Assignment 4. Math

1. $11^7 \text{mod} 13$ =
2. GCD(8376238,1921023) =
3. GCD(102947526,239821932)=
4. 8-bit related to the polynomial $x^5 + x^2 + x$ =
5. 8-bit related to the polynomial $x^6 + x^5 + x^2 + x + 1$ =

1. 2
2. 13
3. 6
4. 00100110
5. 01100111

Task 1

This exercise is about modular arithmetic.

a) Let a = $3^5 \cdot 5^4 \cdot 8^3 \cdot 15^2$ and b = $3^2 \cdot 6^3 \cdot 7^4 \cdot 11$. Compute gcd(a, b) and lcm(a, b).

b) Compute 7^{1234} mod 11.

a) Let's transform equations a bit (decompose a and b into prime numbers):

$$a = 3^7 * 5^6 * 2^9$$
$$b = 3^5 * 2^3 * 7^4 * 11$$

Now we can take the common part which is gcd

$$gcd(a, b) = 2^3 * 3^5 = 1944$$

If we know the greatest common divisor (GCD) of integers a and b, we can calculate the LCM using the following formula.

$$\text{LCM}(a, b) = \frac{a \times b}{\text{GCD}(a, b)}$$

LCM(a,b) = 1944

b) 462086856000000

Task 2

Use the extended Euclidean algorithm to express gcd(26, 91) as a linear combination of 26 and 91.

Step	Pair	Reminder	Linear Comb
0		91	1*91+0*26=91
1	26, 91	26	0*91+1*26=26
2	91, 26	13	1*91 + -3*26=13
3	26, 13	0 - oh, we found the GCD during the step before this one, let's return the coefficients of previous linear combination	-2*91+7*26=0

Task 3

- Describe the next five states of the LFSR if it is initialized according to the box above. The first successor state is already given as illustration, so you have to give the four subsequent ones.

  01110110

  11101100

  11011001

  10110011

- Also do a "rollback" and compute the previous four states, starting from the initial state.

  10101110

11010111

01101011

10110101

- Assume you know that the LFSR is in the initial state given above. After four shifts you intercept 1111 as resulting ciphertext. Reconstruct the 4 bits of plaintext that were encrypted to this ciphertext.

  1010

Tools: python and brains were used.
Python usage:

```
from math import gcd

print(f'1) {11**7%13}')
print(f'2) {gcd(8376238,1921023)}')
print(f'3) {gcd(102947526,239821932)}')
```

```
>>> (2**9*3**7*5**6*3**2*6**3*7**4*11)/1944
462086856000000.0
```

```
>>> 2**3*3**5
1944
```

```
>>> 7**1234%11
3
```

Other tools designed for this lab (scripts for tasks 3 and 4) can be found here: https://github.com/BananaAndBread/FCS/tree/master/Lab4