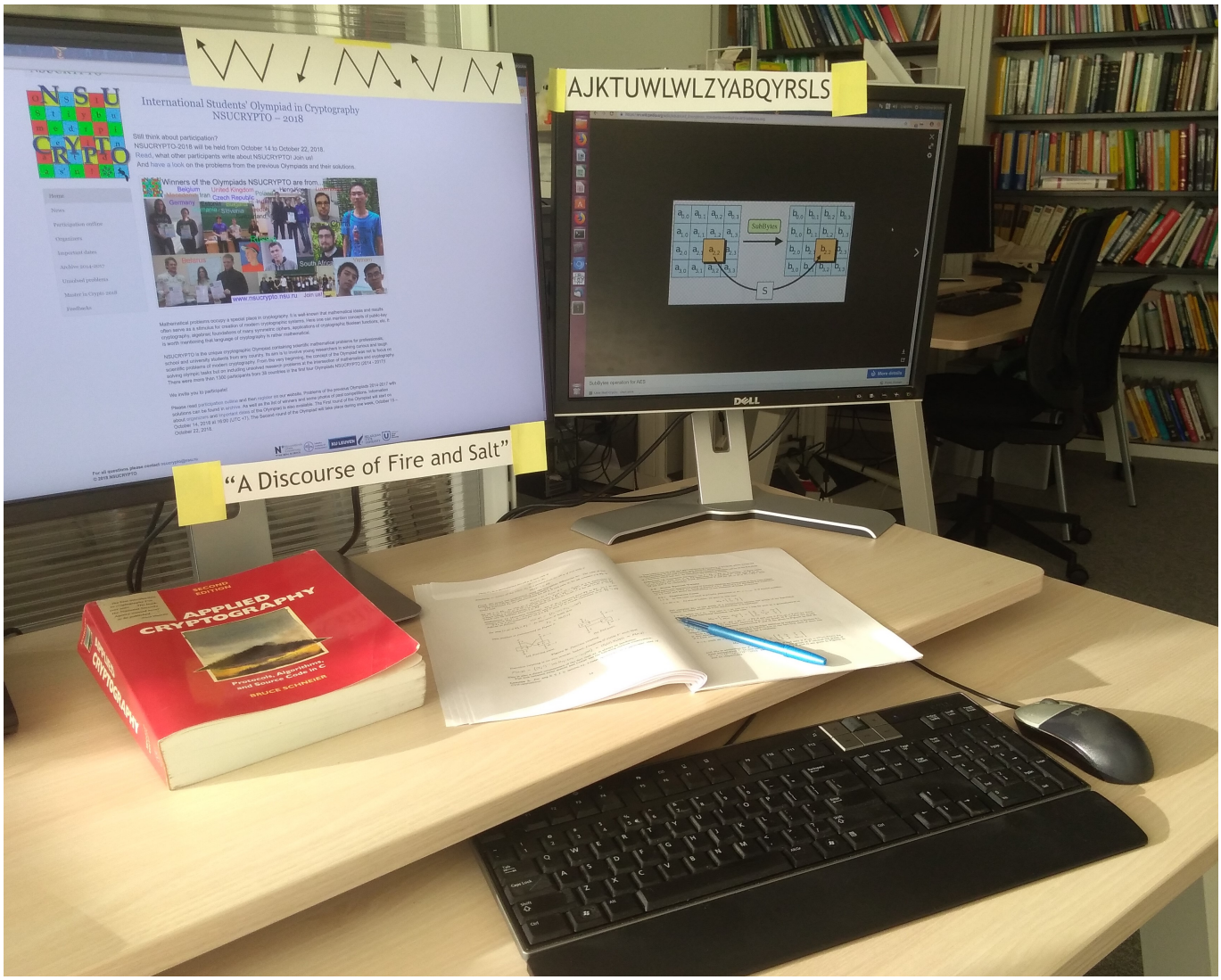
Problem 1. «Stickers»

Bob always takes into account all the recommendations of security experts. He switched from short passwords to long passphrases and changes them every month. Bob usually chooses passphrases from the books he is reading. Passphrases are so lengthy and are changed so often! In order to not forget them, Bob decided to use stickers with hints. He places them on his monitors (ooh, experts...). The only hope is that Bob's hint system is reliable because it uses encryption. But is that true? Could you recover Bob's current passphrase from the photo of his workspace?



