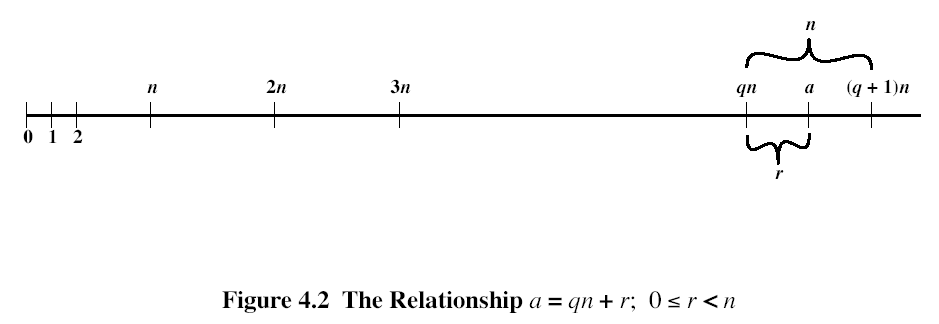
**MODULAR ARITHMETIC**

Given any positive integer n and any integer a, if we divide a by n, we get an integer quotient q and an integer remainder r that obey the following relationship:

a=qn+r 

where  is the largest integer less than or equal to x.



The remainder r is often referred to as a residue.

*a=11, n=7, 11=1x7+4, r=4*

*a=-11, n=7, -11=(-2)x7+3, r=3*

If a is an integer and n is a positive integer, we define a mod n to be the remainder when a is divided by n, Thus, for any integer a, we can always write



11mod7=4, -11mod7=3

Two integers a and b are said to be congruent modulo n, if a mod n = b mod n. This is written as .



**Divisors**

We say that a nonzero b divides a if a=mb for some m, where a, b, and m are integers. That is, b divides a if there is no remainder on division. The notation b|a is commonly used to mean b divides a. Also, if b|a, we say that b is a divisor of a.

The positive divisors of 24 are 1,2,3,4,6,8,12,24.

The following relations hold:

* If a|1 then a=1
* If a|b and b|a, then a=b
* Any b0 divides 0
* If b|g and b|h then b|(mg+nh) for arbitrary integers m and n

To see this last point, note that

If b|g, then g=bg1 for some integer g1

If b|h, then h=bh1 for some integer h1

So

mg+nh=mbg1+nbh1=b(mg1+nh1)

and therefore b divides mg+nh.

*B=7,g=14, h=63, m=3, n=2*

*7|14 and 7|63. To show: 7|(314+263)*

*We have (314+263)=7(32+29)*

*And it is obvious that 7|7(32+29)*

Note that if a0modn, then n|a.

**Properties of the Modulo operator**

1. ab mod n if n|(a-b)
2. ab mod n implies ba mod n
3. ab mod n and bc mod n imply ac mod n

To demonstrate the 1st point, if n|(a-b), then a-b=kn for some k. So we can write a=b+kn. Therefore, (a mod n) (remainder when b+kn is divided by n) = (remainder when b is divided by n) = (b mod n)

*238 mod 5 because 23-8=15=5x3*

*-115 mod 8 because -11-5=-16=8x(-2)*

*810 mod 27 because 81-0=81=27x3*

**Modular arithmetic operations**

Properties of modular arithmetic, working over {0,1,.., n-1}:

1. [(a mod n) + (b mod n)] mod n = (a+b) mod n
2. [(a mod n) - (b mod n)] mod n = (a-b) mod n
3. [(a mod n) x (b mod n)] mod n = (ab) mod n

We demonstrate the 1st property. Define (a mod n) = ra and (b mod n) = rb. Then we can write a=ra+jn for some integer j and b= rb+kn for some integer k. Then

(a+b) mod n = (ra+jn+ rb+kn) mod n = (ra+ rb+(k+j)n) mod n = (ra+ rb) mod n = [(a mod n) + (b mod n)] mod n

*11 mod 8 =3, 15 mod 8 =7*

*[(11 mod 8) + (15 mod 8)] mod 8 = 10 mod 8 =2*

*(11+15) mod 8 = 26 mod 8 = 2*

*[(11 mod 8) - (15 mod 8)] mod 8 = -4 mod 8 =4*

*(11-15) mod 8 = -4 mod 8 = -4*

*[(11 mod 8) x (15 mod 8)] mod 8 = 21 mod 8 =5*

*(11x15) mod 8 = 165 mod 8 = 5*

Exponentiation is performed, as in ordinary arithmetic

*To find 117 mod 13, we can proceed as follows:*



Thus, the rules for ordinary arithmetic involving addition, subtraction, and multiplication carry over into modular arithmetic.

