Question 1 (25 points)
Compute $10^1^{480000023} \mod 35$. Show all your work. Do not use Chinese remainder theorem but other tricks. The simpler the computation the more credit you will get.

$$
\Phi(35) = (5-1)(7-1) = 24 \\
101 \equiv 31 \pmod{35} \quad 4800000023 \equiv 23 \pmod{24} \\
31^{4800000023} \equiv 31^{23} \pmod{35} = 26
$$

Question 2 (25 points)
Using the basic form of Euclid’s algorithm, compute the greatest common divisor of
a. 7469 and 2464,
b. 2689 and 4001,

a. $7469 = 2463 \times 3 + 77$ \\
   $2463 = 77 \times 32$ \\
   $\Rightarrow \text{gcd}(7469, 2463) = 77$

b. $4001 = 2689 \times 1 + 1312$ \\
   $2689 = 1312 \times 2 + 65$ \\
   $1312 = 65 \times 20 + 12$ \\
   $65 = 12 \times 5 + 5$ \\
   $12 = 5 \times 2 + 2$ \\
   $5 = 2 \times 2 + 1$ \\
   $2 = 1 \times 2 + 0$ \\
   $\Rightarrow \text{gcd}(4001, 2689) = 1$
Question 3 (25 points)
One important property which makes DES secure is that the S-boxes are non-linear. How would you verify (not prove of course) the non-linearity of S-box 1 of DES using the following input pairs?

a. \(x_1 = 000000;\) \(x_2 = 000001\)

b. \(x_1 = 111111;\) \(x_2 = 100000\)

S-box 1 of DES

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>14</td>
<td>4</td>
<td>13</td>
<td>1</td>
<td>2</td>
<td>15</td>
<td>11</td>
<td>8</td>
<td>3</td>
<td>10</td>
<td>6</td>
<td>12</td>
<td>5</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>15</td>
<td>7</td>
<td>4</td>
<td>14</td>
<td>2</td>
<td>13</td>
<td>1</td>
<td>10</td>
<td>6</td>
<td>12</td>
<td>11</td>
<td>9</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1</td>
<td>14</td>
<td>8</td>
<td>13</td>
<td>6</td>
<td>2</td>
<td>11</td>
<td>15</td>
<td>12</td>
<td>9</td>
<td>7</td>
<td>3</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>12</td>
<td>8</td>
<td>2</td>
<td>4</td>
<td>9</td>
<td>1</td>
<td>7</td>
<td>5</td>
<td>11</td>
<td>3</td>
<td>14</td>
<td>10</td>
<td>0</td>
<td>6</td>
</tr>
</tbody>
</table>

Non-linearity condition

\[ S_1(x_1 + x_2) \neq S_1(x_1) + S_1(x_2) \]

a. \(S_1(000000) = 14\) and \(S_1(000001) = 0\)

\[ S_1(000000 + 000001) = S_1(000001) = 0 \neq S_1(000000) + S_1(000001) = 14 \]

b. Similar
Question 4. (25 points)

a. Draw the LFSR (linear feedback shift register) corresponding binary polynomial $x^5+x^2+1$ and find the first period of the sequence generated by this LFSR with initialization vector (01011).

b. Is this sequence a maximum-length sequence? State the condition for generating maximum-length sequences.

c. Do the same for the following binary polynomials:
   $x^5+x^3+1$ and $x^5+x+1$

Do they generate maximum-length sequences?

d. Suppose that you are a cryptanalyst and you want to device an attack to break the encryption algorithm of your enemy. Your enemy is not knowledgeable in the area of cryptography (since s/he has not taken any crypto course) and you are almost sure that he is using an LFSR to encrypt her/his data. Please explain what kind of attack method would you employ in order to break her/his code. What kind of data and how much of it would it be needed to implement your attack. Please explain your assumptions.

Output sequence: 110100100010111100011111100  (maximum length)

Output sequence: 110101000010010110011110001101  (maximum length)

Output sequence: 110101001100010000111  (not maximum length)

d. First, I have to be able to apply known-plaintext attack. Using plaintext and ciphertext I obtain the key stream that must be sufficiently long. If the opponent is using an LFSR of length $L$ then I have to obtain key stream of $2L$. Then I apply Berlekamp-Massey algorithm and construct the LFSR.
Question 5 (25 points)
Let \( X \equiv 7^{-1} \mod 15 \)

a. Compute \( X \) using the extended Euclidean algorithm.
b. Compute \( X \) utilizing the Euler’s theorem.
c. Compute \( X \) utilizing the Chinese remainder theorem.

Extended Euclidean algorithm (EEA)
INPUT : Two non-negative integers \( a \) and \( b \) with \( a \geq b \)
OUTPUT : \( d = \gcd(a, b) \) and integers \( x \) and \( y \) s.t. \( a \cdot x + b \cdot y = d \).

1. If \( b = 0 \) then \( d = a, x = 1, y = 0 \) and return \((d, x, y)\).
2. \( x_2 = 1, x_1 = 0, y_2 = 0, y_1 = 1 \).
3. While \( b > 0 \) do the following:
   4. \( q = \lfloor a / b \rfloor, r = a - qb, x = x_2 - qx_1, y = y_2 - qy_1 \)
   5. \( a = b, b = r, x_2 = x_1, x_1 = x, y_2 = y_1, y_1 = y \)
   6. Set \( d = a, x = x_2, y = y_2 \), and return \((d, x, y)\).

\[ 
\begin{array}{cccccccc}
  q & r & x & y & a & b & x_2 & x_1 & y_2 & y_1 \\
  - & - & - & - & 15 & 7 & 1 & 0 & 0 & 1 \\
  2 & 1 & 1 & -2 & 7 & 1 & 0 & 1 & 1 & -2 \\
  7 & 0 & -7 & 15 & 1 & 0 & 1 & -7 & -2 & 15 \\
\end{array}
\]

\( 7^{-1} \mod 15 = -2 \equiv 13 \mod 15 \)

b. \( 7^{-1} \mod 15 = 7^7 \mod 15 \equiv 13 \mod 15 \)

c. \( 7^{-1} \equiv 3 \mod 5 \)
\( 7^{-1} \equiv 1 \mod 3 \)

\( 1 \times 15/3 \times 2 + 3 \times 15/5 \times 2 = 28 \equiv 13 \mod 15 \)