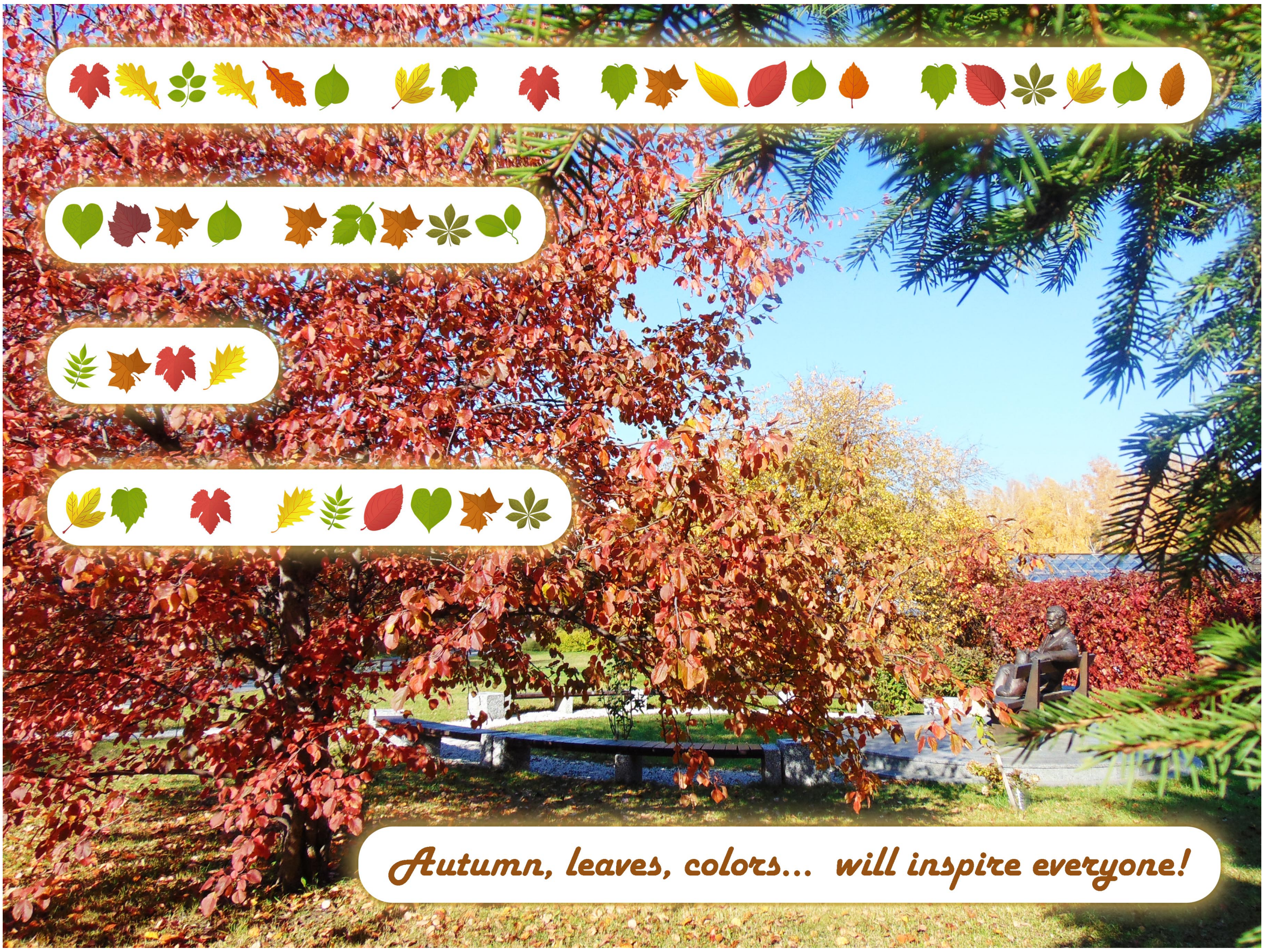
Problem 1. «Autumn leaves»