**Modular arithmetic operations (CONT 1)**

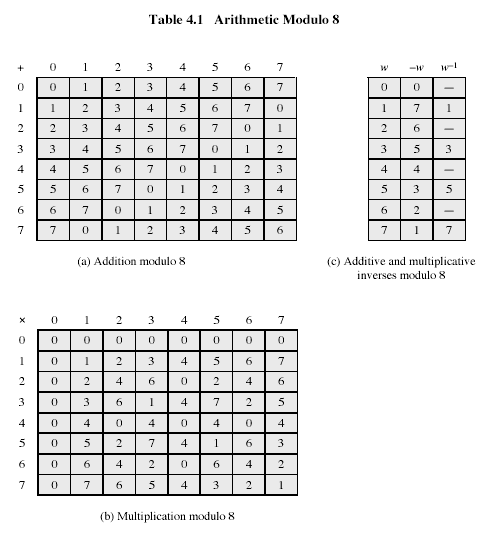


Table 4.1 introduces arithmetic modulo 8. We see that not for all elements exist multiplicative inverses (for 2, 4, 6).

**Euclud’s algorithm**

(3rd century B.C., from Alexandria)

One of the basic techniques of number theory is Euclid’s algorithm, which is a simple procedure for determining the greatest common divisor of two positive numbers.

**Greatest common divisor**

We will use notation gcd(a,b) to mean the greatest common divisor of a and b. The positive integer c is said tob the greatest common divisor of a and b if

1. c is a divisor of a and of b
2. any divisor of a and b is a divisor of c

An equivalent definition is the following:

gcd(a,b)=max[k, such that k|a and k|b]

Because we require that the greatest common divisor be positive, gcd(a,b)=gcd(a,-b)=gcd(-a,b)=gcd(-a,-b). In general, gcd(a,b)=gcd(|a|,|b|).

gcd(60,24)=gcd(60,-24)=12

Also, because all nonzero integers divide 0, we have gcd(a,0)=|a|.

We stated that two integers are relatively prime if their only common positive integer factor is 1. This is equivalent to saying that a and b are relatively prime if gcd(a,b)=1.

8 and 15 are relatively prime because the positive divisors of 8 are 1,2,4, and 8, and the positive divisors of 15 are 1,3,5, and 15, so 1 is the only integer on both lists.

**Finding the greatest common divisor**

Euclid’s algorithm is based on the following theorem: For any nonnegative integer a and any positive integer b,

gcd(a, b)=gcd(b, a mod b) (4.1)

*gcd(55,22)=gcd(22, 55 mod 22) = gcd(22,11) = 11*

To see, that (4.3) works, let d=gcd(a,b). Then, by the definition of gcd, d|a and d|b. For any positive integer b, a can be expressed in the form

a = kb+rr mod b

a mod b = r

with k, r integers. Therefore, (a mod b) = a-kb for some integer k. But because d|b, it also divides kb. We also have d|a. Therefore, d|(a mod b). This shows, that d is a common divisor of b and (a mod b). Conversely, if d is a common divisor of b and (a mod b), then d|kb and thus d|[kb+(a mod b)], which is equivalent to d|a. Thus, the set of common divisors of a and b is equal to the set of common divisors of b and (a mod b). Therefore, the gcd of one pair is the same as the gcd of the other pair, proving the theorem.

*Equation (4.1) can be used repetitively to determine the greatest common divisor:*

*gcd(18,12)=gcd(12,6)=gcd(6,0)=6*

*gcd(11,10)=gcd(10,1)=gcd(1,0)=1*

Euclid’s algorithm makes repeated use of (4.1) to determine the greatest common divisor, as follows. The algorithm assumes a>b>0. It is acceptable to restrict the algorithm to positive integers because gcd(a,b) = gcd(|a|,|b|)

**EUCLID’S ALGORITHM**

EUCLID(a,b)

1. A:=a; B:=b
2. if B=0 return A=gcd(a,b)
3. R=A mod B
4. A:=B
5. B:=R
6. goto 2

The algorithm has the following progression:

A1=B1xQ1+R1

A2=B2xQ2+R2

A3=B3xQ3+R3

*To find gcd(1970,1066)*

*1970=1x1066+904 gcd(1066,904)*

*1066=1x904+162 gcd(904,162)*

*904=5x162+94 gcd(162,94)*

*162=1x94+68 gcd(94,68)*

*94=1x68+26 gcd(68,26)*

*68=2x26+16 gcd(26,16)*

*26=1x16+10 gcd(16,10)*

*16=1x10+6 gcd(10,6)*

*10=1x6+4 gcd(6,4)*

*6=1x4+2 gcd(4,2)*

*4=2x2+0 gcd(2,0)*

*Therefore, gcd(1970,1066)=2*

This process should terminate, otherwise we would get an endless sequence of positive integers, each one is strictly smaller than the one before, and this is clearly impossible.

