**Lecture 1: Integer Division**

**LEARNING OUTCOME**

**By the end of the lesson the student will be able to:**

a) compute a gcd between two positive integers via Euclidean algorithm.

b) compute an inverse of a modulo b via an extended Euclidean algorithm.

Suppose 0 < *a* < *b* for some integers *a* and *b*. Then there is a unique pair of integers *q* and *r* such that

*b* = *a* ⋅ *q* + *r*

where 0 ≤ *r* < *a*, or *r* = *b* mod *a*, and *q* = .

Technically, *b* is divided by *a*, however, a proper mathematical expression,

we want to say, *a* divides *b* (with zero remainder) …

The number *q* is called the quotient and *r* is called the remainder. Once we know a remainder *r*, the quotient *q* is implicitly known.

For instance, take *b* = 23 and *a* = 7. Here,

23 = 3 ⋅ 7 + 2,

so *q* = 3 and *r* = 2. Observe also that the restriction that the remainder *r* lies in the range 0 ≤ *r* ≤ *a* −1 is essential for uniqueness. For example, it is true that 23 = 2 ⋅ 7 + 9, but we cannot use *r* = 9 as a remainder because it is larger than the divisor 7; given *b* = 23, *a* = 7, the only values of *q* and *r* satisfying 23 = 7*q* + *r*, 0 ≤ *r* < 7 are 3 and 2, respectively.

**Modular Ring**

Here are the addition and multiplication tables for modulo 7 arithmetic:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| +7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |  | ⋅7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 0 |  | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 2 | 2 | 3 | 4 | 5 | 6 | 0 | 1 |  | 2 | 0 | 2 | 4 | 6 | 1 | 3 | 5 |
| 3 | 3 | 4 | 5 | 6 | 0 | 1 | 2 |  | 3 | 0 | 3 | 6 | 2 | 5 | 1 | 4 |
| 4 | 4 | 5 | 6 | 0 | 1 | 2 | 3 |  | 4 | 0 | 4 | 1 | 5 | 2 | 6 | 3 |
| 5 | 5 | 6 | 0 | 1 | 2 | 3 | 4 |  | 5 | 0 | 5 | 3 | 1 | 6 | 4 | 2 |
| 6 | 6 | 0 | 1 | 2 | 3 | 4 | 5 |  | 6 | 0 | 6 | 5 | 4 | 3 | 2 | 1 |

A ring comes with an addition and multiplication. A ring is closed under addition and multiplication operations. An identity under addition is zero while an identity under multiplication is one.

Let R = ({0, 1 , … , 6}, +7, ⋅7) be a ring.

**Primes**: A positive integer greater than 1 which is not divisible by a smaller positive integer but 1 and itself.

The first 25 prime numbers (all the prime numbers less than 100) are *p*0, *p*1, *p*2, *p*3, …, *p*25

= 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97.

**Greatest Common Divisor**(gcd)

Greatest common divisor gcd(*m*, *n*) is the greatest number that divides both *m* and *n*. It is a common factor with minimum powers in their prime number decomposition.

Let *n* = 24 ⋅ 32 ⋅ 53 ⋅ 7 = 126,000 and *m* = 26 ⋅ 31 ⋅ 54 ⋅ 7 = 840,000 then

gcd(*m*, *n*) = 24 ⋅ 31 ⋅ 53 ⋅ 7 = 42,000.

*b* = *a* ⋅ *q* + *r*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Algorithm 0: Euclidean Algorithm**

To compute a greatest common divisor between *a* and *b*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

function gcd(*a*, *b*)

*r* = *a*.

while *r* > 0 do

*r* ← *b* mod *a*, 

*b* ← *a*

*a* ← *r*

return *a*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Take a simple example *a*=42 and *b*=283.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *i* | *b* = | *a* | *q* | + *r* |
| 0 | 283 | 42 | 6 | 31 |
| 1 | 42 | 31 | 1 | 11 |
| 2 | 31 | 11 | 2 | 9 |
| 3 | 11 | 9 | 1 | 2 |
| 4 | 9 | 2 | 4 | 1 |
| 5 | 2 | 1 | 2 | 0 |

Since *b*=283 is prime, gcd(*a*, *b*) =1. An *a* and *b* do not share any prime factors. They are relatively prime.

Let us move on to an extended Euclidean algorithm in order to compute an inverse of *a* mod *b*. We will extend and add variable *u*, *v* and *w* on the right;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Algorithm 1:** Extended Euclidean Algorithm

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

function exgcd(*a*, *b*)

*u* ← 0, *v* ← 1.

while *a* > 0 do

*r* ← *b* mod *a*, , *w* ← *u* – *v* ⋅ *q*,

*b* ← *a*, *u* ← *v*,

*a* ← *r*, *v* ← *w*.

return *v*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Table 1: A sample computation of 42−1 mod 283.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *i* | *b* = | *a* | *q* | + *r* | *u* | *v* | *w*=*u**v**q* |
| 0 | 283 | 42 | 6 | 31 | 0 | 1 | -6 |
| 1 | 42 | 31 | 1 | 11 | 1 | -6 | 7 |
| 2 | 31 | 11 | 2 | 9 | -6 | 7 | -20 |
| 3 | 11 | 9 | 1 | 2 | 7 | -20 | 27 |
| 4 | 9 | 2 | 4 | 1 | -20 | 27 | -128 |
| 5 | 2 | 1 | 2 | 0 | 27 | -128 | 283 |

Since an element of this ring 0 ≤ *a*−1 < 283, we will take *a*−1 = −128+283= 155.

In order to check that our answer is correct, we will compute

*a*⋅*a*−1 ≡1 (mod 283)

42⋅155 = 6510 = 23⋅283+1 ≡ 1 (mod 283).

Take a simple exercise Tutorial 1

1. Take an *a* = 1000 + *x* where *x* = your matrix id number mod 1000(the last 3 digit of your matrix id). X = 039 a = 1039
2. Take a prime number, *b* = 4001.
3. Compute an inverse *a*−1 mod *b*. = 2349

You can get an overview on this topic on youtube at

1. <https://www.youtube.com/watch?v=B1fZBLRdvso>
2. <https://www.youtube.com/watch?v=fz1vxq5ts5I>

**Sieve of Eratosthene**

How to generate prime list?

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |

Starts from 1, then first prime is 2. Delete a factor of 2.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |

Then we will walk on the next prime is 3. Delete a factor of 3.

Then we will walk on the next prime is 5. Delete a factor of 5.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |

In mathematics, the sieve of Eratosthenes, one of a number of prime number sieves, is a simple, ancient algorithm for finding all prime numbers up to any given limit. Let *n* =100. Suppose we want to collect the prime numbers less than 100. First, we will list down the numbers 2, …, 100.

Second, we identify the first/initial prime *p*0 = 2. Then we erase multiples of 2.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 |  |

Then we walk forward, from 2 to 3. Set *p*1 = 3. Then we erase multiples of 3.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 |  |

Then we walk forward, from 3 to 5. Set *p*2 = 5. Then we erase multiples of 5.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 |  |

Then we walk forward, from 5 to 7. Set *p*3 = 7. Then we erase multiples of 7.

Then we walk forward, from 7 to 11. Set *p*4 = 11. Then we erase multiples of 11.

The primes less than 100 must less than =10.

Then we walk forward, from 5 to 7. Set *p*3 = 7. Then we erase multiples of 7.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 |  |

Then we walk forward, from 7 to 11. Set *p*4 = 11. Since , we need to check/sieve a prime less than 10. Whatever left must be primes. They are 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 61, 67, 71, 73, 79, 83, 89 and 97.