1 | Number bases

1.1 Exercises

An interesting exercise that can be found in [5, p.49] can now be solved with the knowledge of numerical systems: What is 84 equal to, if $8 \times 8 = 54$?

1. Find the polynomial $f(x)$ such that $f(7) = 89183$, where the coefficient a_i is such that $0 \leq a_i < 7$.
   Hint: Try to find the coefficients of the polynomial first. What does a polynomial represent?

2 | Modular arithmetic

An important tool in computer programming is modular arithmetic. It allows us to split up complex problems into smaller parts, find specific information without large calculations, understand mathematical properties and perform operations that have a set pattern. The practical usage of modular arithmetic is found in cryptography – something that surrounds us every day, even if we might not think about it. In this chapter, we will see different kinds of applications of modular arithmetic in computer programming.

2.1 Introduction

In order to understand what modular arithmetic is about, let us consider the way different cultures describe time. In United Kingdom, for instance, the day is split up into two parts and each part is split up into 12 hours. In order to distinguish between these two parts, either AM or PM is used. So 6 AM would be early in the morning, while 6 PM would be close to dinner time, in the evening. In Sweden and some other counties, the day is said to be 24 hours, so 6 AM in the morning would be described as 6 o'clock, while 6 PM would be 18 o'clock. There is clearly some sort of relation between 6 and 18 as well as 7 and 19, namely that these pairs of numbers share a common remainder, when divided by 12. For many people, the *equivalence* between

Why not define a new kind of *day* with a period of 48 hours but keep using the analogue clock in Figure 2.1? Where would the hand be when 32 hours have passed?

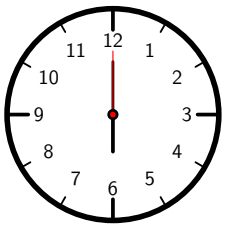

Figure 2.1: An analogue clock. The modulus is 12.

6 and 18 in the analogue clock comes almost intuitively. But, what if we continue to turn the hand three or even hundred times and then want to

know the new position of the hand? This might seem as a lot of calculations, so it would be good to find a pattern when dealing with small numbers to be able to explain what happens when numbers get large enough.

One way of thinking about it is by considering the remainder. The relation between 18 and 6 is that both of them have a remainder of 6. In fact, $30, 42, 54 \ldots 12n + 6$ will also have 6 as their remainder. It turns out that if we are given the number of hours that have elapsed n, by finding the remainder when this value is divided by 12 will tell us where the hand will be. The ability to find the remainder seems quite important, so it is good if there is an easy way it can be obtained.

One of the ways to find the remainder is by converting the fraction $\frac{n}{12}$ to a mixed fraction.

$$\frac{n}{12} = a + \frac{r}{12} \qquad a, r \in \mathbb{Z} \tag{2.1}$$

The r in this equation is the remainder. So, if we want to find where the hand is going to be after 32 hours,

$$\frac{32}{12} = 2 + \frac{\mathbf{8}}{12} \tag{2.2}$$

which means that the remainder is 8. For small numbers, this method works fine, but it is not good at all when the remainder is to be found numerically, using a computer.

In conclusion, we have seen that it is possible to group numbers according to their remainder when divided by a common divisor (we have used 12), which makes it possible to, without performing many calculations, get some information (that is, where the hand will be), even if hundreds of operations have been performed.

2.2 Definition

Now that we have hopefully got some grasp of modular arithmetic, we can start to generalize our findings. Please go through this section quickly first and try to solve some of the problems in (section 2.3), and then return to this section again. It can appear to be difficult in the beginning.

2.2.1 Generalizing 'mod'

Examples of operators are '+','−','×', '/'

You might have seen the mod operator as in $a \bmod c$ or equations as $a \equiv b \pmod{c}$. These are two different notations that refer to the same concept – the remainder. In order to find a definition of mod, we have to look at the way a number can be written. A number n can be expressed as in (2.3) where r is the remainder, q is the quotient, and c is the modulus (in the clock example, it was 12).

$$n = c \times q + r \qquad 0 \leq r < c \tag{2.3}$$

The remainder is expressed as $r = n \bmod c$ using the mod operator. In other words,

$$n = c \times q + n \bmod c$$

The quotient q, on the other hand, is a bit trickier. For that, we will have to introduce another function called *floor*.