Problem 2. «Key matrices»

Let n be an **odd** positive integer. In some cipher, a key is a binary $n \times n$ matrix

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix},$$

where $a_{i,j}$ is either 0 or 1, such that each diagonal of any length $1, 2, \dots, n-1, n$ contains an **odd** number of 1s.

What is the minimal and the maximal number of 1s that can be placed in a key matrix A?

Example. For $n = 3$, diagonals are the following ten lines:

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix}$$

Problem 3. «A sequence»

Two friends, Roman and Anton, are very interested in sequences and ciphers. Their new cryptosystem encrypts binary messages of length n, $X = (x_1, x_2, \ldots, x_n)$, where each x_i is either 0 or 1. A key K of the cipher is a set of n integers $a_1, a_2, \ldots, a_n$. The ciphertext C for the message X encrypted with the key K is the integer

$$Y = x_1 \cdot a_1 + x_2 \cdot a_2 + \ldots + x_n \cdot a_n.$$

Roman and Anton change their key regularly. Today, the key K is defined by

$$a_i = 2^i + (-1)^i \quad \text{for all } i = 1, \ldots, n.$$

The friends can easily decipher any message using the key defined by this sequence for any n!

1. Prove that the encryption is correct for this key K for any n: there are no two distinct input messages X^1 and X^2 such that their ciphertexts C^1 and C^2 are equal, i. e. $C^1 = C^2$.

2. Describe an algorithm which can be used to easily decipher any ciphertext C encrypted with today's key K. Here "easily" means that the algorithm should work much faster than checking all possible variants for an input message X.

Problem 4. «Quantum circuits»

Alice and Bob are interested in quantum circuits. They studied quantum operations and would like to use them for their simple cipher. Let an input plaintext be $P = (p_1, p_2, \ldots, p_{16}) \in \mathbb{F}_2^{16}$. The ciphertext $C \in \mathbb{F}_2^{16}$ is calculated as

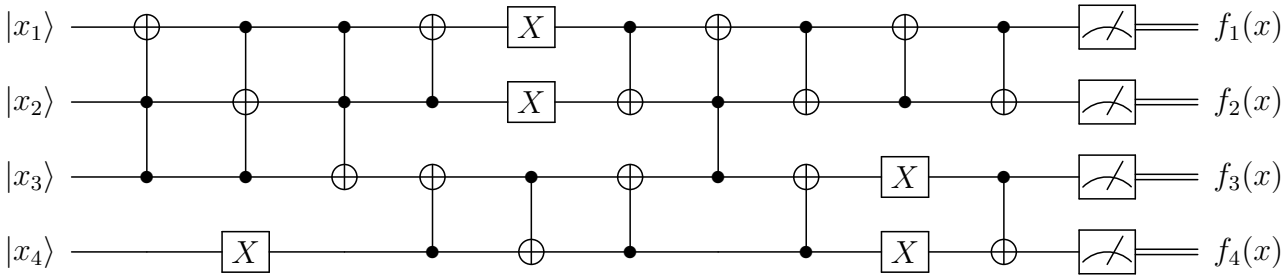
$$C = K \oplus \big(F(p_1, \ldots, p_4),\ F(p_5, \ldots, p_8),\ F(p_9, \ldots, p_{12}),\ F(p_{13}, \ldots, p_{16})\big),$$

where $K \in \mathbb{F}_2^{16}$ is a secret key and F is a function from $\mathbb{F}_2^4$ to $\mathbb{F}_2^4$; $\oplus$ is bitwise XOR.

