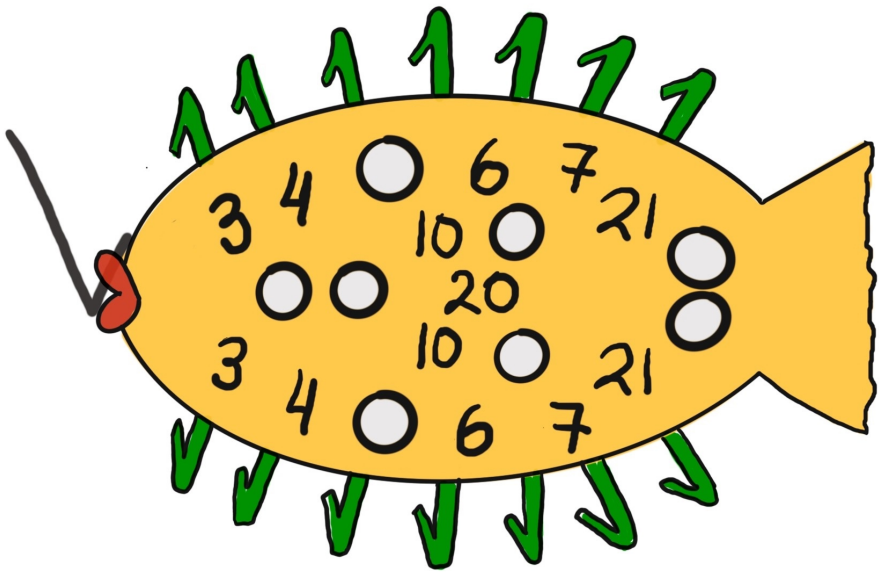
Problem 1. Cryptographic Fish

There are several ciphers in modern world named after «fish». There are ciphers BlowFish, TwoFish, ThreeFish and even cryptocurrency CryptoFish.

But Alice has found a new fish-like crypto object, here it is.



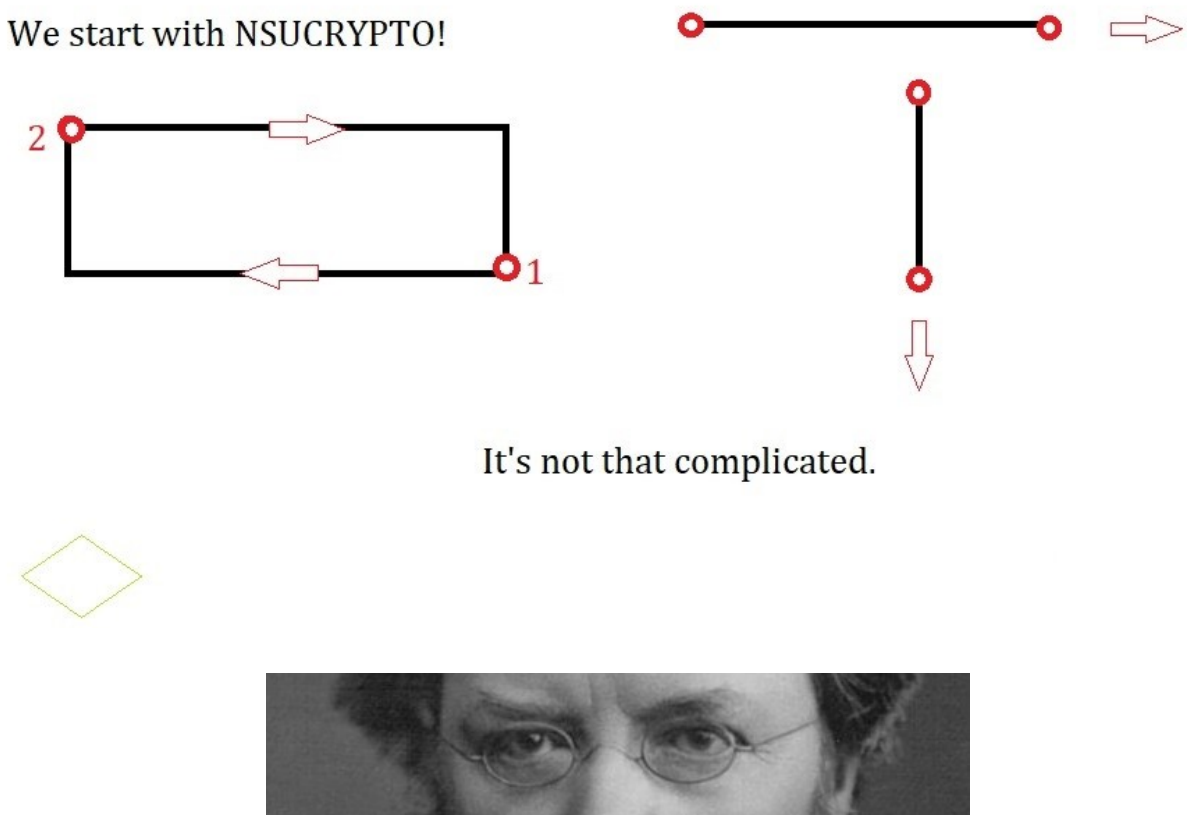
Please, could you find a key from it? It is a sum of the missed circled numbers.

