x^4 + 16x^3 + 56x^2 + 48x + 64$

Second, polynomials can be multiplied by polynomials. Here is a simple example.

$$(3x^4 + 2x^3 + 7x^2 + 6x + 8) \times (x + 1) = 3x^5 + 5x^4 + 9x^3 + 13x^2 + 14x + 8$$

How do we figure this?

		x^5	x^4	x^3	x^2	x	1
			3	2	7	6	8
x	1	3	2	7	6	8	
1	1		3	2	7	6	8
		3	5	9	13	14	8

Notice how multiplication by x shifts the whole row one position leftwards.

To be a little more ambitious, let's find the square of the first polynomial: the polynomial multiplied by itself. Instead of shifting rows leftwards, we'll keep the rows aligned but add up along diagonals angling leftwards from the top.

					3	2	7	6	8
$3\times$					9	6	21	18	24
$2\times$					6	4	14	12	16
$7\times$					21	14	49	42	56
$6\times$					18	12	42	36	48
$8\times$					24	16	56	48	64
$=$	9	12	46	64	121	116	148	96	64

So $3x^4 + 2x^3 + 7x^2 + 6x + 8$
$\times \quad (3x^4 + 2x^3 + 7x^2 + 6x + 8)$
$= \quad 9x^8 + 12x^7 + 46x^6 + 64x^5 + 121x^4 + 116x^3 + 148x^3 + 96x + 64$

To help visualizing this, here are the powers of x involved for the terms in this example.

					x^4	x^3	x^2	x^1	x^0
x^4					x^8	x^7	x^6	x^5	x^4
x^3					x^7	x^6	x^5	x^4	x^3
x^2					x^6	x^5	x^4	x^3	x^2
x^1					x^5	x^4	x^3	x^2	x^1
x^0					x^4	x^3	x^2	x^1	x^0
	x^8	x^7	x^6	x^5	x^4	x^3	x^2	x^1	x^0

9. Polynomial division. Polynomials add, subtract and multiply the same way numbers do. Polynomials divide like integers. Polynomial division gives quotients and remainders.

As with integer division, polynomial division requires enough experience with multiplication to be able to guess the answer. So let's try the simple example we saw before (Note 8).

$$(3x^5 + 5x^4 + 9x^3 + 13x^2 + 14x + 8) \div (x + 1) = 3x^4 + 2x^3 + 7x^2 + 6x + 8$$

Division may not be exact:

$$3x^5 + 5x^4 + 9x^3 + 13x^2 + 14x + 7 = (3x^4 + 2x^3 + 7x^2 + 6x + 8) \times (x + 1) + (-1)$$

Here the *remainder* is -1. The *quotient* is $3x^4 + 2x^3 + 7x^2 + 6x + 8$ as in the exact case.

The factors of $3x^4 + 2x^3 + 7x^2 + 6x + 8$ are

$$x^2 - 0.73678006553031x + 2.09370677108510$$

and

$$x^2 + 1.40344673219697x + 1.27365813756461$$

and

3

We see that polynomials need not have integer coefficients. In fact, these coefficients are not even rational (ratios of integers) although they can be expressed in closed form.