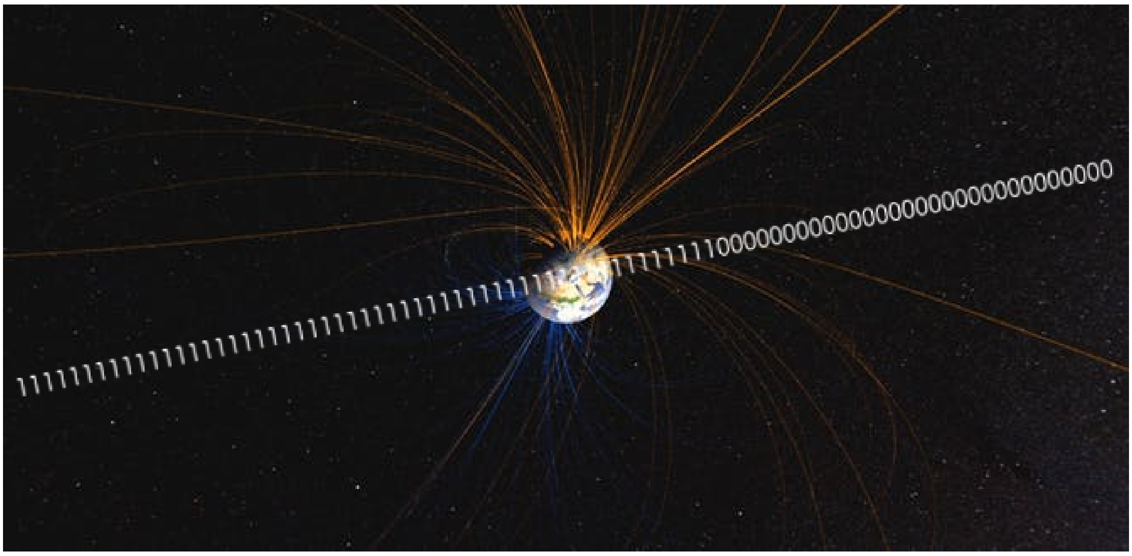
Read a hidden message!..



Problem 2. «The magnetic storm»

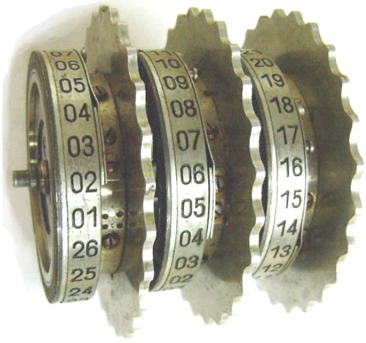
A hardware random number generator is a device that generates random sequences consisting of 0s and 1s. Unfortunately, a disturbance caused by a magnetic storm affected this random number generator. As a result, the device had generated a sequence of 0s of length k (where k is a positive integer), and then started to generate an infinite sequence of 1s.

Prove that at some point the generator will produce a number $1 \ldots 10 \ldots 0$ that is divisible by 2019.

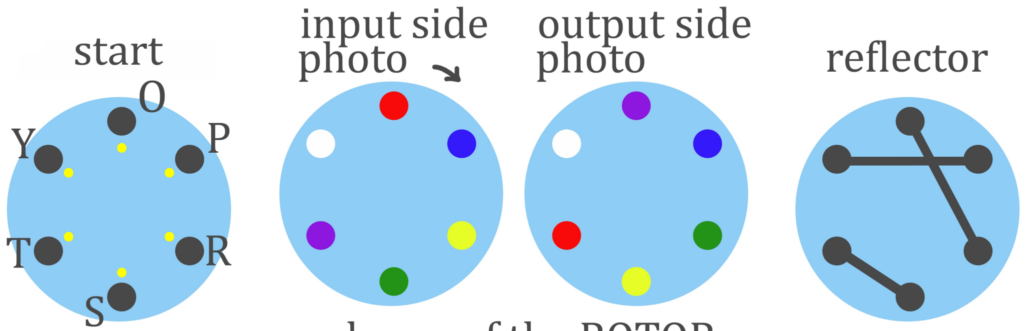


Problem 3. «A rotor machine»

In one country rotor machines were very useful for encryption of information.