**FINITE FIELDS OF THE FORM GF(p)**

Finite fields play crucial role in many crypto algorithms. It can be shown that the order of a finite field must be a power of a prime pn, where n is a positive integer. Prime is an integer whose only positive integer factors are itself and 1. The finite field of order pn is usually denoted by GF(pn); GF stands for Galois field in honor of the French mathematician Evarist Galois (1811-1832, <http://scienceworld.wolfram.com/biography/Galois.html> ).

**Finite Fields of Order p**

For a given prime p, GF(p) is defined as the set Zp={0,1,..,p-1} of integers together with arithmetic operations modulo p. For such prime numbers, holds (M7) - Multiplicative inverse axiom.

Because elements w of Zp are relatively prime to p, if we multiply all the elements of Zp by w, the resulting residues are all of elements Zp, permuted. Thus, exactly one of the residues has the value 1, respective multiplier is just the inverse element for w, designated w-1. We can write then:

If  then  (4.2)

Consequence is obtained by multiplication of both parts of (4.2) by a-1.

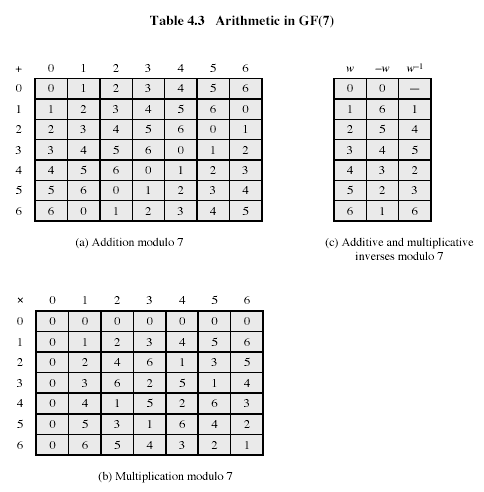
The simplest finite field is GF(2):

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| + | 0 | 1 |  | X | 0 | 1 |  | w | -w | w-1 |
| 0 | 0 | 1 |  | 0 | 0 | 0 |  | 0 | 0 | - |
| 1 | 1 | 0 |  | 1 | 0 | 1 |  | 1 | 1 | 1 |

Addition Multiplication Inverses

**Finite Fields of Order p (CONT 1)**

Next is for GF(7):



**Finding the Multiplicative Inverse in GF(p)**

Table 4.3b may be used to find multiplicative inverse, but for large values of p it is not practical.

If gcd(m,b)=1, then b has a multiplicative inverse modulo m. That is, for positive integer b<m, there exists a b-1<m such that b b-1=1 mod m. Euclid’s algorithm can be extended so that, in addition to finding gcd(m,b), if the gcd is 1, the algorithm returns the multiplicative inverse of b.

**Finding the Multiplicative Inverse in GF(p) (CONT 1)**

EXTENDED EUCLID(m,b)

1. (A1,A2,A3):=(1,0,m); (B1,B2,B3):=(0,1,b);
2. if B3=0 return A3=gcd(m,b); no inverse
3. if B3=1 return B3 = gcd(m,b); B2= b-1 mod m
4. Q=
5. (T1,T2,T3):=(A1-QB1, A2-QB2, A3-QB3)
6. (A1,A2,A3):= (B1,B2,B3)
7. (B1,B2,B3):= (T1,T2,T3)
8. goto 2

Throughout the computation, the following relationships hold:

mT1+bT2=T3 mA1+bA2=A3 mB1+bB2=B3

To see that algorithm correctly returns gcd(m,b), note that if we equate A and B in Euclid’s algorithm with A3 and B3 in the extended Euclid’s algorithm, then the treatment of the two variables is identical. Note also that if gcd(m,b)=1, then on the final step we would have B3=0 and A3 =1. Therefore, on the preceding step, B3=1. But if B3=1, then we can say the following:

mB1+bB2=B3

mB1+bB2=1

bB2=1-mB1

bB21 mod m

Hence, B2 is the multiplicative inverse of b.