The floor function will round a number down, and thus return an integer. By taking floor of a number, we simply remove the fractional part (all the decimals) and keep the whole value that is left. For example, floor of π will be 3 and the floor of e will be 2. Using mathematical notation, floor of a value is expressed using '$\lfloor$' and '$\rfloor$'.

Since $\pi = 3.14\ldots$, $e = 2.72\ldots$

The *quotient* is the whole number of times the divisor can fit inside a specific number. We can use floor to express the the quotient as $q = \lfloor n/c \rfloor$. Now, by substituting everything we know into (2.3) we get:

$$n = c \times \underbrace{\lfloor n/c \rfloor}_{\text{quotient}} + \underbrace{n \bmod c}_{\text{remainder}} \tag{2.4}$$

Using simple algebra, we can obtain a numerical definition of the mod operator that is,

$$n \bmod c = n - c \times \lfloor n/c \rfloor \qquad c \neq 0 \tag{2.5}$$

This definition accepts not only negative integers, but also arbitary real numbers [4, p.82].

This result will be particularly handy when we start to apply mod in computer code. Let us keep it in mind in meantime.

2.2.2 Operations and rules

There are some few things we have to go through before we can start applying the theory into practice. The proofs of some theorems are left as an exercise to the interested reader!

Ways of expressing 'mod'

The *mod* can either be used directly as an operator (as in (2.5)) or in equation systems – congruence equations. We can go from one form to another by the following relation:

$$a \equiv b \pmod{c} \qquad \Longleftrightarrow \qquad a \bmod c = b \bmod c \tag{2.6}$$

We say that 'a is congruent to b modulo c' [4, p.123].

The use of mod operator in equivalence relations emphasises the idea that numbers can be grouped into groups with the same remainder given a modulus. So, if we use the clock where the modulus is 12, the relation between 6 and 18 can be expressed as:

$$18 \equiv 6 \pmod{12}, \qquad \text{since} \quad 18 \bmod 12 = 6 \bmod 12$$

In the clock, there are twelve different groups, $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$ and each group shares a common remainder. Because each group has a

Remainder	General term
0	$12n$
1	$12n+1$
2	$12n+2$
3	$12n+3$
4	$12n+4$
5	$12n+5$
6	$12n+6$
7	$12n+7$
8	$12n+8$
9	$12n+9$
10	$12n+10$
11	$12n+11$

Table 2.1: The remainder and the corresponding general term

common remainder, we can generalize each term as following (by using eq. (2.3)), as can be seen in Table 2.1.

This can of course be done with any other modulus. So, for a modulus c, the clock would look as the one in Figure 2.2. This clock will have c different groups, that is $\{1, 2, 3, \ldots, c-3, c-2, c-1\}$. The general term of

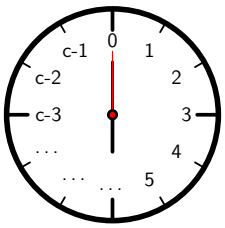

Figure 2.2: A general clock with modulus c

each remainder group is given in Table 2.2. So, if we would take the 2nd group with modulus c, it would be in the form of:

$$-3c+2, \quad -2c+2, \quad \underbrace{-c+2}_{n=-1}, \quad \underbrace{2}_{n=0}, \quad \underbrace{c+2}_{n=1}, \quad 2c+2, \quad 3c+2$$

Note that $n \in \mathbb{Z}$, so it can also be negative. Similarly, using the general term for $c-1$ in Table 2.2 we obtain:

$$-2c-1, \quad -c-1, \quad \underbrace{-1}_{n=-1}, \quad \underbrace{c-1}_{n=0}, \quad \underbrace{2c-1}_{n=1}, \quad 3c-1, \quad 4c-1$$