For example, -0.73678006553031 is only an approximation for

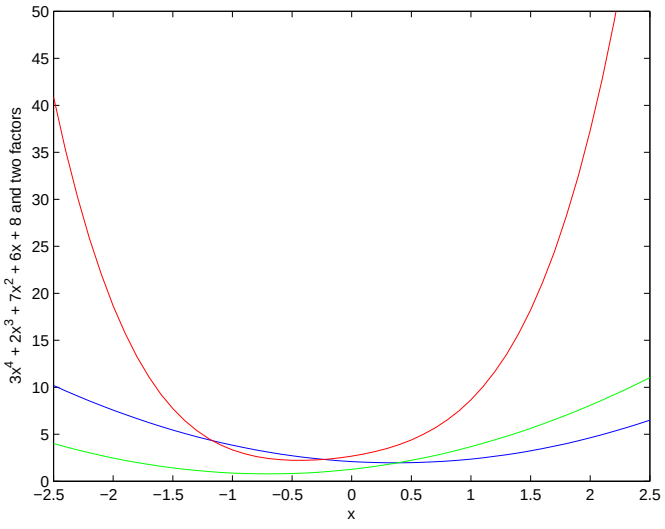
$$
\begin{aligned}
&1/3 - 1/36\times\sqrt{}(-468 - 602/\sqrt[3]{}(4193/5832+7/972\times i\times\sqrt{}(5493)) \\
&\qquad\qquad - 648\times\sqrt[3]{}(4193/5832+7/972\times i\times\sqrt{}(5493)) \\
&\qquad)\\
&\quad - 1/36\times\sqrt{}(-936 + 602/\sqrt[3]{}(4193/5832+7/972\times i\times\sqrt{}(5493)) \\
&\qquad\qquad + 648\times\sqrt[3]{}(4193/5832+7/972\times i\times\sqrt{}(5493)) \\
&\qquad\quad -14688/\sqrt{}(-468 - 602/\sqrt[3]{}(4193/5832+7/972\times i\times\sqrt{}(5493)) \\
&\qquad\qquad\qquad - 648\times\sqrt[3]{}(4193/5832+7/972\times i\times\sqrt{}(5493)) \\
&\qquad\qquad)\\
&\qquad)\\
&-\text{ conj}(\ 1/36\times\sqrt{}(-468 - 602/\sqrt[3]{}(4193/5832+7/972\times i\times\sqrt{}(5493)) \\
&\qquad\qquad - 648\times\sqrt[3]{}(4193/5832+7/972\times i\times\sqrt{}(5493)) \\
&\qquad)\\
&\quad + 1/36\times\sqrt{}(-936 + 602/\sqrt[3]{}(4193/5832+7/972\times i\times\sqrt{}(5493)) \\
&\qquad\qquad + 648\times\sqrt[3]{}(4193/5832+7/972\times i\times\sqrt{}(5493)) \\
&\qquad\quad -14688/\sqrt{}(-468 - 602/\sqrt[3]{}(4193/5832+7/972\times i\times\sqrt{}(5493)) \\
&\qquad\qquad\qquad - 648\times\sqrt[3]{}(4193/5832+7/972\times i\times\sqrt{}(5493)) \\
&\qquad\qquad)\\
&\qquad)\\
&\quad)
\end{aligned}
$$

Here, some new symbols appear (i and conj) which we won't understand until Week 4. Suffice it to say here that these two symbols cancel each other out in the end for this case, although they are needed to write the coefficient down as I have done.

It actually comes from a MATLAB program for solving "quartic equations".

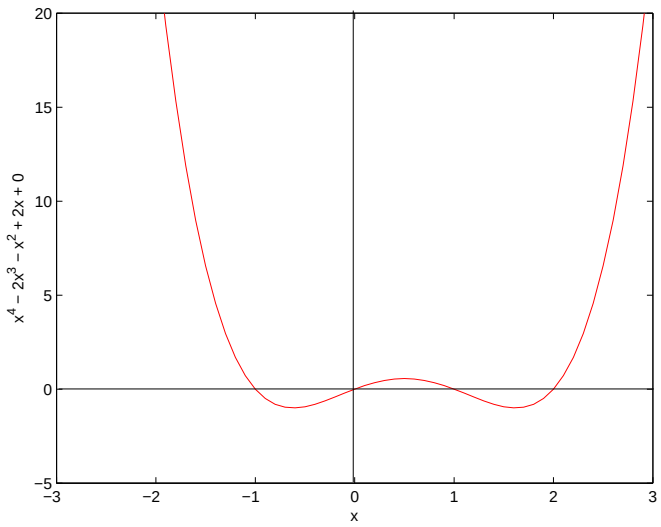
We won't try to write out the other three non-1 coefficients.