Table 4.4 is an example of the execution of the algorithm. It shows that gcd(550,1759)=1 and that the multiplicative inverse of 550 is 355; that is, 550x3551 mod 1759.



**POLYNOMIAL ARITHMETIC**

We are concerned with polynomials in a single variable x, and we can distinguish three classes of polynomial arithmetic:

* Ordinary polynomial arithmetic, using the basic rules of algebra
* Polynomial arithmetic in which the arithmetic on the coefficients is performed modulo p; that is, coefficients are in Zp
* Polynomial arithmetic in which the coefficients are in Zp, and the polynomials are defined modulo a polynomial m(x) whose highest power is some integer n

We consider these variants below.

Ordinary Polynomial Arithmetic

A polynomial of degree n (integer n0) is an expression of the form



where  are elements of some designated set of numbers S, called the coefficient set, and . We say that polynomials are defined over S.

A zeroth-degree is called a constant polynomial and is simply an element of S. An n-th degree polynomial is said to be a monic polynomial if .

In the context of abstract algebra, we are usually not interested in evaluating a polynomial for a particular value of x [e.g., f(7)]. To emphasize this point, the variable x is sometimes referred to as the indeterminate.

Polynomial arithmetic includes the operations of addition, subtraction, and multiplication:



Division is similarly defined, but requires that S be a field. Examples of fields include the real numbers, rational numbers, and Zp for p prime. Note that the set of all integers is not a field and does not support polynomial division.

**Polynomial Arithmetic with Coefficients in Zp**

Within a field, given two elements a and b, the quotient a/b is also an element of the field. However, in general division will result in quotient and remainder; that is, not exact division.

If the coefficient set S is integers, then  does not have a solution, because it would require a coefficient with the value of 5/3, which is not

**Polynomial Arithmetic with Coefficients in Zp (CONT 1)**

in the coefficient set. Suppose, we perform the same polynomial division over Z7. Then we have =4x, which is a valid polynomial over Z7.

However, in general, even if the coefficient set is a field, division will produce quotient and remainder:

 (4.3)

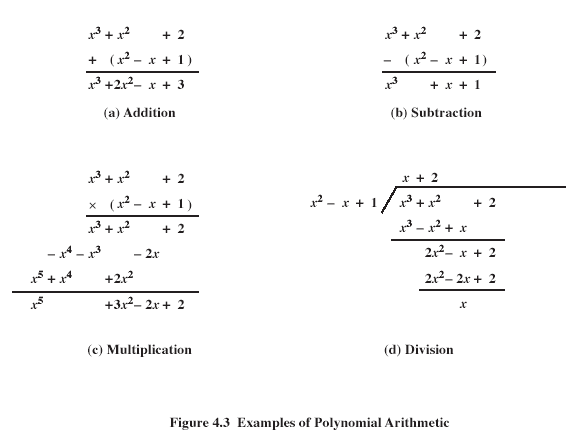
If the degree of f(x) is n and degree of g(x) is m, (), then the degree of the quotient q(x) is n-m and the degree of the remainder r(x) is at most m-1. With the understanding that remainders are allowed, we can say that the polynomial division is possible if the coefficient set is a field.

In an analogy to integer arithmetic, we can write f(x) mod g(x) for the remainder r(x) in (4.3), that is, r(x) = f(x) mod g(x). If remainder r(x)=0, then we say that g(x) divides f(x), written as g(x)|f(x); equivalently, we can say that g(x) is a factor of f(x) or g(x) is a divisor of f(x).

If  produces quotient q(x)=x+2, and remainder r(x)=x, as shown in Fig. 4.3d. This clearly verified by

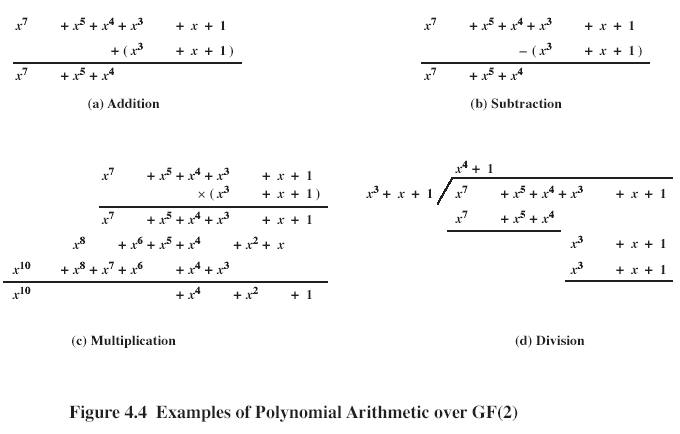


**Polynomial Arithmetic with Coefficients in Zp (CONT 2)**



For our purposes, polynomials over GF(2) are of the most interest. Fig.4.4 shows an example of polynomial arithmetic over GF(2):

