**Молдавский государственный университет**

**Факультет математики и информатики**

**Департамент Информатики**

**УПРАЖНЕНИЕ 1**

**Возведение в степень по модулю,**

**Нахождение обратного по модулю,**

**Операции сложения и умножения в поле Галуа**

**Выполнил**: Гуцу Даниил.

**Группа:** IA2102

**Проверила:** др.,Чербу О.

Кишинев, 2023

**КИШИНЕВ – 2023**

# Алгоритм Exponentierea modulară

**Algoritmul:**

z=1;

**for** i =s-1 **downto** 0 **do**

z= z2 **mod** n

**if** ci =1 **then** z=z\*x **mod** n

**[Prelegerea 10 RSA] pag 3**

## **Exemplul practic:**

2311^1571mod 2501= 152

**c** = 1571

**x** = 2311

**n** = 2501

(1571)10 = (11000100011)2 = Ci

|  |  |  |
| --- | --- | --- |
| i | ci | Z=1 |
| 10 | **1** | **z= z2 mod n** = 12 mod 2501=1 \* 2311 mod 2501 = **2311** |
| 9 | **1** | **z= z2 mod n** = 23112 mod 2501 = 1086 \* 2311 mod 2501 = **1243** |
| 8 | **0** | **z= z2 mod n** = 12432 mod 2501 = **1932** |
| 7 | 0 | **z= z2 mod n** 19322 mod 2501 = **1132** |
| 6 | 0 | **z= z2 mod n** 11322 mod 2501 = **912** |
| 5 | 1 | **z= z2 mod n** 9122 mod 2501 = 1412 \* 2311 mod 2501 = **1828** |
| 4 | 0 | **z= z2 mod n** 18282 mod 2501 = **248** |
| 3 | 0 | **z= z2 mod n** 2482 mod 2501 = **1480** |
| 2 | 0 | **z= z2 mod n** 14802 mod 2501 = **2025** |
| 1 | 1 | **z= z2 mod n** 20252 mod 2501 = 1486 \* 2311 mod 2501 = **273** |
| 0 | 1 | **z= z2 mod n** 2732 mod 2501 = 2000 \* 2311 mod 2501 = **152** |

## Код программы:

import java.util.Scanner;  
  
  
public class BigPow {  
  
 public static void main(String[] args) {  
 System.*out*.println("Grad mare cu modulo");  
 System.*out*.println("a ^ k mod n (unde k - numar mare)");  
 System.*out*.println("Introduce a k n: ");  
  
 long a = *readLong*();  
 long k = *readLong*();  
 long n = *readLong*();  
  
 System.*out*.println("Resultat: " + *pow\_mod*(a, k, n));  
 }  
  
 */\*\* a ^ k mod n \*/* public static long pow\_mod(long a, long k, long n) {  
 long b = 1;  
 while (k > 0) {  
 if (k % 2 == 0) {  
 k /= 2;  
 a = (a \* a) % n;  
 } else {  
 k--;  
 b = (b \* a) % n;  
 }  
 System.*out*.println(b);  
 }  
 return b;  
 }  
  
 public static long readLong() {  
 return new Scanner(System.*in*).nextLong();  
 }  
}

### Вывод программы:

Grad mare cu modulo

a ^ k mod n (unde k - numar mare)

Introduce a k n:

2311

1571

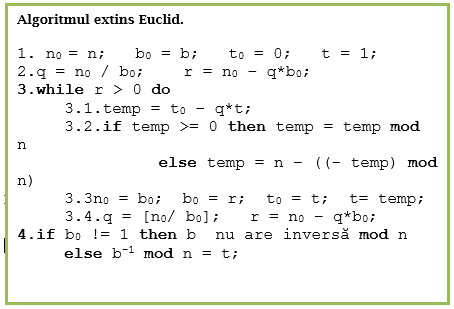
2501

Resultat: 152

# **Algoritmul extins Euclid.**

## **Algoritmul utilizat pentru calculul inversei unui numar mare modulo (b^(-1)mod n)**

## **Exemplul practic nr. 1:**



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | n0 | b0 | q | r | t0 | t | temp |
| 1 | 2699 | 1777 | 1 | 922 | 0 | 1 | 2698 |
| 2 | 5177 | 922 | 2 | 855 | 1 | 2698 | 2 |
| 3 | 922 | 855 | 1 | 67 | 2698 | 2 | 2696 |
| 4 | 855 | 67 | 12 | 51 | 2 | 2696 | 38 |
| 5 | 67 | 51 | 1 | 16 | 2696 | 38 | 2658 |
| 6 | 51 | 16 | 3 | 3 | 38 | 2658 | 161 |
| 7 | 16 | 3 | 5 | 1 | 2658 | 161 | **1853** |
| 8 | 3 | 1 | 3 | 0 | 161 | 1853 |  |