The friends found a representation of F from wires and elementary quantum gates which form a quantum circuit. They use Dirac notation and denote computational basis states by $|0\rangle$ and $|1\rangle$. Further, quantum bits (qubits) are considered only in quantum states $|0\rangle$ and $|1\rangle$. Alice and Bob used the following quantum gates and circuit symbols:

Pauli-X gate	$\|x\rangle$ — X — $\|x \oplus 1\rangle$	acts on a single qubit in the state $\|x\rangle$, $x \in \{0, 1\}$.
controlled-NOT gate (CNOT gate)	$\|x\rangle$ — $\|x\rangle$ $\|y\rangle$ — $\|y \oplus x\rangle$	acts on two qubits in the states $\|x\rangle, \|y\rangle$, $x, y \in \{0, 1\}$; it flips the second qubit if and only if the first qubit is in the state $\|1\rangle$.
Toffoli gate (CCNOT gate)	$\|x\rangle$ — $\|x\rangle$ $\|y\rangle$ — $\|y\rangle$ $\|z\rangle$ — $\|z \oplus (x \wedge y)\rangle$	acts on three qubits in the states $\|x\rangle, \|y\rangle, \|z\rangle$, $x, y, z \in \{0, 1\}$; it flips the third qubit if and only if the states of the first and the second qubits are both equal to $\|1\rangle$.
$\|x\rangle$ — measurement — x	a measurement of a qubit in the state $\|x\rangle$, $x \in \{0, 1\}$, in the computational basis $\{\|0\rangle, \|1\rangle\}$.	
———	a wire carrying a single qubit (time goes left to right).	
═══	a wire carrying a single classical bit.	

A quantum circuit which describes action of F on $x = (x_1, x_2, x_3, x_4) \in \mathbb{F}_2^4$, where $F = (f_1, f_2, f_3, f_4)$ and $f_i, i = 1, 2, 3, 4$, are Boolean functions in 4 variables, is the following:



The problem. The friends encrypted the plaintext $P = (0011010111110010)$ and got the ciphertext $C = (1001101010010010)$. Find the secret key K!

Problem 5. «Bash-S3»

The sponge function Bash-f uses the permutation $S3$ that transforms a triple of 64-bit binary words a, b, c in the following way:

$$S3(a, b, c) = (b \vee \neg c \oplus a,\ \ a \vee c \oplus b,\ \ a \wedge b \oplus c).$$

Here $\neg$, $\wedge$, $\vee$, $\oplus$ denote the binary bitwise operations "NOT", "AND", "OR", "XOR" respectively. The operations are listed in descending order of priority. Let w^k also denote the cyclic shift of a 64-bit word w to the left by $k \in \{1, 2, \ldots, 63\}$ positions.

Alice wants to strengthen $S3$. She can add by XOR any input a, b, c or its cyclic shift to any output. She must use at least one cyclic shift and she cannot add two identical terms to the same output.

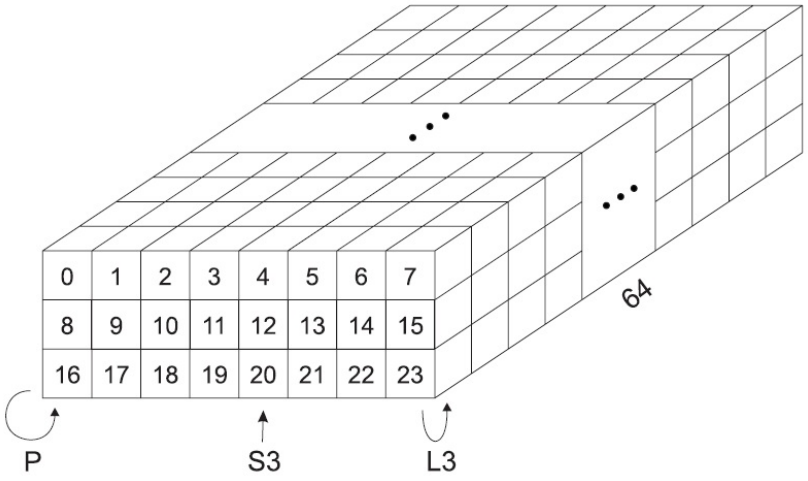
Help Alice to change $S3$ in such a way that a modified $S3$ will be still a permutation!

Remark 1. For example, in the expression $b \vee \neg c \oplus a$, we firstly calculate $\neg c$, then calculate $b \vee \neg c$, and after that the final result (according to descending order of operations priority).