The important conclusion that should be drawn from the above examples and the general terms in Table 2.2 is that the difference between any two

neighbouring numbers in a remainder group is a multiple of the modulus c. In general,

$$a \equiv b \pmod{c} \iff a - b \text{ is a multiple of } c \tag{2.7}$$

It is often easier to apply (2.7) instead of computing the mod of both a and b. [4, p.124]. It will be useful in exercise 2.

since,

$$cn + r \equiv r \pmod{c} \implies cn + r - r = cn, \quad n \in \mathbb{Z}$$

Remainder	General term
0	cn
1	$cn + 1$
2	$cn + 2$
3	$cn + 3$
$\vdots$	$\vdots$
$c - 3$	$cn + c - 3$
$c - 2$	$cn + c - 2$
$c - 1$	$cn + c - 1$

Table 2.2: The remainder and the corresponding general term when modulus is c. Note that n can be any integer (even negative).

Addition and subtraction

Addition and subtraction are quite obvious if we think about what happens using the clock model. If it is 4 o'clock and we move the hand 5 hours forward, we would end up being at 9 o'clock. However, if we would instead start at 16 o'clock (we should by now know that 16 and 4 have the same remainder) and move 5 hours forward, we would end up at 21 o'clock, which is sure enough equivalent to 9 (if we look at an analogue clock). In fact, even if we would start at 16 (=4 PM) and move 17 (=5 PM) hours, the hand would still point at 9 o'clock. This occurs because every time we pass the 12 mark, we restart. Therefore, the number of times we passed 12 does not matter and the only thing that is of real importance is the remainder group that each numbers we add (or subtract) belongs to. Using mathematical notation, this can be expressed as:

Or at least, using common sense, we should know that they have something in common.

$$(a + b) \bmod c = (a \bmod c + b \bmod c) \bmod c \tag{2.8}$$

Try to plug in some values for a, b, c to see that it works!

or,

$$(a - b) \bmod c = (a \bmod c - b \bmod c) \bmod c \tag{2.9}$$

We have already seen the use of mod in congruence relations, so it would be good to know how addition works in that case. It turns out to be similar, that is, given that:

$$\begin{aligned} a \equiv b \pmod{c} \quad &\Longrightarrow \quad a-b \quad \text{multiple of} \quad c \\ k+a \equiv k+b \pmod{c} \quad &\Longrightarrow \quad (k+a)-(k+b) = a-b \quad \text{multiple of} \quad c \end{aligned}$$

Note that the constant k can be two different numbers too, as long as they are equal to each other congruent modulo c. This can be further generalized to obtain following result:

$$\begin{aligned} a \equiv b \pmod{c} \quad k \equiv j \pmod{c} \quad &\Longrightarrow \quad a+k \equiv b+j \pmod{c} \\ a \equiv b \pmod{c} \quad k \equiv j \pmod{c} \quad &\Longrightarrow \quad a-k \equiv b-j \pmod{c} \end{aligned}$$

The proof is quite similar to the example where only k was used; we apply the relation in (2.7) to both the original expression $a \equiv b$ and $a+k \equiv b+j$. That is, since we know that $a-b$ and $j-k$ are multiples of c from (2.7), the $(a+k)-(j+b)$ should also be a multiple of c, because $(a+k)-(b-j) = (a-b)+(k-j)$, and two multiples (i.e. $(a-b)$ and $(k-j)$) added together form a new multiple. You might have already noticed that if the modulus is kept constant, we do not need to rewrite the '(mod c)' every time.

Multiplication

There are two rules that are going to be described in this section. The first one states that if we would multiply an expression where a modular operation is involved, the constant would affect both the number in front of the mod operator and the modulus as shown in (2.9).