**Polynomial Arithmetic with Coefficients in Zp (CONT 3)**



A polynomial f(x) over a field F is called irreducible if and only if f(x) cannot be expressed as a product of two polynomials, both over F, and both of degree lower than that of f(x). By analogy to integers, an irreducible polynomial is also called a prime polynomial.

The polynomial f(x)= over GF(2) is reducible, because .

Consider polynomial f(x)=. It is clear by inspection that x is not a factor of f(x). Also, x+1 is not a factor of f(x). Thus, f(x) has not factors of degree 1. But it is clear, that if f(x) is reducible then it must have one factor of degree 2 and one factor of degree 1. Therefore, f(x) is irreducible.

**Finding the Greatest Common Divisor**

The polynomial c(x) is said to be the greatest common divisor of a(x) and b(x) if

1. c(x) divides both a(x) and b(x)
2. any divisor of a(x) and b(x) is a divisor of c(x)

An equivalent definition: gcd[a(x),b(x)] is the polynomial of maximum degree that divides both a(x) and b(x).

We can adapt Euclid’s algorithm to compute gcd. The equation (4.1) can be rewritten as the following theorem:

gcd[a(x),b(x)]= gcd[b(x), a(x) mod b(x)] (4.4)

Euclid’s algorithm below assumes that the degree of a(x) is greater than the degree of b(x):

EUCLID[a(x),b(x)]

1. A(x):=a(x); B(x):=b(x)
2. if B(x)=0 return A(x)= gcd[a(x),b(x)]
3. R(x):=A(x) mod B(x)
4. A(x):=B(x)
5. B(x):=R(x)
6. goto 2

Find gcd[a(x),b(x)] for a(x)= and b(x)=

A(x)=a(x), B(x)=b(x)

R(x)=A(x) mod B(x) =

A(x)= , B(x)= 

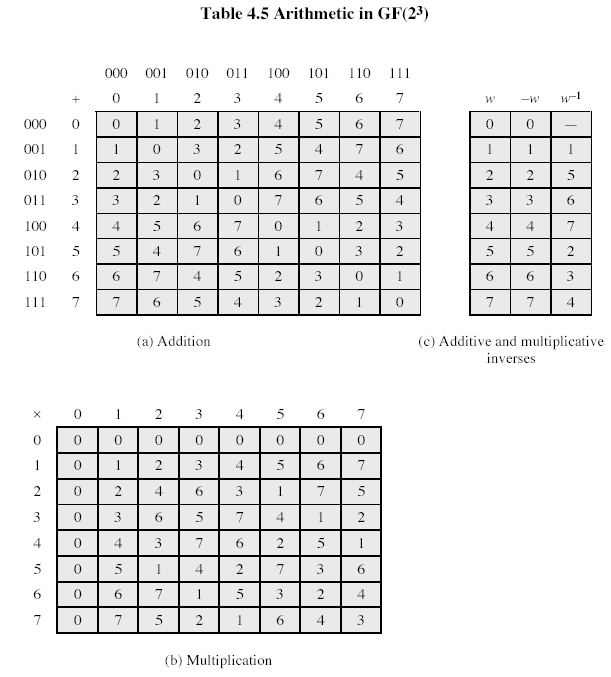
R(x)= A(x) mod B(x) = 0

A(x)= , B(x)=0

**Finding the Greatest Common Divisor (CONT 1)**

gcd[a(x),b(x)]=A(x)= 

**Finite Fields of the Form GF(2n)**



In Table 4.5, arithmetic operations are defined in special way, so, contrary to previously considered case of 8 elements, here we have multiplicative inverses for all non-zero values.

