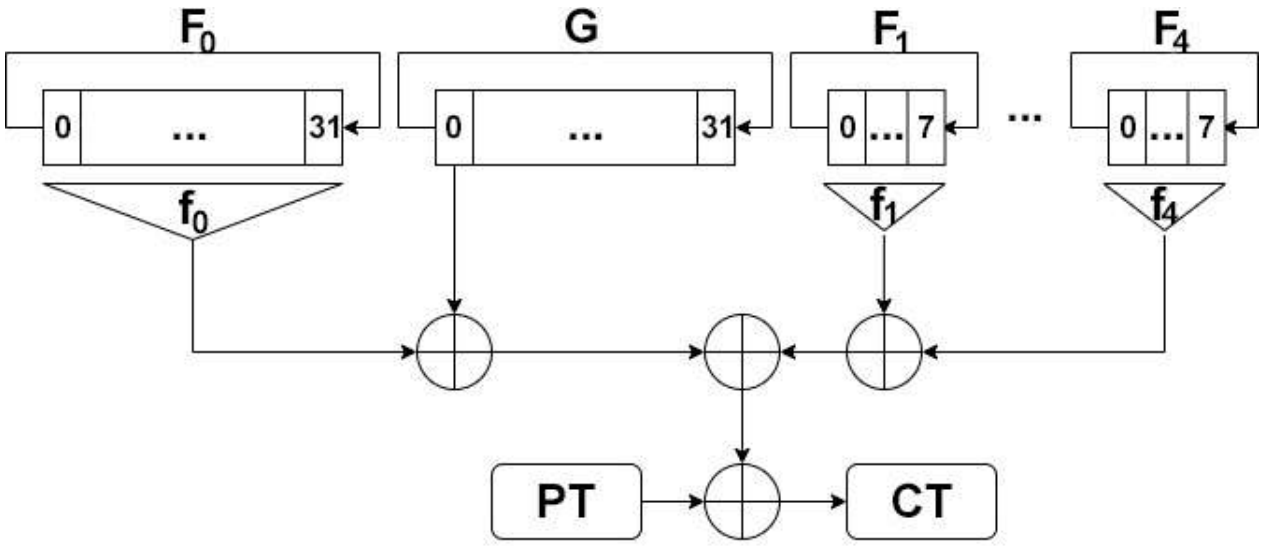
Problem 4. «Negotiations»

Before the upcoming negotiations, our intelligence managed to intercept an encrypted telegram sent from the enemy headquarters to its negotiation team:

```
289e1fa87b606203a790f93e3a3e4c1a
7b60474a1df415badf49b12b6b6e16ef
7e559f015c.
```

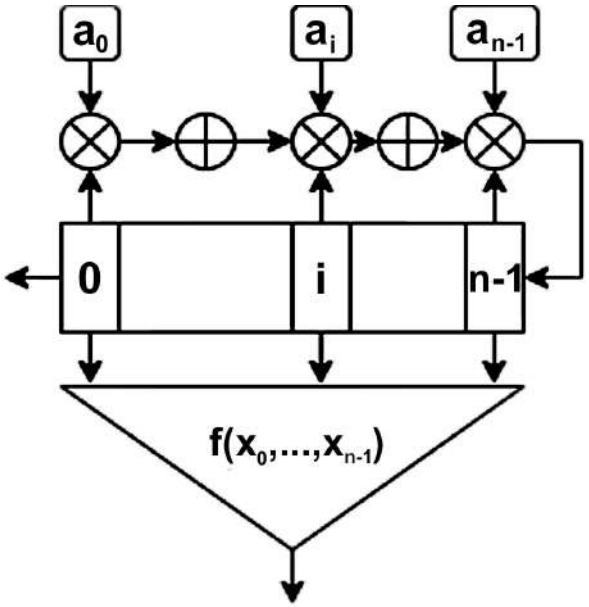
Undercover operative Jack Beam managed to recover the schematics of the cipher device in use. It is given below:



The cipher device is a stream cipher consisting of six binary shift registers, where for each clock cycle a bit of the output keystream is XORed with a bit of plaintext.

Another undercover operative, Jim Daniels, managed to provide a more detailed description of the shift registers.