A picture is more directly useful. Here are $3x^4+2x^3+7x^2+6x+8$ and the two polynomial factors I've just given.



Note that none of these curves cross the x-axis. This means that there are no "linear factors", i.e., of the form $x-a$, for the original polynomial. Why?

Here is a polynomial which does have linear factors, and its picture.



10. Bases and polynomials The formal processes of arithmetic on integers (or on rational numbers expressed using a decimal point ("binary point", "ternary point", ..) are based on the arithmetic of polynomials we have just seen, with one additional complication.

Let's find the square of 32768. This is the polynomial $3x^4 + 2x^3 + 7x^2 + 6x + 8$ with $x = 10$.

So we can repeat the multiplication procedure of Note 8.

					3	2	7	6	8
3×					9	6	21	18	24
2×					6	4	14	12	16
7×					21	14	49	42	56
6×					18	12	42	36	48
8×					24	16	56	48	64
=	9	12	46	64	121	116	148	96	64

But each "coefficient" must now be a digit in the representation of the result, and so cannot have more than one digit.

We must introduce "carrying", to carry any extra digits leftwards to the next position ("coefficient").

Thus, $3\times(3 \times 10^4 + 2 \times 10^3 + 7 \times 10^2 + 6 \times 10 + 8)$ (the second row in the table above) is not $9 \times 10^4 + 6 \times 10^3 + 21 \times 10^2 + 18 \times 10 + 24$ but $9 \times 10^4 + 8 \times 10^3 + 3 \times 10^2 + 0 \times 10 + 24$ Check how the "carries" work from right to left to give this answer.

Doing this same process for each row of the table, we get

						3	2	7	6	8
3×						9	8	3	0	4
2×						6	5	5	3	6
7×					2	2	9	3	7	6
6×					1	9	9	6	0	8
8×					2	6	2	1	4	4
=		9	16	11	25	25	21	7	12	4
=>	1	0	7	3	7	7	1	8	2	4

The seventh row is the sum of the rows as in polynomial multiplication in Note 8, added up along diagonals angling leftwards from the top.

But we must also reduce each of these coefficients to a single digit, so we carry here, too, producing the answer in the last row: $32\ 768^2 = 1\ 073\ 771\ 824$.

This example illustrates addition and multiplication. Similar analogies between integers and polynomials may be made for subtraction and division.

11. Summary

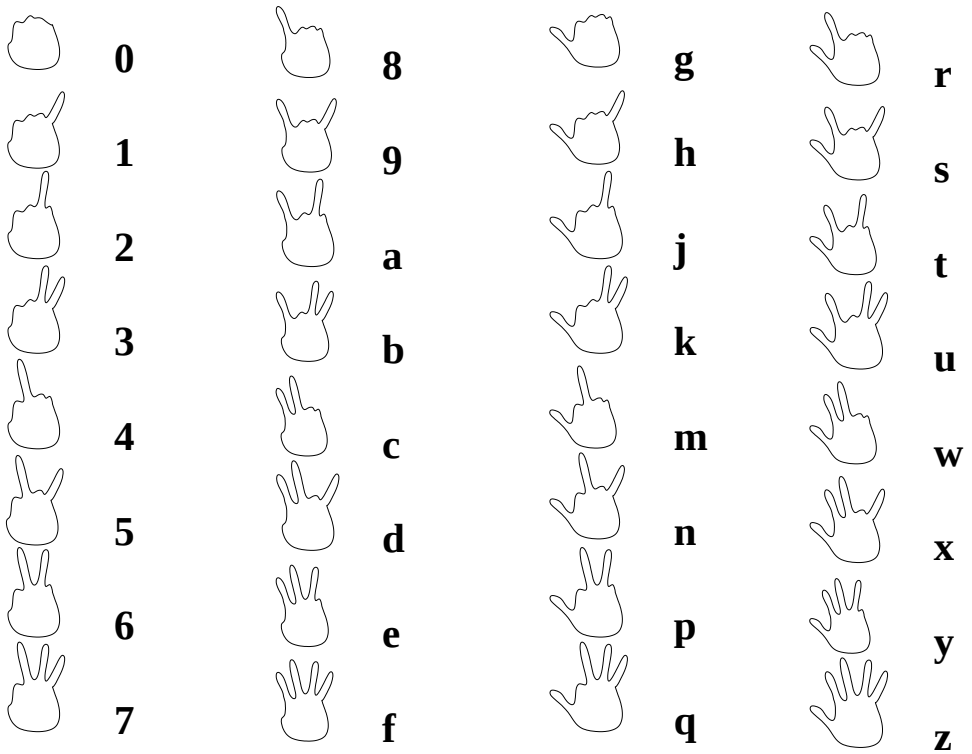
(These notes show the trees. Try to see the forest!)

