Q-1-) Users A and B use the Diffie-Hellman key exchange technique with a common prime $q = 71$ and a primitive root $\alpha = 7$.
a) If user A has a private key XA = 5, what is A's public key YA?
YA = alpha^XA mod q = 7^5 mod 71 = 51

b) If user B has a private key XB = 12, what is B's public key YB?
YB = alpha^XB mod q = 7^12 mod 71 = 4

c) What is the shared secret key?
Shared key is alpha^XA^XB mod q = 7^5^12 mod 71 = 23

d) In the Diffie-Hellman protocol, each participant selects a secret number x and sends the other participant (α^x mod q) for some public number α. What would happen if the participants sent each other (x^α mod q) instead?

Q-2-) A network resource X is prepared to sign a message by appending the appropriate 64-bit hash code and encrypting that hash code with X's private key as described in class (also in the textbook, Page 330.
a) Describe the Birthday Attack where an attacker receives a valid signature for his fraudulent message?
The attacker generates n hashes for valid messages and n hashes for fraudulent messages, until they find a match where a valid and fraudulent message have the same hash. Then they have X sign the valid message and use its signature with their fraudulent message, which will validate for it as well because they have the same hash. Then they have a signed fraudulent message.

b) How much memory space does attacker need for an M-bit message?

As per the slides you need to generate 2^(64/2) valid and fraudulent hashes, so that's 2^33 hashes total. Multiply that by the length of the message and you need M*2^33 bits.

c) Assuming that attacker's computer can process 220 hash/second, how long does it take at average to find pair of messages that have the same hash?

2^33 hashes / 220 hash/second = 39045157 seconds = ~1.23 years

d) Answer (b) and (c) when 128-bit hash is used instead.

Need M*2^65 bits of hashes instead.
2^65 hashes / 220 hash/second = <big> seconds = ~5317658339 years

Q-3-) Use Trapdoor Oneway Function with following secrets as described in lecture notes to encrypt plaintext P = '0101 0111'. Decrypt the resulting ciphertext to obtain the plaintext P

back. Show each step to get full credit.
S = {5, 9, 21, 45, 103, 215, 450, 946}
a = 1019, p = 1999

Public key:
T = 1019 * S mod 1999 = {1097, 1175, 1409, 1877, 1009, 1194, 779, 456}

Encrypting:
Y = 0*1097 + 1*1175 + 0*1409 + 1*1877 + 0*1009 + 1*1194 + 1*779 + 1*456 = 5481

Decrypting:
Z = 1019^-1 * Y mod 1999
Z = 1589 * Y mod 1999 = 1665

Then you can decrypt:
1665 > 946 ⇒ P8 = 1, Z' = 1665 - 946 = 719
719 > 450 ⇒ P7 = 1, Z' = 719 - 450 = 269
269 > 215 ⇒ P6 = 1, Z' = 269 - 215 = 54
54 < 103 ⇒ P5 = 0
54 > 45 ⇒ P4 = 1, Z' = 54 - 45 = 9
9 < 21 ⇒ P3 = 0
9 >= 9 ⇒ P2 = 1, Z' = 9 - 9 = 0
0 < 5 ⇒ P1 = 0

P = 0101 0111