Eve knows that for some secret communication a simple rotor machine was used. It works with letters `O`, `P`, `R`, `S`, `T`, `Y` only and has an input circle with lamps (start), one rotor and a reflector. See the scheme below.



scheme of the ROTOR

The input circle and the reflector are fixed in their positions while the rotor can be in one of 6 possible positions. After pressing a button on a keyboard, an electrical signal corresponding to the letter goes through the machine, comes back to the input circle, and the appropriate lamp shows the result of encryption. After each letter is encrypted, the rotor turns right (i. e. clockwise) on 60 degrees. Points of different colors on the rotor sides indicate different noncrossing signal lines within the rotor.

For instance, if the rotor is fixed as shown on the picture above, then if you press the button `O`, it will be encrypted as `T` (the signal enters the rotor via red point, is reflected and then comes back via purple line). If you press `O` again, it will be encrypted as `R`. If you press `T` then, you will get `S` and so on.

Eve intercepted a secret message: `TRRYSSPRYRYROYTOPTOPTSPSPRS`. Help her to decrypt it keeping in mind that Eve does not know the initial position of the rotor.

Problem 4. «16QAM»

For sending messages, Alice and Bob use a fiber-optic communication via 16QAM technology. This technology allows to send messages whose alphabet consists of 16 letters, where each letter is usually encoded with a 4-bit Gray code. While a message is transmitted in the channel, single errors in codewords of the Gray code are possible.

Alice has read an interesting book and would like to share her enthusiasm with Bob! Alice sent a short fragment from the book to Bob. Due to the characteristics of the communication channel used, she divided the text into two parts and sent them separately. In the first part, she placed all of the 16 consonants that occurred in this fragment; in the second part, she placed vowels, a hyphen and punctuation marks. Then Alice also encoded the letters with Hamming code to be able to correct single errors. She applied a 7-bit Hamming code with the parity-check matrix whose columns are written in lexicographical order.

Bob received the following two parts of ciphertext (given in hexadecimal notation):

Part 1

```
66674C36666F43D3C199900AA1AA325992A
67A59D9B4A8B69330D1BC000153367A5E33
D30E6692D0F349D3321FFFF0ED706667A7F
670D999679F4AA67561BA679B4AA54F34D5
AB0F4AACCF000055CE633670D9DA54CE37F
660DE19CD995335495523CCAAA8F1E03325
86CF48A98CD9B387FD9D546A99E9D200033
3201513FE5B4AA00CCCE9667554CD2CCCB3
330F32A666553CD756AC3E0674E9D369E1D
C6A9999780007F00961E66465519FEA8B25
14CCCB332AA63332CCCE6D2A99AACCCC004
```

Part 2

```
66CA61967319CCD2CE76998CE6433332D19
B46784C65334E999A402ADA0265A99A6633
33319B32D3299698CCC96986619967134CC
B4CE23333334CC6730CE90170CCCD2CE669
996A61999EA63332CCA4C3332D4CD3334CC
D3319994730CCCD3A6669D96A66999699B3
98640CC86CE619676AD4CD3308999866D33
79321C33210B4C6732199B53218019A404C
D2DE65A986663398CCCCCB5319CC6665997
B96A63398CD9CCD2CD9A399A66339866619
98CD9CC325A6339CCE619998C04C66CE633
```
996A61998CF66967334CC66CA6199865E$(0)_2$