There are so many uses of modular arithmetic, yet there is no proper name for the number in front of the mod operator ...

$$k \times (a \bmod c) = (ka) \bmod (kc) \tag{2.10}$$

If we go back to the definition in (2.5), we can show this as following:

$$\begin{aligned} k \times (a \bmod c) = k \times (a - c\lfloor a/c \rfloor) = \\ = k \times a - k \times c \times \left\lfloor \frac{ka}{kc} \right\rfloor \\ = (ka) \bmod (kc) \end{aligned}$$

□

... maybe we should call it 'modumor', as suggested in [4, p.82]

The second rule is similar to the addition/subtraction rule and can be expressed as following:

$$(a \times b) \bmod c = (a \bmod c \times b \bmod c) \bmod c \tag{2.11}$$

For example,

$$\begin{aligned} \text{LHS} \quad &= \quad (4 \times 5) \bmod 3 = 20 \bmod 3 = 2, \\ \text{RHS} \quad &= \quad (4 \bmod 3 \times 5 \bmod 3) \bmod 3 = (1 \times 2) \bmod 3 = 2 \end{aligned}$$

The general case can be proved by generalising the operations with (2.3), as described in [2].

Let us now translate the multiplication properties to congruence equations. When we combine (2.6) together with what we proved in (2.10) we get that

$$a \equiv b \pmod{c} \qquad \Longleftrightarrow \qquad ka \equiv kb \pmod{kc} \tag{2.12}$$

There is quite a big difference between ' $\Longleftrightarrow$ ' and ' $\Longrightarrow$ ' The first one means that one statement implies the other (we can switch between the forms and it will still work). The second one only says that it works in one direction only.

Note that since multiples of kc are also multiples of c, we can say that

$$ka \equiv kb \pmod{kc} \qquad \Longrightarrow \qquad ka \equiv kb \pmod{c} \tag{2.13}$$

So when (2.12) and (2.13) are combined we get that,

$$a \equiv b \pmod{c} \qquad \Longrightarrow \qquad ka \equiv kb \pmod{c} \tag{2.14}$$

The constant k can be two different numbers here too (as we saw earlier in *Addition and subtraction*), as long as they are congruent to each other. In other words,

$$a \equiv b \pmod{c} \quad k \equiv j \pmod{c} \qquad \Longrightarrow \qquad ka \equiv jb \pmod{c}$$

The proof of this is mentioned in [4, p. 124] and states that $ka - jb = (a-b)k + b(k-j)$. We can explain this as following: Both $(a-b)$ and $(k-j)$ are multiples of c, which can be expressed as $(a-b) = xc$, $(k-j) = yc$. So, by substitution, $(a-b)k + b(k-j) = xkc + byc = c(xk + by)$. But $(xk + by)$ is a new constant, so $(a-b)k + b(k-j)$ is also a multiple of c. It turns out that $(a-b)k + b(k-j) = ak - bk + bk - bj = ka - jb$. So, $ka - jb$ is also a multiple of c, and therefore this multiplications statement is true.

Division

So far, everything we have seen seems to be quite straight forward. Both addition and multiplication properties behave similarly as the ones in arithmetic and are very intuitive. Division rules in modular arithmetic can be a bit trickier though. First, let us start with the ones we already know from the multiplication section. From (2.12) we get:

$$ka \equiv kb \pmod{kc} \qquad \Longleftrightarrow \qquad a \equiv b \pmod{c} \tag{2.15}$$

Note that this works because of the ' $\Longleftrightarrow$ ', which states that we can go in both directions. Now, if we would have done the same reasoning in (2.14), that is, by reversing it (going from right to left), we would end up with something that feels to be right but is not. For example, in $7 \equiv 3 \pmod{4}$, both sides can be multiplied by say 2 so that we get $2 \times 7 \equiv 2 \times 3 \pmod{4}$. The result is $14 \equiv 6 \pmod{4}$. Sure enough, we can divide both sides by 2 again to get back to the original expression. Unfortunately, this is not always

the case. Imagine if we divide $10 \equiv 6 \pmod 4$ by 2, that is $2 \times 5 \equiv 2 \times 3 \pmod 4$. The result is $5 \not\equiv 3 \pmod 3$.

