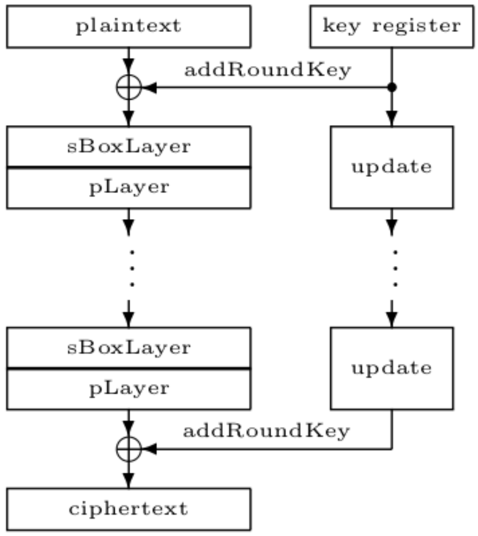
Problem 3. «A modification of PRESENT»

Peter decided to modify the wellknown cipher PRESENT.

At first we give a description of PRESENT according to the paper PRESENT: An Ultra-Lightweight Block Cipher



It is a classical Substitution-Permutation network (SP-network) that consists of 31 rounds with the block size equal to 64 bits and the key size equal to 80 bits. Each of the 31 rounds consists of an XOR operation to introduce a round key K_i for $1 \leqslant i \leqslant 32$, where K_{32} is used for post-whitening, a non-linear substitution layer, and a linear bitwise permutation P. The non-linear layer uses a single 4-bit S-box S which is applied 16 times in parallel in each round.

addRoundKey. Given current `state` $b_{63} \dots b_0$ and round key $K_i = k^i_{63} k^i_{62} \dots k^i_0$ for $1 \leqslant i \leqslant 32$, `addRoundKey` consists of the operation $b_j \to b_j \oplus k^i_j$ for $0 \leqslant j \leqslant 63$.

sBoxlayer. The S-box is a permutation from $\mathbb{F}_2^4$ to $\mathbb{F}_2^4$. For `sBoxLayer` the current `state` $b_{63} \dots b_0$ is considered as sixteen 4-bit words $w_{15} \dots w_0$ where $w_i = b_{4 \cdot i+3} || b_{4 \cdot i+2} || b_{4 \cdot i+1} || b_{4 \cdot i}$ for $0 \leqslant i \leqslant 15$ and the output nibble $S[w_i]$ provides the updated `state` values in the obvious way. The action of this box in hexadecimal notation is given by the following table.

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S(x)$	c	5	6	b	9	0	a	d	3	e	f	8	4	7	1	2

pLayer. The bit permutation is given by the table. Bit i of `state` is moved to bit position $P(i)$.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$P(i)$	0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51	4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55

i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
$P(i)$	8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63

The key schedule. The user-supplied key is stored in a key register `K` and represented as $\mathtt{k}_{79}\mathtt{k}_{78} \dots \mathtt{k}_0$. At round i the 64-bit round key $K_i = k_{63} k_{62} \dots k_0$ consists of the 64 leftmost bits of the current contents of register `K`. Thus at round i we have that: $K_i = k_{63} k_{62} \dots k_0 = \mathtt{k}_{79}\mathtt{k}_{78} \dots \mathtt{k}_{16}$. After extracting the round key K_i, the key register $\mathtt{K} = \mathtt{k}_{79}\mathtt{k}_{78} \dots \mathtt{k}_0$ is updated as follows. The key register is rotated by 61 bit positions to the left, then the left-most four bits $\mathtt{k}_{79}\mathtt{k}_{78}\mathtt{k}_{77}\mathtt{k}_{76}$ are passed through the PRESENT S-box, and finally the `round_counter` value i is XORed with bits $\mathtt{k}_{19}\mathtt{k}_{18}\mathtt{k}_{17}\mathtt{k}_{16}\mathtt{k}_{15}$ of `K` with the least significant bit of `round_counter` on the right.

see the next page

What Peter has modified:

• In **sBoxlayer**, he changed S-box to the following

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S(x)$	c	8	d	1	e	a	7	b	4	0	5	9	6	2	f	3

• In **pLayer**, he applied permutation P^3 instead of P.

• In **the key schedule**, he rotated the key register by 16 bit positions to the left instead of 61. And he used his new S-box from **sBoxlayer** here.

• Finally, he reduced the number of rounds to 15.

As a result Peter got the new cipher `Peter-PRESENT`. Below you can find examples of test vectors for `Peter-PRESENT` that are given as integers in hexadecimal notation.

plaintext	key	ciphertext
0000000000000000	00000000000000000000	f778777b0774f772
ffffffffffffffff	00000000000000000000	888708847883888d
0000000000000000	ffffffffffffffffffff	7ff8fffb0ffc7ffa
ffffffffffffffff	ffffffffffffffffffff	00078004700b0005

Peter states that his modification is rather good. But his friend Mark does not think so. He claims that it is enough to get only two pairs «plaintext–ciphertext» (P_1, C_1), (P_2, C_2), where $C_i =$ `Peter-PRESENT`(P_i, K), $i = 1, 2$, and K is the unknown key, for reading any message C encrypted with this key K in the ECB mode.

Peter decides to argue with Mark and presents the following pairs, where P_1 and P_2 forms the message `!NSUCRYPTO-2015!` (ASCII codes of letters and little-endian order of bytes are used to form 64-bits integers as the inputs $b_{63}b_{62}\dots b_0$):

`!NSUCRYP` $\rightarrow$ $P_1 =$ `5059524355534e21` $\rightarrow$ $C_1 =$ `2ddbf038b201448f`
`TO-2015!` $\rightarrow$ $P_2 =$ `21353130322d4f54` $\rightarrow$ $C_2 =$ `d4bf134bd57f4df2`

And he asks Mark to read the secrete message whose ciphertext C is:

```
C =  37aa471c953defe1  91aa595c0236edc9  80f10a020c33e5cb
     ddf14e15923df8dc  8cf8470d027af1db  9caa061e9537ead1
     92e10a1e072ea2c0  d1f1501e9b27f2c3  94e750140134e386
     92f6595b093de3d2  99ec435b0235ebdc  83ef4b099b37f886
     9eef461e4f76eecf  9eaa4912093df8d2  ddf15e129231f8c7
     89ec45184f3ee4cf  94e25e5b9c36eddc  87e55a0b9221a2d2
     ddae471d0e36a2d2  9aec4b159533efca  98e5495b0b34eb86
     9cf643180e34ffc3  89aa4c124f21e4c9  ddf6594ad57aefce
     dbfb500e9b34efc5
```

Can Mark win the argument?