- Decimal number systems and zero.
- Base 2 and conversion between base 10, base 2.
- Base 3 and conversion between base 10, base 3.
- Bases, powers, logarithms: number of bits or digits to represent a given number.
- Fractions and conversions.
- Polynomials: addition, subtraction, multiplication, division; divisors and factors; linear factors and zeros.
- Polynomials and integer arithmetic in various bases.

II. The Excursions

You've seen lots of ideas. Now *do* something with them!

1. Try using your fingers and toes to count up to 99 in the way shown in the first figure of Note 1. How high can you count using the way shown in the second figure of Note 1?
2. Try some other ways to use your fingers to count to 10.
3. Look up the history of the idea of zero.
4. Study the *abacus*, the first computing machine, and discuss it in terms of the five fingers on each hand.
5. How should we approximate a number such as 3541_{10} to the nearest thousand?
6. Since Beru has only two arms (or paws), shouldn't hey be able to count only in base 2? What happened? Can you think of ways to use one hand to count in base 6 or two hands in base 11?
7. How would we convert between base 4 and base 10? Write the first 30-odd integers using four different symbols.
8. Here is a way you can count on your hands in base 32. Can you figure it out? (It may take you some practice to be able to do it, even to get your fingers in the right positions.)



How can you count on your hands in base 1024?

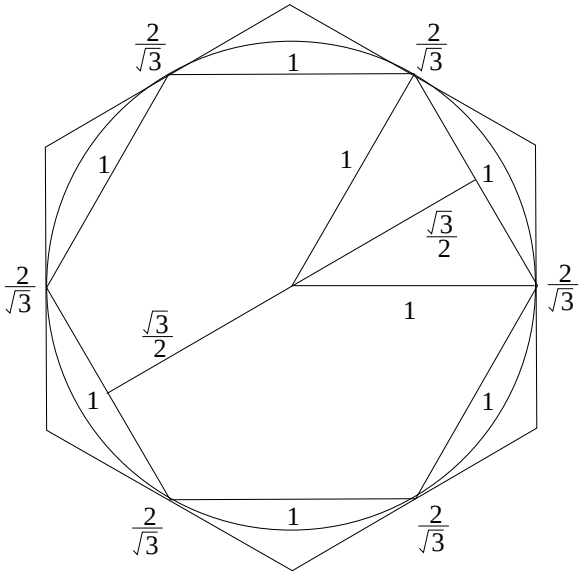
9. Can Beru find a way to count in base 4?

10. The Tokyo University computer scientist, Yasumasa Kanada, and his group used 400 hours on a Hitachi supercomputer in 2002 (after five years of programming) to calculate π to 1 241 100 000 000 000 decimal places. How many bytes would be needed to store the whole result? Give the result both in gigabytes and Gigabytes.

11. One way to find π, but probably not the way used by Prof. Kanada, starts as follows. By drawing two hexagons, one inside ("inscribed in") a circle of radius 1 and one outside ("circumscribing") the circle, we can see that

$$6 < 2\pi < 6\frac{2}{\sqrt{3}} = 6.928..$$

so π is between 3 and 3.46, because the side of the inner hexagon is 1 and of the outer is $2/\sqrt{3}$.