For each clock cycle, the register output is formed by feeding its state into a filter function f (least significant bits are on the left). The state is then bitwise multiplied with the coefficients $a_0, \ldots, a_{n-1} \in GF(2)$ of a feedback polynomial $F(x) = x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$. The results of the multiplications are summed modulo 2, the register state is shifted one cell towards the least significant bits, and the result of the sum is written into the most significant cell, see the next figure.



Jim Daniels wrote down the feedback polynomials:

$F_0(x) = x^{32} + x^{15} + x^9 + x^7 + x^4 + x^3 + 1,$
$G(x) = x^{32} + x^{31} + x^{29} + x^{25} + x^{19} + x^{18} + x^{17} + x^{16} + x^9 + x^8 + x^7 + x^3 + x^2 + x + 1,$
$F_1(x) = x^8 + x^7 + x^6 + x^5 + x^2 + x + 1,$
$F_2(x) = x^8 + x^5 + x^3 + x + 1,$
$F_3(x) = x^8 + x^6 + x^5 + x^2 + 1,$
$F_4(x) = x^8 + x^6 + x^5 + x + 1.$

He found the truth tables in hexadecimal format for the filter functions $f_1, \ldots, f_4$:

f_1 : 431DCF02529EF04356F2B59E90860BD22FBCEFDB380F8838767DD716E9712A04,
f_2 : 4617142FE4475B3FF3D50CBB58E7A0F40CAC85B8C278A0C131FFD85413BE9E4A,
f_3 : 0138CCC35F889EAA5D8F1EE0442FB54D2AEAEFB96288DB3303FD1FBF65860A20,
f_4 : 095F8FFC8CA0C53C5D3782243D36EE575186DBD244CB9DE1E0ED1730CACF2053.

Also he got the function f_0. Thus, $f_0(x_0, \ldots, x_{31}) = f(x_5, x_{21}, x_{29}, x_1, x_0, x_{27}, x_{12})$, where it holds $f(x_0, \ldots, x_7) = x_0x_2x_5x_6 + x_0x_3x_5x_6 + x_0x_1x_5x_6 + x_1x_2x_5x_6 + x_0x_2x_3x_6 + x_1x_3x_4x_6 + x_1x_3x_5x_6 + x_0x_2x_4 + x_0x_2x_3 + x_0x_1x_3 + x_0x_2x_6 + x_0x_1x_4 + x_0x_1x_6 + x_1x_2x_6 + x_2x_5x_6 + x_0x_3x_5 + x_1x_4x_6 + x_1x_2x_5 + x_0x_3 + x_0x_5 + x_1x_3 + x_1x_5 + x_1x_6 + x_0x_2 + x_1 + x_2x_3 + x_2x_5 + x_2x_6 + x_4x_5 + x_5x_6 + x_2 + x_3 + x_5$.

The second register lacks a filter function, its output is the least significant (leftmost) bit of the register state.

It is known that the first 22 bytes of the intercepted message contain direct output of the cipher device without XORing any plaintext, and the subsequent 15 bytes contain an encrypted message, capable of altering the course of the upcoming negotiations.

The encrypted message consists of letters of the Latin alphabet in a 5-bit encoding:

Symbol	Code	Symbol	Code	Symbol	Code	Symbol	Code
A	00001	H	01000	O	01111	V	10110
B	00010	I	01001	P	10000	W	10111
C	00011	J	01010	Q	10001	X	11000
D	00100	K	01011	R	10010	Y	11001
E	00101	L	01100	S	10011	Z	11010
F	00110	M	01101	T	10100		
G	00111	N	01110	U	10101		

Question Could you help your group to decrypt the message and in this way to strengthen its negotiation position?

Remark. Let us present here some test vectors for the problem.

Register 1.
Initial state: 00000000000000001000001010001000
LFSR output: 0000000000000000100000101000100001000000000011001100
Register output: 0101001100000001001011011101111011010110110100011100

Register 2.
Initial state: 11010001000010010000101011001011
Register output: 1101000100001001000010101100101100100000111011101101

Register 3.
Initial state: 11111111
LFSR output: 11111111011101010011
Register output: 00100011010010010110

Register 4.
Initial state: 00101110
LFSR output: 00101110110111100101
Register output: 10001101001001011010

Register 5.
Initial state: 00101110
LFSR output: 00101110111101100111
Register output: 00110100100101101000

Register 6.
Initial state: 00101100
LFSR output: 00101100110110000111
Register output: 00101100010011011000