Problem 2. «Decrypt with a hint»

Alice has found an interesting message from Bob. Here it is:

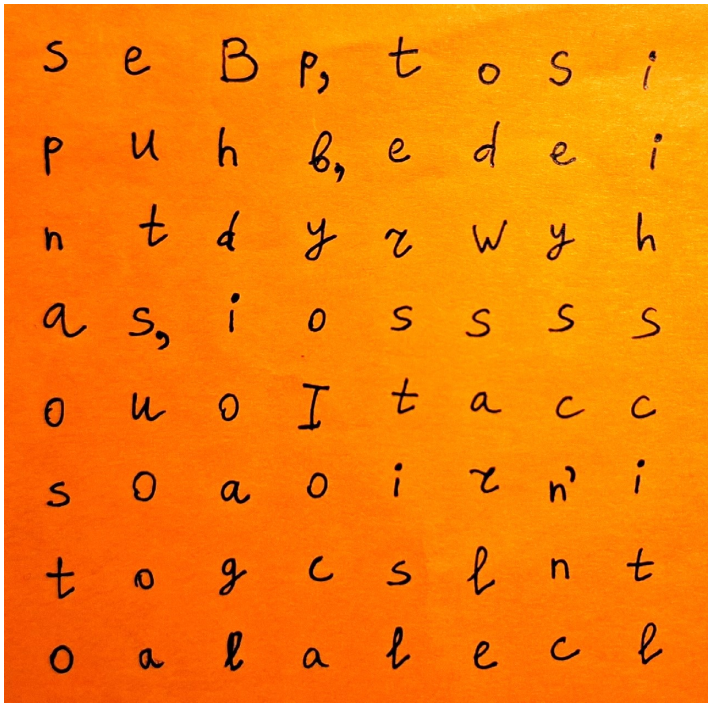
LFUZGAEPPXLSOANA

And here is a hint how the text was encrypted. Could you decrypt the text?

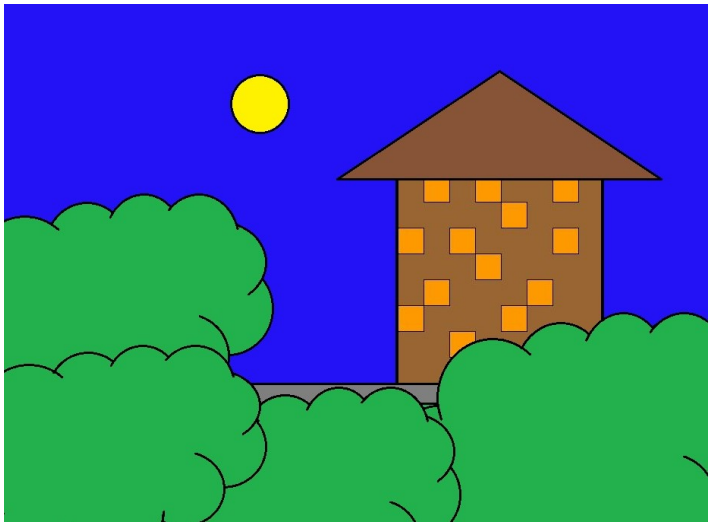


Problem 3. Alice's house

When Bob was returning from Alice's house, he found an encrypted letter on the road:



He guessed that it was a message from Alice. Bob looked around and immediately read it. What was the letter about?