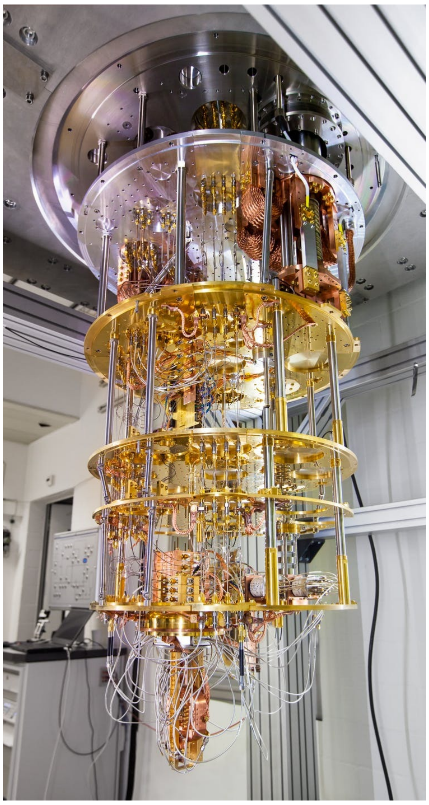
Also, he received the following number sequence: 22, 19, 3, 3, 36, 53, 3, 33, 20, 28. Each number indicates how many consonants are contained between the punctuation marks.

Recover the text and find the main character of the book Alice has read!

Problem 5. «A promise and money»

A group of young cryptographers are interested in quantum computings and really want to buy a quantum computer. A millionaire gave them a certain amount of money (say, n cryptographers; X_i for each of them, $i = 1, \ldots, n$). He also made them promise that they would not tell anyone, including each other, how much money everyone of them had received.

- Could you help the cryptographers to invent an algorithm how to find out (without breaking the promise) whether the total amount of money they have, $\sum_{i=1}^{n} X_i$, is enough to buy a quantum computer?
- What do you think whether there are such algorithms protecting the secrets of honest participants from dishonest ones?
- What weaknesses does your algorithm have (if someone breaks the promise)? Does it always protect the secret of the honest participants from the dishonest ones?



IBM's 50 qubit quantum computing system

Problem 6. «Calculator»

Alice and Bob are practicing in developing toy cryptographic applications for smartphones. This year they have invented `Calculator` that allows one to perform the following operations modulo 2019:

- to insert at most 4-digit positive integers (digits from 0 to 9);
- to perform addition, subtraction and multiplication of two numbers;
- to store temporary results and read them from the memory.

Suppose that Alice wants to send Bob a ciphertext y (given by a 4-digit integer). She sends y from her smartphone to Bob's `Calculator` memory. To decrypt y, Bob needs to get the plaintext x (using his `Calculator`) by the rule $x = f(y) \bmod 2019$, where f is a secret polynomial known to Alice and Bob only.

At the most inopportune moment, Bob dropped his smartphone and broke its screen. Now, the button $\boxed{+}$ as well as all digits except $\boxed{2}$ are not working.

Help Bob to invent an efficient algorithm how to decrypt any ciphertext y using `Calculator` in his situation if the current secret polynomial is $f(y) = y^5 + 1909y^3 + 401y$. More precisely, suggest a short list of commands, where each command has one of the following types ($1 \leqslant j, k < i$):

$$\begin{array}{llll} S_i = y, & S_i = 2, & S_i = 222, & S_i = S_j - S_k, \\ & S_i = 22, & S_i = 2222, & S_i = S_j * S_k. \end{array}$$

The first command has to be $S_1 = y$. In the last command, the resulted plaintext x has to be calculated. We remind that all calculations are modulo 2019. In particular, the integer 2222 becomes 203 immediately after entering. The shorter the list of commands you suggest, the more scores you get for this problem.

Example. The following list of commands calculates $x = y^2 - 4$:

Command	Result
$S_1 = y$	y
$S_2 = S_1 * S_1$	y^2
$S_3 = 2$	2
$S_4 = S_3 * S_3$	4
$S_5 = S_2 - S_4$	$y^2 - 4$

Problem 7. «APN + Involutions»

Alice wants to construct a block cipher with heavy use of **involutions** as its subcomponents; this minimizes difference in the algorithms for encryption and decryption. She knows that **almost perfect nonlinear permutations** (**APN permutations**) are the best choice of subcomponents to resist attacks based on differential technique. She wants to construct a set of APN permutations that are involutions for every $n \geqslant 2$.