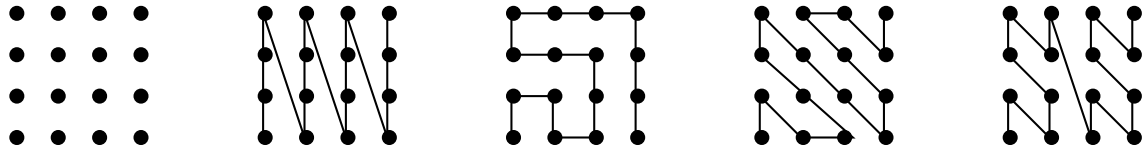


To find the $2/\sqrt{3}$ we need Pythagoras: the "radius" of the inner hexagon is $\sqrt{3}/2$ because it forms a right-angled triangle bisecting an equilateral triangle of side 1. To scale up from the inner to the outer hexagon, we must increase the centre–vertex distance from 1 to $1/(\sqrt{3}/2)$. Check these arguments.
This is a very crude estimate, but can be refined by increasing the number of sides of the polygon inscribed and circumscribed. If you know trigonometry (see Week 1), try doubling the number of sides to a dodecagon. Try doubling again.

12. Find the first few perfect numbers (excursion *Perfect numbers* in Week ii) in binary? What is the pattern for all perfect numbers?

13. What underlies the following pattern?

14. **Z-order** ("zee-order"). How can we make a simple ordering of a grid of points in two dimensions? Here is a way in which we are helped by binary numbers.



The figure shows four ways we could connect all the points of a 2-D grid into a single line, and you can certainly think of more. The choice among them will be dictated by which gives the easiest way to calculate the position on the line given the two indices, and vice-versa.
For example, the Scan method would turn the 1,1 position into the 1st point on the line, the 1,2 position into the 2nd point, .., the 2,1 position into the 5th point, and so on. There is a way to calculate this: see the Excursion *Array addressing* in Week iv. But we must know how high the grid is for a formula to work.
You can try deriving formulas for the Circumference and the Diagonal orderings: both are awkward.
The Peano ordering (Giuseppe Peano, 1858–1932) at first seems pretty awkward, too, but turning everything into binary makes calculations both ways very easy. It also generalizes to any number of dimensions, not just 2-D, in which form it is called "zee-order" with the U.S. pronunciation acknowledging its creator, Jack Orenstein.
It helps to start our index and address counting from 0 not 1, so we do that from now on. Here are the addresses of the points in this 4-by-4 case (that is, their positions along the Z-order line).

j					
3	5	7	13	15	
2	4	6	12	14	
1	1	3	9	11	
0	0	2	8	10	
	0	1	2	3	k

j					
11	0101	0111	1101	1111	
10	0100	0110	1100	1110	
01	0001	0011	1001	1011	
00	0000	0010	1000	1010	
	00	01	10	11	k

The version to the right is the same as that on the left, but with all numbers converted to binary. Can you see the simple procedure to get, say, 0101 from 00 and 11? Just *interleave* the bits:

0		0	
	1		1
0	1	0	1

a) Write a MATLAB program to combine two 0|1 row vectors of the same length into a single 0|1 row vector of twice the length, with the bits interleaved, e.g., 10110 and 11001 combine to 1101101001.
b) Write a MATLAB program to reverse this process.
c) If the 4-by-4 example worked above were only 3-by-3 how would the results change? Are the generated addresses still in order? Are they consecutive integers?

15. Four bytes, or 32 bits, constitute a "word" for many computers and provide the basis for locating a memory address in many more. (Usually each byte in computer RAM, "random-access memory", has its own address.) How many addresses are provided by 4 bytes? If each address identifies a byte of memory, how much memory can we have before we run out of addresses? Look up "flash memory" and find the typically largest flash-card memories sold.