Problem 4. «Weak key schedule for DES»

Alice is a novice cryptographer. She figured out how the DES encryption algorithm works and decided to implement it in order to exchange secret messages with Bob. She used the simplest ECB mode. But in her implementation, Alice made a mistake: inside the function F in addition of data with a round secret subkey, she forgot to change the index. So, in her implementation, in each round, the data is added modulo 2 with the first round key. Carol really wants to know what Alice and Bob are exchanging messages about. She even managed to get hold of a couple of files once. The `Book.txt` file contains an open message, and the `Book_Cipher.txt` file contains the corresponding encrypted text. Help Carol to find the secret encryption key and read the message she intercepted (the message is in hexadecimal format):

```
86991641D28259604412D6BA88A5C0A6471CA722
2C52482BF2D0E841D4343DFB877DC8E0147F3D5F
20FC18FF28CB5C4DA8A0F4694861AB5E98F37ADB
C2D69B35779D9001BB4B648518FE6EBC00B2AB10
```

Some explanations. Description of DES algorithm can be found in the web, see for inst. `https://csrc.nist.gov/files/pubs/fips/46/final/docs/nbs.fips.46.pdf` Consider an example. We are talking about the correct implementation of DES, where all 16 round subkeys are used correctly. For example, if we take the plaintext `8787878787878787`, and encrypt it with the DES key `0E329232EA6D0D73`, we end with the ciphertext `0000000000000000`. If the ciphertext is decrypted with the same secret DES key `0E329232EA6D0D73`, the result is the original plaintext `8787878787878787`. In the `Book.txt` file, each character corresponds to one byte of information according to the ASCII table. Since the DES algorithm processes 64 bits at a time, the first 8 characters of «Three Ri» will be used as the input message, which corresponds to the hexadecimal sequence `5468726565205269`. It should be noted that moving the carriage return to a new line in the file also takes two bytes `0D0A`.

Problem 5. «A simple hash function»

Carol invented a new hash function. The key $k = (k_1, \ldots, k_6)$ for this hash function is a binary vector of length 6. The input for the hash function is a sequence of digits. It should be divided into blocks of length 6. If the length of the sequence is not a multiple of 6 then it can be completed with 1, 2, 3, and so on up to the necessary length. For example, if an input is 7256 then it should be changed to 725612 first.

Then each block of input, say $(p_1, \ldots, p_6)$ should be transformed into the number by the rule: $(-1)^{k_1} \cdot p_1 + \ldots + (-1)^{k_6} \cdot p_6$. Here $(-1)^0 = 1$ and $(-1)^1 = -1$. Results of such calculations for blocks, say $n_1, n_2, n_3, n_4, \ldots$, then form a resulting hash value as $H = n_1 - n_2 + n_3 - n_4 + \ldots$.

For example, if the key is (001101) then hash for the sequence 134875 512293 is $H = (1 + 3 - 4 - 8 + 7 - 5) - (5 + 1 - 2 - 2 + 9 - 3) = -6 + 8 = 2$.

Carol applied her hash function in the system for logging at the bank website. Every user enters his password, say P, then system counts hash value $H(P, K)$ and if it coincides with the hash value from the data base, then user is logged.

But after some time Carol realized that the system is not secure. Malefactors can construct collisions and enter the system illegally. How do they do it? Propose a simplest algorithm how to get a collision of the first order for any known input sequence P if the key K is unknown.

By the way, find the shortest collision for the sequence from the example, $P =$ 134875 512293. Since K is unknown, the hash value $H(P, K)$ is still unknown to you.

The picture is taken from `https://www.lepuchin.com/Security-Brief-P13-Birthday-Attack`.

Problem 6. «RSA signature»

We want to sign the message M using the RSA-signature. As usually, let $N = p \cdot q$ be the RSA-modulus, where p and q are two big primes. Let e be the RSA-public exponent and d be the RSA-secret exponent satisfying that $e \cdot d = 1 \mod (p-1)(q-1)$. The desired signature is given by

$$S = M^d \mod N.$$

Suppose that the attacker knows the value

$$M_p := M^{d_p} \mod p,$$

but he doesn't know the value

$$M_q := M^{d_q} \mod q,$$

where

$$d_p := d \mod (p-1), \quad d_q := d \mod (q-1).$$

If the attacker knows the modulus N (but not p and q), the public exponent e (but not d), and the original message M, what secret signature parameters can he calculate? Justify the answer.