Problem 1. (basic requirements for cryptographic hash functions) Using a block cipher $E_K(x)$ with block length $k$ and key $K$, a hash function $h(m)$ is provided in the following way:

Append $m$ with zero bits until it is a multiple of $k$, divide $m$ into $n$ blocks of $k$ bits each.

$c \leftarrow E_{m_0}(m_0)$

for $i$ in $1..(n-1)$ do

$d \leftarrow E_{m_i}(m_i)$

$c \leftarrow c \oplus d$

end for

$h(m) \leftarrow c$

a) Does this function fulfill the basic requirements for a cryptographic hash function?
b) Can these requirements be fulfilled by replacing the operation XOR ($\oplus$) by AND ($\odot$)?

Problem 2. (proof of Example 10.2) Complete the proof of Example 10.2 from the lecture notes. Show that from

$$k(x_1 - x'_1) \equiv x'_0 - x_0 \mod (p - 1)$$

the discrete logarithm $k = \log_a(b) \mod p$ can be efficiently computed.

Problem 3. (Collision in hash functions) Consider the following function:

$$h : \{0,1\}^* \rightarrow \{0,1\}^*, \; k \rightarrow \left( \left\lfloor \frac{10000(k_{10}(1 + \sqrt{5})/2 - \lfloor (k_{10}(1 + \sqrt{5})/2 \rfloor)}{2} \right\rfloor \right).$$

Here, $\lfloor x \rfloor$ is the floor function of $x$ (round down to the next integer smaller than $x$). For computing $h(k)$, the bitstring $k$ is identified with the positive integer it represents. The result is then converted to binary representation.

(example: $k = 10011$, $k_{10} = 19$, $h(k) = (7426)_2 = 1110100000010$)

a) Determine the maximal length of the output of $h$.
b) Give a collision for $h$. 