It turns out that if the number that we divide both sides with and the modulus are relatively prime, this division will work.

$$ka \equiv kb \pmod c \quad \iff \quad a \equiv b, \quad k \text{ and } c \text{ are relatively prime} \tag{2.16}$$

Exponent

What about repetead exponent? Check out *Knuth's up-arrow notation*

In arithmetic, exponent is repeated multiplication, as multiplication is repeated addition. One of the properties of modular arithmetic is that when we have performed different kinds of operations such as *addition* and *multiplication*, we end up with a number that only tells us what group the final value belongs to. For example, if our aim is to find where the hand will be in 50 hours, given it is at 9, we don't need to fnd that value by adding up these numbers and then taking mod 12. We can be lazy and take mod of 50 and 9 separately and then add them up, i.e. $(50 + 9) \bmod 12 = (2 + 9) \bmod 12 = 11 \bmod 12 = 11$. In conclusion, we know where the hand is, but we don't know that 59 hours have passed since the the clock started (from 0).

This trick can be used when we compute mod of a number that is raised to the power of another.

$$a^b \bmod c = ((a \bmod c)^b) \bmod c \tag{2.17}$$

For example, if we want to calculate $2^{32} \bmod 12$, we could write this as $(2^8)^4 \bmod 12$, because of the exponent laws. We know that $2^8 = 256$, and that $256 \bmod 12 = 4$. So, by replacing 2^8 with 4 in the original statement we get $4^4 \bmod 12 = 256 \bmod 12 = 4$. Thus, $2^{32} \bmod 12 = 4$.

We should now be more comfortable with the idea behind modular exponents. In the reaming part of this section we are going to look at how we can find the exponent more systematically and finally how this relates to congruent equations.

When we want to calculate the modular exponent, there are two cases to consider with (2.17) in mind. The first case is when b is a power of two and the second one when it isn't.

$b = 2^n, n \in \mathbb{Z}$

Exponent - Powers of 2 Let's say we want to find $5^{256} \bmod 12$ by somehow applying (2.17). Since the exponent is a power of two, we can keep multiplying 2 with itself until we get 256. The idea is to keep taking mod each time we multiply by two so that we don't have to deal with big numbers.

$$\begin{aligned}
&5^1 \bmod 12 = 5 \\
&5^2 \bmod 12 = (5^1 \times 5^1) \bmod 12 = (1 \times 1) \bmod 12 = 1 \\
&5^4 \bmod 12 = (5^2 \times 5^2) \bmod 12 = (1 \times 1) \bmod 12 = 1 \\
&\vdots \\
&5^{256} \bmod 12 = (5^{128} \times 5^{128}) \bmod 12 = 1
\end{aligned}$$

This example leads to two conclusions. The first is that modular arithmetic allows us to examine properties of huge numbers without actually knowing how the real number looks like quite quickly. As an example, 5^{256} contains 179 digits. Secondly, especially in this example, once we get to 1 during an iteration, we can conclude that the final value will be 1 also.

Exponent - Other Powers (not powers of 2) It might seem as if we would need to develop an entirely new method to cover all other powers that are not a power of 2. But why? We already have a working method for all powers of 2, so it would be better to reuse it instead. Let's try to figure out the last digit of 3^{13}; this time, we use a smaller number. Since we know how to find when something is to the power of two, we try to break up the exponent into powers of two. In the case of 13, this can be written as $13 = 2^0 + 2^2 + 2^3 = 1 + 4 + 8$ This is good news since we can rewrite 3^{13} as 3^{1+4+8}. Using the laws of exponents, $3^{1+4+8} = 3^1 \times 3^4 \times 3^8$. Again, this is good since we only need to find the modulus of each individual factor (i.e. 3^1, 3^4, 3^8) separately and then multiply the resulting values together (using multiplication law (2.11)).