Remark 2. The modification

$$(b \vee \neg c \oplus a \oplus a^{11},\ \ a \vee c \oplus a^7 \oplus c,\ \ a \wedge b \oplus b^{32})$$

is allowed but it does not satisfy the permutation condition.



Problem 6. «Metrical cryptosystem – 2»

Let $\mathbb{F}_2^n$ be an n-dimensional vector space over the field $\mathbb{F}_2 = \{0, 1\}$. Alice and Bob exchange messages using the following cryptosystem.

- First, they use a supercomputer to calculate two special large secret sets $A, B \subseteq \mathbb{F}_2^n$ which have the following property: there exists a constant ℓ ($\ell \geqslant 26$), such that for any $x \in \mathbb{F}_2^n$ it holds
$$d(x, A) + d(x, B) = \ell,$$
where $d(x, A)$ denotes Hamming distance from the vector x to the set A.

- Alice then saves the number ℓ, the set A and a set of vectors $a_1, a_2, \ldots, a_r$ such that for any $k : 0 \leqslant k \leqslant \ell$, there is a vector a_i at distance k from A. Similarly, Bob saves the number ℓ, the set B and a set of vectors $b_1, b_2, \ldots, b_s$ such that for any $k : 0 \leqslant k \leqslant \ell$, there is a vector b_i at distance k from B.

- Text messages are encrypted letter by letter. In order to encrypt a letter Alice replaces it with its number in the alphabet, say k. Then she chooses some vector a_i at distance k from the set A and sends this vector over to Bob. Bob then calculates the distance $d(a_i, B)$ and using the property of the sets A, B, calculates $k = \ell - d(a_i, B)$. So, he gets the letter Alice sent. If Bob wants to send an encrypted message to Alice, he does the same but using his saved vectors and the set B.

Eve was able to hack the supercomputer when it was calculating the sets A and B. She extracted the set C from its memory, which consists of all vectors of $\mathbb{F}_2^n$ that are at distance 1 or less from either A or B. She also learned that ℓ is even.

Help Eve to crack the presented cryptosystem (to decrypt any short intercepted message)! You know that she has an (illegal) access to the supercomputer, which can calculate and output the list of distances from all vectors of $\mathbb{F}_2^n$ to any input set D in reasonable (but not negligible) time.

Remark I. Recall several definitions and notions. The *Hamming distance* $d(x, y)$ between vectors x and y is the number of coordinates in which these vectors differ. Distance from vector $y \in \mathbb{F}_2^n$ to the set $X \subseteq \mathbb{F}_2^n$ is defined as $d(y, X) = \min_{x \in X} d(y, x)$.

Problem 7. «A fixed element»

A polynomial $f(X_1, \ldots, X_n) \in \mathbb{F}_2[X_1, \ldots, X_n]$ is called *reduced* if the degree of each X_i in f is at most 1. For $0 \leqslant r \leqslant n$, the rth order Reed — Muller code of length 2^n, denoted by $R(r, n)$, is the $\mathbb{F}_2$-space of all reduced polynomials in $X_1, \ldots, X_n$ of total degree less or equal than r. We also define $R(-1, n) = \{0\}$.

The general linear group $\mathrm{GL}(n, \mathbb{F}_2)$ acts on $R(r, n)$ naturally: Given $A \in \mathrm{GL}(n, \mathbb{F}_2)$ and $f(X_1, \ldots, X_n) \in R(r, n)$, Af is defined to be the reduced polynomial obtained from $f((X_1, \ldots, X_n)A)$ by replacing each power X_i^k $(k \geqslant 2)$ with X_i. Consequently, $\mathrm{GL}(n, \mathbb{F}_2)$ acts on the quotient space $R(r, n)/R(r-1, n)$.

Let $A \in \mathrm{GL}(n, \mathbb{F}_2)$ be such that its characteristic polynomial is a primitive irreducible polynomial over $\mathbb{F}_2$. Prove that the only element in $R(r, n)/R(r-1, n)$ fixed by the action of A is 0.