Alice also knows that any involution can be expressed as the product of disjoint **transpositions**. So, she decides to study the following involution

$$g = \prod_{i=1}^{d} (\alpha_i, {\alpha'}_i),$$

where $\{\alpha_i, {\alpha'}_i\} \cap \{\alpha_j, {\alpha'}_j\} = \emptyset$ for all $i, j \in \{1, ..., d\}$, $i \neq j$, $1 \leqslant d \leqslant 2^{n-1}$.

Alice needs your help to get APN permutations among such involutions g. Find answers to the following questions!

Q1 Let

$$\Lambda(g) = \big\{\alpha_i \oplus {\alpha'}_i : \ i = 1, ..., d\big\}, \qquad \widehat{\Lambda}(g) = \big[\alpha_i \oplus {\alpha'}_i : \ i = 1, ..., d\big],$$
$$\mathrm{B}(g) = \big\{x \oplus y : \ \{x, y\} \subseteq \mathrm{FixP}(g), \ x \neq y\big\}, \quad \widehat{\mathrm{B}}(g) = \big[x \oplus y : \ \{x, y\} \subseteq \mathrm{FixP}(g), \ x \neq y\big],$$

where $\mathrm{FixP}(g)$ is the set of all **fixed points** of g, i. e. $\mathrm{FixP}(g) = \{x \in \mathbb{F}_2^n : \ g(x) = x\}$.

Suppose that g is an APN permutation. Get necessary conditions for multisets $\widehat{\Lambda}(g)$, $\widehat{\mathrm{B}}(g)$ and sets $\Lambda(g)$, $\mathrm{B}(g)$. Prove that if your conditions are not satisfied, then g is not an APN permutation.

Q2 Let

$$\mathrm{d}_{a,b}(g) = |\{x \in \mathbb{F}_2^n : \ g(x \oplus a) \oplus g(x) = b\}|, \quad a, b \in \mathbb{F}_2^n.$$

Let g be an involution and APN. Find $\mathrm{d}_{a,a}(g)$ for each nonzero $a \in \mathbb{F}_2^n$.

Q3 Can you get the nontrivial upper bound on $|\mathrm{FixP}(g)|$?

Remark. Let us recall the relevant definitions.

- $\mathbb{F}_2^n$ is the vector space of dimension over $\mathbb{F}_2 = \{0, 1\}$.
- A vector $x \in \mathbb{F}_2^n$ has the form $x = (x_1, ..., x_n)$, where $x_i \in \mathbb{F}_2$. For two vectors $x, y \in \mathbb{F}_2^n$ their sum is $x \oplus y = (x_1 \oplus y_1, ..., x_n \oplus y_n)$, where $\oplus$ stands for XOR operation.
- Let $\widehat{X} = \big[x_1, ..., x_d\big]$ be a multiset with the underlying set $\mathbb{F}_2^n$, where $x_1, ..., x_d \in \mathbb{F}_2^n$.
  Note that all elements in a set are distinct. Unlike a set, a multiset allows for multiple instances for each of its elements.
- A **permutation** s is a mapping from $\mathbb{F}_2^n$ to $\mathbb{F}_2^n$ such that $s(x) \neq s(y)$ for all $x, y \in \mathbb{F}_2^n$, $x \neq y$.
- An **involution** s is a permutation that is its own inverse, $s^2(x) = s(s(x)) = x$ for all $x \in \mathbb{F}_2^n$.
- For any different vectors $\alpha, \beta \in \mathbb{F}_2^n$, a permutation s is called a **transposition** if $s(\alpha) = \beta$, $s(\beta) = \alpha$ and $s(x) = x$ for all $x \in \mathbb{F}_2^n \backslash \{\alpha, \beta\}$; it is denoted by $s = (\alpha, \beta)$.
- A permutation s is called **APN** (Almost Perfect Nonlinear) if, for every nonzero $a \in \mathbb{F}_2^n$ and every $b \in \mathbb{F}_2^n$, the equation $s(x \oplus a) \oplus s(x) = b$ has at most 2 solutions.