A quick way to express a number as a sum of powers of two is to convert that number to base two.

$$\begin{aligned} &3^{13} \bmod 10 = \\ &= 3^{1+4+8} \bmod 10 = \\ &= 3^1 \times 3^4 \times 3^8 \bmod 10 \end{aligned}$$

Using the algorithm to for the powers of two, we get:

$$3^1 \times 3^4 \times 3^8 \bmod 10 = 3 \times 1 \times 1 \bmod 10 = 3$$

2.3 Practical application

Already in section 2.1 we saw how easy it is to find the position of the hand after that it was rotated a large number of times. We should by now start to see that modular arithmetic can be applied to problems where there is a specific pattern involved. Later on, when we begin with basic cryptography, we will be able to see the way it can be used in computer programming.

2.3.1 Number trick and the power of 9

Now after that we have gone through the theory behind modular arithmetic, let us consider a mathematical trick where our knowledge can be applied. Here is how it works: *Say you are to guess the encircled digit. You ask that person to multiply* 45 *by any number and write the result down on paper. You then tell the person to circle one digit in that number and tell you the*

remaining digits in any order. This turns out to be all you need and you finally find the encircled digit. Let us first see which calculation that has to be performed and later why it works.

So, you have asked a person, Bob, to multiply 45 by the number of his choice. Bob will multiply say $\mathbf{75} \times 45$ to get 3375, and then encircle the digit 3. He will now tell you the non encircled digits in any order, for example 375. Now, you need to find the sum of all digits of 375, then find the sum of all digits in the recently found number, and continue doing so until you get to a one digit number. In this case,

The value we get can be referred to as a *digit root/numerical root* [3, p.105]

$$3 + 7 + 5 = 15$$
$$1 + 5 = 6$$

Since we know that the digit sum should add up to 9 (we will see why later), we can solve a simple equation to get the number that was missing (the encircled number), as shown below:

Note that we would not be able to distingusih between $x = 0$ and $x = 9$. A good tip is to tell Bob to avoid encircling 0 (or 9).

$$6 + x = 9$$
$$x = 3$$

This turns out to work, and it would be nice to know why this it is the case.

First, let us clarify the choice of the number 45 and why the digit sum would always adds up to 9. In the beginning of the trick, we could have told Bob to multiply his number by $9, 18, 27, 36, 45, 54, 63 \ldots$ too, the effect would be the same. The numbers above have the same digit sum, that is, 9. From mathematics where divisibility is concerned, we might already be familiar with the fact that all numbers that are divisible by 9 have a digit sum that is divisible by 9 also (see exercise 3 for a detailed proof). This means that if we multiply a multiple of 9 by a constant, the new number will also be a multiple of 9, and so the digit sum will be equal to 9 (this is shown in exercise 4). This is why we know for sure that the new number Bob got will have a digit sum of 9, and so if it is less than 9, we can figure out the digit that was removed.

Now let us look at how we would be able to apply our knowledge of modular arithmetic to find the missing digit. You might have already figured out that the *digit root* (digit sum) method is another way of finding the remainder when the modulus is 9. So, when we were finding the digit root of 375, we could simply calculate

It is good to connect different methods together to be able to pick the easiest and fastest one in different situations.

$$375 - 9 \times \lfloor 375/9 \rfloor$$

which would lead us directly to the answer. But, it can be so much easier to cancel out 9s and work it out by repeated calculation of the sum of digits. A computer, on the other hand, would be able to find the value quite quickly using method in (2.5). In Java Script, this can be expressed as:

```
function mod(n,c)
{
    return (n - c * Math.floor(n / c));
}
```

So, in order to find the digit root, we would use the following code:

```
var number = 375;

var digitRoot = mod(number, 9);

function mod(n,c)
{
    return (n - c * Math.floor(n / c));
}
```

2.3.2 Finding the last digit of a

2.4 Exercises

The exercises below require different solving techniques, so if you are not able to solve a question, please look it up under *Solutions*.