16. How many trits are needed to store a 5-digit integer?

17. How many coins must you carry around in order to be able to make exact change on one single occasion for anywhere from 1 cent to 99 cents?
a) Suppose that the only coins in the currency are pennies and dimes.
b) Now allow nickels and 50-cent pieces in the coinage.
c) Now include quarters as well.
d) Now switch to binary coinage: 1-cent, 2-cent, 4-cent, 8-cent, 16-cent, 32-cent and 64-cent pieces.
e) Now switch to ternary coinage: 1-cent, 3-cent, 9-cent, 27-cent and 81-cent pieces.
f) Finally, using ternary coinage, suppose that change-making is shared by both parties to the exchange, so that instead of possibly having to provide, say, two 1-cent pieces, I can give you one 3-cent piece and you give me back one 1-cent piece: how many coins must we *each* carry around to be prepared for one exchange?

18. "Balanced ternary" is an alternative to the base-3 system of Note 3. It is nicely discussed in [Hay08, Ch.10]. Here is how it works.
Instead of the symbol 2 use $\overline{1}$. This means *subtract* the quantity corresponding to the position it appears in. Thus $25 = 27 - 3 + 1$ is written $10\overline{1}1$ instead of 221 for $2 \times 9 + 2 \times 3 + 1$.
a) How many weights of 1g, 3g, 9g, 27g and 81g must we have to weigh exactly any object from 1g to 242g on a balance, if we can put weights on *either* pan of the balance along with the object on one of the pans?
b) How must we adapt the decimal-to-ternary conversion of Note 3 so that it generates balanced ternary?

19. Work out the fractional numbers shown in the figure in Note 5. How would you calculate the base-b representation of a fraction larger than 1, i.e., whose numerator n is larger than its denominator d?

20. Why must the base-b representation repeat with a pattern of no more different symbols than d, the denominator of the fraction? What happens if the fraction is not in lowest terms, i.e., d and n, the numerator, have divisors greater than 1 in common?

21. In Week i the excursion on finding the twenty-five primes less than 100 gives the trick that integers divisible by 3 have digits (base 10) that sum to a multiple of 3. It is also true that numbers divisible by 9 have digits (base 10) that sum to a multiple of 9.
The excursion left open the question of divisibility by 7. We can now address this by working in base 8. Numbers divisible by 7 have octits (base 8) that sum to a multiple of 7 (doing all arithmetic to base 8). Try it! (In fact, this rule works for any integer 1 less than the base it is expressed in: try divisibility by 23 in base 24.)
This rule also works for *divisors* of integers one less than the base they are expressed in. Try it for 3, 5 and 15 to base 16.
There is another rule for integers one larger than the base they are expressed in. In base 10, an integer is divisible by 11 if its digits alternate-sum to zero. For example the alternate sum of the digits of 11 is $1 - 1 = 0$; for 121 is $1 - 2 + 1 = 0$; for 1331 is $1 - 3 + 3 - 1 = 0$. Try this also for 17 in base 16.
Find the fewest bases, b, such that all twenty-five primes less than 100 can be expressed either as $b + 1$ or as a divisor of $b - 1$.

22. Write down the inverse of the genetic code in Note 6. That is, for each amino acid, write all the codons that code for it. Also write the redundancy of the coding for each amino acid, i.e., the number of different codons representing it. The fact that some amino acids have more codons than others suggests that they are more important somehow: explore this.

23. Look up the alternatives to the genetic code discussed in [Hay08, Ch.4] and the history of how scientists tried to figure it out.

24. Explore other bases used in everyday life, such as: base 7 for days of the week; base 8 for the classical western music scale (do, re, mi, .., or A, B, C, ..); base 12 for months of the year, for some currencies (British before 1968: 12 pence in the shilling) and for some number systems (a dozen eggs, Aztec and Maya counting); base 20 for some currencies (British before 1968: 20 shillings in the pound) and for some number systems (Maya); and base 60 for some counting systems (Babylonian).