**Modular Polynomial Arithmetic**

Let set S of polynomial coefficients is a finite field Zp, and polynomials have degree from 0 to n-1. There are totally pn different such polynomials. The definition consists of the following elements:

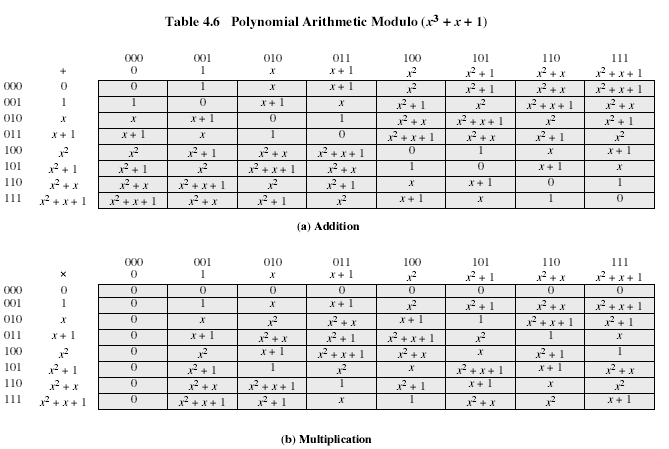
1. Arithmetic follows the ordinary rules of polynomial arithmetic using the basic rules of algebra, with the following two refinements
2. Arithmetic on the coefficients is performed modulo p. That is, we use the arithmetic for the finite field Zp
3. If multiplication results in polynomial of degree greater than n-1, then the polynomial is reduced modulo some irreducible polynomial m(x) of degree n. That is, we divide by m(x) and keep the remainder. For a polynomial f(x), the remainder is expressed as r(x) = f(x) mod m(x)

The AES algorithm uses arithmetic in the finite field GF(28), p=2, n=8, with the irreducible polynomial m(x)=.

It can be shown that the set of all polynomials modulo an irreducible nth degree polynomial m(x) satisfies the axioms in Fig. 4.1 and thus forms a finite field. Furthermore, all finite fields of a given order are isomorphic; that is, any two finite-field structures of a given order have the same structure, but the representation, or labels, of the elements may be different.

To construct the finite field GF(23), we need to choose an irreducible polynomial of degree 3. There are only two such polynomials:  and . Using the latter, Table 4.6 shows the addition and multiplication tables for GF(23):

**Modular Polynomial Arithmetic (CONT 1)**



**Finding the Multiplicative Inverse**

Just as Euclid’s algorithm can be adapted to find gcd of two polynomials, the extended Euclid’s algorithm can be also adapted to find the multiplicative inverse of a polynomial. The algorithm will find multiplicative inverse of b(x) modulo m(x) if the degree of b(x) is less than degree of m(x) and gcd[m(x),b(x)]=1. If m(x) is an irreducible polynomial, then it has no other factor than itself or 1, so that gcd[m(x),b(x)]=1. The algorithm follows:

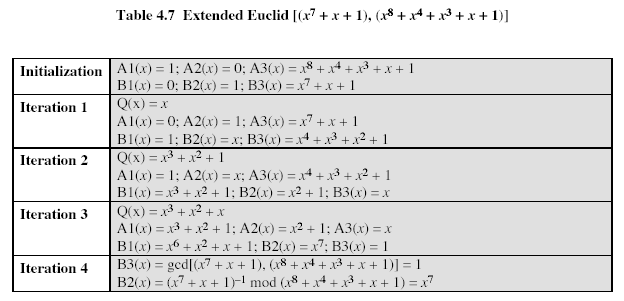
**Finding the Multiplicative Inverse (CONT 1)**

EXTENDED EUCLID[m(x),b(x)]

1. [A1(x), A2(x), A3(x)]:=[1,0,m(x)]; [B1(x), B2(x), B3(x)]:=[0,1,b(x)];
2. if B3(x)=0 return A3(x)= gcd[m(x),b(x)]; no inverse
3. if B3(x)=1 return B3(x)= gcd[m(x),b(x)]; B2(x)=b(x)-1 mod m(x)
4. Q(x):= quotient of A3(x)/B3(x)
5. [T1(x), T2(x), T3(x)]:= [A1(x)-QB1(x), A2(x) –QB2(x), A3(x) –QB3(x)]
6. [A1(x), A2(x), A3(x)]:= [B1(x), B2(x), B3(x)]
7. [B1(x), B2(x), B3(x)]:= [T1(x), T2(x), T3(x)]
8. goto 2

Table 4.7 shows the calculation of the multiplicative inverse of  mod . The result is that  = . That is

()()1 mod .



To get Table 4.5 a,b from Table 4.6 a,b it is sufficient to replace polynomials expressed as ordinary formulae by their codes (by sets of respective coefficients, for example,  - by 101).