1. What is remainder when $511 \times 421 + 311$ is divided by 19?
2. Find the largest possible value of x, given that $16 \equiv 4 \pmod{x}$
3. Prove, using modular arithmetic (with congruent equations), that if the sum of all digits is divisible by 9, the number itself is divisible by 9 as well.
4. Show that multiplying a number that is a multiple of 9 by constant will result in a number that is also a multiple of 9.
5. Find possible moduli in $\begin{cases} 17 \equiv 43 \\ -6 \equiv 72 \end{cases}$ given that the modulus is the same in both expressions.

2.5 Solutions

1. The answer is 1. Applying the addition and multiplication properties, we get $1 \times 13 + 5 \bmod 17 = 18 \bmod 7 = 1$.
2. This can be solved by applying the relation in (2.7), that is: $16 - 4 = 12 = k \times x$. Since the k is an integer (both positive and negative), the only possible value for k is if it is 1 so that $x = 12$.

3. A number can be represented as $100c + 10b + 1a$, where a, b, c are positive integers. For this number to be divisible by 9, the remainder when it is divided by 9 must be 0. In other words, the number has to be congruent to 0 modulo 9, that is $100c+10b+1a \equiv 0 \pmod 9$. This is equivalent to $1c + 1b + 1a \equiv 0 \pmod 9$ (we use the multiplication property for $100 = 10 \times 10$). This means that if the number is a multiple of 9, the digit sum, $(a + b + c)$, must be a multiple of 9 also, and vice versa.

Might be good to revise the chapter about numerical systems

4. Say A is a multiple of 9, that is $A = 9n, n \in \mathbb{Z}$. By multiplying A by a constant c that is an integer, we get $c \times A = c \times 9n = 9nc$. But nc is a new constant, that is $nc = B$. So, $c \times A = 9B, B \in \mathbb{Z}$, which is also a multiple of 9 (proved in exercise 4).

5. This is a tricky problem since each equation does not have to share the same remainder. We can start by applying (2.7) to find multiples of modulus c

$$43 - 17 = 26$$
$$72 - (-6) = 78$$

Since we know that 26 and 78 are multiples of the same modulus, this can be expressed as:

$$26 = c \times n_1, \qquad \text{where} \quad c \quad \text{is the modulus}$$
$$78 = c \times n_2$$

The ratio between 78 and 26 is $78/26 = 3$, i.e. (n_2/n_1). Now, since we know that constants n_1 and n_2 are integers (using the definition of a multiple), we can try to factorise 26 and 78 (we end with a prime number).

n_2	$c \times n_2$
1	78
2	39
3	26
4	19.5
5	15.6
6	13

n_1	$c \times n_1$
1	26
2	13

We can use these tables to be able to predict the original modulus. Since the ratio should be 3 between n_2/n_1 and the modulus constant, a possible way is to try different combinations that would result in 3 and have the same modulus. Sure enough, it is either 3/1 or 6/2, so the possible moduli are 13 and 26.

[6] [4] [5] [1] [7]

Bibliography

[1] Khan Academy, *Congruence modulo*, https://www.khanacademy.org/computing/computer-science/cryptography/modarithmetic/a/congruence-modulo, Last accessed 2014.05.30.

[2] ———, *Modular multiplication*, https://www.khanacademy.org/computing/computer-science/cryptography/modarithmetic/a/modular-multiplication, Last accessed 2014.06.07.

[3] M. Gardner, *Math wonders and secrets (org. mathematics magic and mistery)*, Nauka, 1986.

[4] Knuth D.E. Potashnik O. Graham R. L., *Concrete mathematics - a foundation for computer science*, Addison-Wesley, USA, 1998.

[5] Pelerman Ya. I., *Zanimatelnaja algebra*, Gosudarstvenoe izdatelstvo tekhniko-teoriticheskoj literatury, Moscow, USSR., 1957.

[6] Ch. N. Rolich, *From 2 to 16*, Vysshaja Shkola, Minsk, USSR., 1981.

[7] Longe B. (translated by Bankrashkova A.V.), *Matematicheskie fokusy (org. the magical math book)*, Astrel, 2006.