25. Dentists count teeth in base 8. Humans have 32 teeth, organized in four sets of 8. Find out how dentists identify individual teeth. How would you have to change their notation to do tooth arithmetic? (What might tooth arithmetic mean?) If a dentist looked after a family of four and wanted to label all their teeth without using first names, how might hey do it?

26. "How might Hallowe'en be an octopus's Christmas?" (Thanks The Guardian Weekly 183.22 Nov 12–18 2010, Maslanka puzzles p.42.) Hint: what is the DECimal 25 in OCTal?

27. Like the Fibonacci numbers of Week ii, polynomials can have recurrence relationships. Here are two. Work out some examples and write programs (MATLAB or calculator) to plot them.
a) **Legendre polynomials.** (Plot these from -1 to 1. What is special about these endpoints?)

$$\begin{aligned} L_0(x) &= 1 \\ L_1(x) &= x \\ L_k(x) &= \frac{(2k-1)xL_{k-1}(x) - (k-1)L_{k-2}(x)}{k} \end{aligned}$$

b) **Hermite polynomials.** (In principle these may be plotted from $-\infty$ to ∞.)

$$\begin{aligned} H_0(x) &= 1 \\ H_1(x) &= 2x \\ H_{k+1}(x) &= 2xH_k(x) - 2kH_{k-1}(x) \end{aligned}$$

Variants of the Legendre and Hermite polynomials are developed in an Excursion to Note 4 of Book 9c. The Hermite polynomials also play a role in Note 39 of Book 8c.

28. a) Picking $x = 10$, calculate the value of the polynomial $3x^4 + 2x^3 + 7x^2 + 6x + 8$. This is called "evaluating" the polynomial at the given value of x.
b) Keep track of how many additions and how many multiplications you must do for this evaluation, finding powers of x by doing the appropriate number of multiplications. What would this be for a polynomial in powers of x up to x^k?
c) Compare the costs (essentially, numbers of multiplications) for evaluating this polynomial using the way of writing polynomials in Notes 7, 8 and 9 (also used for bases in Note 4) versus the way of writing in Notes 1, 2, 3 and 5. What do these become for a polynomial in powers of x up to x^k? Which of these ways of writing gives less work for evaluation?

29. **Modular arithmetic.** Apparently similar to arithmetic on different bases is *modular* arithmetic. But it has quite different properties. Here are the addition and multiplication tables for "arithmetic modulo 2".

Modulo 2

$+$	0	1
0	0	1
1	1	0

$\times$	0	1
0	0	0
1	0	1

They almost look like the tables for base-2 arithmetic. But in the addition table there is no carry.

What is the motivation for this? Let's think about adding the days of the week: we'll represent them as 0 for Saturday, 1 for Sunday, 2 for Monday, 3 for Tuesday, 4 for Wednesday, 5 for Thursday and 6 for Friday. If we want to add two days to Tuesday we'll write this as 3 + 2. This gives 5, which is Thursday. So the numbers stand either for the day of the week itself or for the number of days to be added to it: we must *encode* the problem of adding so many days to a certain day, and we must *decode* the result.
What if we add three days to Thursday? 5 + 3 = 1: it must be Sunday. How do we get this answer? Do the usual addition 5 + 3 = 8, then find the *remainder* after dividing by 7, the number of days in the week: 8 modulo 7 = 1.
This introduces the word "modulo", which is just a mathematician's word for "remainder". Here is the addition table modulo 7. You can check that it gets the right days after adding the given number of days to each day of the week.

Modulo 7

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

×	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

You see I've also included the times table. It works in exactly the same way: do the usual multiplication, then divide by 7 (not 6!) and take the remainder. You'll have to invent a meaning for it if you want. We'll just take it as a new kind of arithmetic.
a) What patterns do you see in these tables? Do you see that they are the same if we swap rows and columns? Do you see the different patterns of 0 in the two tables? How about 1? What is the very last entry in the times table?
b) Now make your own tables for modulo 3, modulo 4, modulo 5 and modulo 6.
Look at the 1s in the times tables. Every 1 in the interior of a times table says that the outside numbers in the row and column are *inverses* of each other: the inverse of a number is the number you multiply it by to get 1.
c) Which of these modular arithmetics give every number an inverse? Which do not? If you think you have found a pattern, use it to decide whether modulo 9 has an inverse for every number, and check your answer by working out the times table.