## Код программы:

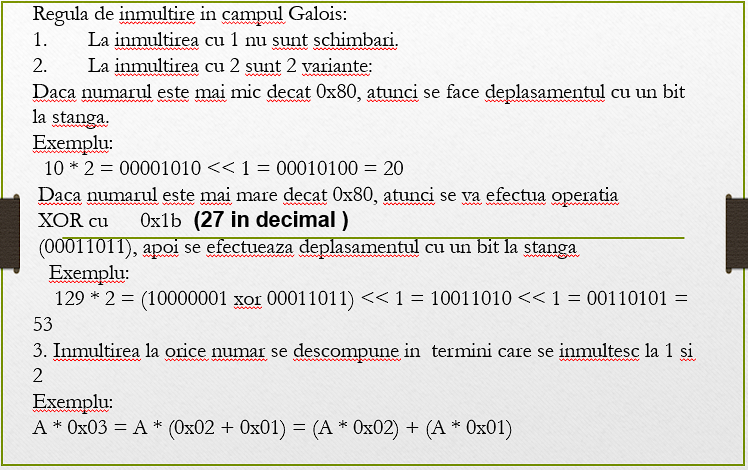
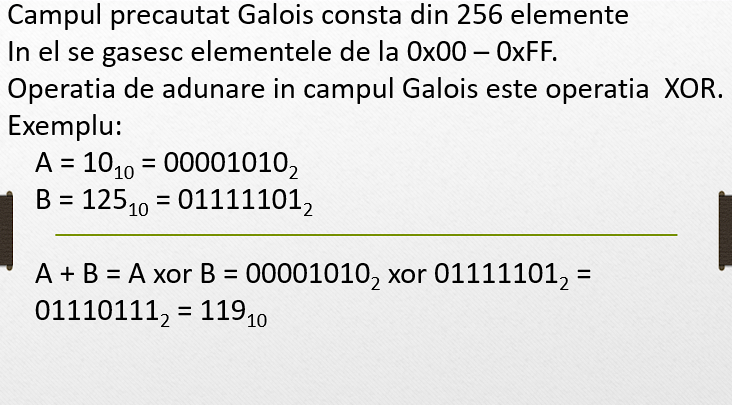
public class InvertModulo {  
 public static void main(String[] args) {  
 final int a = 4127;  
 final int m = 1303;  
 // Before calling computeModularInverse, check if a and m are valid.  
 if (a <= 0 || m <= 0) {  
 throw new IllegalArgumentException("Invalid values: a and m should be positive.");  
 }  
 try {  
 final int inv = *computeModularInverse*(a, m);  
 System.*out*.println(a + "^(-1) mod " + m + " = " + inv);  
 } catch (ArithmeticException e) {  
 System.*out*.println("No modular inverse exists.");  
 }  
 }  
  
 public static int computeModularInverse(int a, int m) {  
 int m0 = m;  
 int y = 0, x = 1;  
 if (m == 1)  
 return 0;  
 while (a > 1) {  
 if (m == 0) throw new ArithmeticException("Modulus cannot be zero");  
 int quotient = a / m;  
 int t = m;  
 m = a % m;  
 a = t;  
 t = y;  
 y = x - quotient \* y;  
 x = t;  
 }  
 if (x < 0) x += m0;  
 return x;  
 }  
}

## Вывод программы:

1777^(-1) mod 2699 = 1853

# Операции в поле Галуа

## Теоретическая часть



## Код программы:

import java.util.InputMismatchException;  
import java.util.Scanner;  
  
public class GaloisField {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 try {  
 System.*out*.println("Enter two numbers: ");  
 int a = scanner.nextInt();  
 int b = scanner.nextInt();  
 System.*out*.println("a add b = " + *add*(a, b));  
 System.*out*.println("Double of 10 = " + *doubleValue*(10));  
 System.*out*.println("a multiply b = " + *multiply*(a, b));  
 } catch (InputMismatchException e) {  
 System.*out*.println("Invalid input. Please enter integers only.");  
 }  
 }  
  
 public static int add(int a, int b) {  
 return (a ^ b) & 0xFF;  
 }  
  
 public static int multiply(int a, int b) {  
 int result = (b & 1) == 1 ? a : 0;  
 while (b > 1) {  
 a = *doubleValue*(a);  
 b >>= 1;  
 result = ((b & 1) == 1) ? *add*(a, result) : result;  
 }  
 return result;  
 }  
  
 private static int doubleValue(int a) {  
 return (a < 0x80) ? (a << 1) : *add*((a << 1), 0x1b) ^ 1;  
 }  
}

## Вывод программы:

Enter two numbers:

23

129

a add b = 150

Double of 10 = 20

a multiply b = 105