30. Not every polynomial has linear factors, but they all have quadratic factors. Find a simple quadratic polynomial with no linear factors.
    As a matter of fact, all polynomials can be reduced to some mix of (0 or more) linear factors and (0 or more) quadratic factors.

31. What is $(x^k + x^{k-1} + .. + x^2 + x + 1) \times (x - 1)$? What does this tell us about a factor of $x^n - 1$, polynomials which are powers of x, minus 1?

32. What is the remainder of $(3x^5 + 5x^4 + 9x^3 + 13x^2 + 14x + 8) \div (x^2 + 1)$? Find a polynomial to divide $3x^5 + 5x^4 + 9x^3 + 13x^2 + 14x + 8$ by which gives a remainder which is a polynomial in x^3 and smaller powers.

33. Write out polynomials corresponding to the rows of Pascal's triangle (Week ii, Note 6). Find the simplest polynomial factors of any of these polynomials. ("Simplest" polynomials involve x only to the first power. They are called *linear* polynomials, or polynomials of *degree* one.)

34. Work out a procedure, very similar to long division of integers, to divide polynomials. Use it to find $(3x^5 + 5x^4 + 9x^3 + 13x^2 + 14x + 8) \div (x + 1)$ and check your answer against Note 9.

35. a) What are the four linear factors of $x^4 - 2x^3 - x^2 + 2x + 0$, the second polynomial plotted in Note 9? Note that if $x - a$ is a factor then setting $x = a$ in the original polynomial must set the value of the polynomial to zero. So look for the x-values in the plot where the polynomial is zero and then multiply the linear polynomials you get, to check them out.
b) How many divisors does this polynomial have? (Note the distinction between *factors* and *divisors*, as given by the example of the integer 12: its factors are 2, 2 and 3; its divisors are 1, 2, 3, 4, 6 and 12.) How many of these divisors are linear (highest power of x is 1), how many are quadratic (highest power of x is 2), how many are cubic (highest power of x is 3), and how many are quartic (highest power of x is 4)?

36. In Note 9, $-0.736\ 780\ 065\ 530\ 31$ is described as "only an approximation" to the precise expression. How good an approximation is it? Work out how much it could be wrong by, using the terms ".., micro, nano, pico, femto, .." presented in Week ii. Given that the most precisely tested scientific theory to date (general relativity) has agreed with observation to 13 significant figures (and quantum electrodynamics, the second most precisely tested theory, to 11), could such an error be measured at present?

37. Does the polynomial $x^4 + x^3 + x^2 + x + 1$ have any linear factors? Plot it by calculator or by computer program to find out.

38. Why does any polynomial of odd degree (i.e., its highest power of x is an odd integer) have at least one linear factor? (Think about the behaviour of $x^3, x^5, ..$ a) as x becomes a very large positive value and b) as x becomes a very large negative value. And think about the relative sizes of all the lesser powers of x in the polynomial.)

39. Work out the procedure for multiplication of two integers written in base 2, in terms of polynomial multiplication with carrying, along the lines of Note 10.

40. a) On a calculator, or in a programming language, capable of displaying ten or more digits, find 2^{30} (one Giga).
b) Use a calculator, or a programming language, which cannot display as many as ten digits, to help you calculate 2^{30} by finding 2^{15} and its multiples which you can add by hand according to Note 10.

41. Any part of the Preliminary Notes that needs working through.

References

[Hay08] Brian Hayes. *Group Theory in the Bedroom, and Other Mathematical Diversions*. Hill and Wang (Farrar, Straus and Giroux), New York, 2008.