4CR3 A1

1.

a. (Full steps at the end)

The protocol would be as follows:
Alice generates the 50 key bits using the LFSR
To get the bits, she would do:
$a_{20} = a_0+a_1+a_3+a_5+a_9+a_{13}+a_{14}+a_{16}+a_{18}$ (mod 2)
a20= 0 + 0 + 1 + 1 + 1 + 1 + 1 + 1 + 1 (mod 2) = 1
a21= a1 + a2 + a4 + a6 + a10 + a14 + a15 + a17 + a19 (mod 2)
a21= 0 + 1 + 0 + 1 + 0 + 1 + 0 + 1 + 1 (mod 2) = 1
(…)
a48= a28 + a29 + a31 + a33 + a37 + a41 + a42 + a44 + a46 (mod 2)
a48= 0 + 1 + 1 + 1 + 1 + 1 + 0 + 0 + 1 (mod 2) = 0
a49= a29 + a30 + a32 + a34 + a38 + a42 + a43 + a45 + a47 (mod 2)
a49= 1 + 1 + 1 + 0 + 1 + 0 + 0 + 0 + 1 (mod 2) = 1

Final generated key bits: 00110110010101101111111010110111110111100100001101
Next, Alice would XOR the bits of the generated bit with the plain text
To do this, she would do the steps:
Generated bits:
00110110010101101111111010110111110111100100001101
Plaintext bits:
00111101011100111000000111001100010011010110110100

bit 0 of ciphertext = 0 XOR 0 = 0
bit 1 of ciphertext = 0 XOR 0 = 0
(...)
bit 48 of ciphertext = 0 XOR 0 = 0
bit 49 of ciphertext = 1 XOR 0 = 1

This results in the ciphertext of 00001011001001010111111101111011100100110010111001

For Bob to decrypt this text, he would first generate the same key bits as Alice using the LFSR. The process is the same as above. To decrypt the ciphertext using the generated bits, since the inverse operation of XOR is also XOR, he can XOR the bits of the ciphertext with the generated bits. The process is as follows:
Generated bits:
00110110010101101111111010110111110111100100001101
Ciphertext bits:
00001011001001010111111101111011100100110010111001

bit 0 of plaintext = 0 XOR 0 = 0
bit 1 of plaintext = 0 XOR 0 = 0

(...)
bit 48 of plaintext = 0 XOR 0 = 0
bit 49 of plaintext = 1 XOR 1 = 0

The resulting plaintext is 00111101011100111000000111001100010011010110110100, which is the same as what Alice had originally.

b.
If Trudy gets access to the output of the key stream, he can set up a sequence of linear equations. Using the public initial value, 00110110010101101111 he knows that:
s20 = 0(p0) + 0(p1) + 1(p2) + 1(p3) + 0(p4) + 1(p5) + 1(p6) + 0(p7) + 0(p8) + 1(p9) + 0(p10) + 1(p11) + 0(p12) + 1(p13) + 1(p14) + 0(p15) + 1(p16) + 1(p17) + 1(p18) + 1(p19) mod 2
or generically,
$s_m = s_{m-20}p_0 + s_{m-19}p_1 + \ldots + s_{m-2}p_{18} + s_{m-1}p_{19}$ (mod 2)
using this, he can set up a system of linear equations with 2m bits that can be solved to find the coefficients. This is as follows:
s20 = s0p0 + s1p1 + s2p2 + s3p3 + s4p4 + s5p5 + s6p6 + s7p7 + s8p8 + s9p9 + s10p10 + s11p11 + s12p12 + s13p13 + s14p14 + s15p15 + s16p16 + s17p17 + s18p18 + s19p19 (mod 2)
s21 = s1p0 + s2p1 + s3p2 + s4p3 + s5p4 + s6p5 + s7p6 + s8p7 + s9p8 + s10p9 + s11p10 + s12p11 + s13p12 + s14p13 + s15p14 + s16p15 + s17p16 + s18p17 + s19p18 + s20p19 (mod 2)
(...)
s38 = s18p0 + s19p1 + s20p2 + s21p3 + s22p4 + s23p5 + s24p6 + s25p7 + s26p8 + s27p9 + s28p10 + s29p11 + s30p12 + s31p13 + s32p14 + s33p15 + s34p16 + s35p17 + s36p18 + s37p19 (mod 2)
s39 = s19p0 + s20p1 + s21p2 + s22p3 + s23p4 + s24p5 + s25p6 + s26p7 + s27p8 + s28p9 + s29p10 + s30p11 + s31p12 + s32p13 + s33p14 + s34p15 + s35p16 + s36p17 + s37p18 + s38p19 (mod 2)

After solving this, he will get the coefficients:
$p_0=1$, $p_1=1$, $p_2=0$, $p_3=1$, $p_4=0$, $p_5=1$, $p_6=0$, $p_7=0$, $p_8=0$, $p_9=1$, $p_{10}=0$, $p_{11}=0$, $p_{12}=0$, $p_{13}=1$, $p_{14}=1$, $p_{15}=0$, $p_{16}=1$, $p_{17}=0$, $p_{18}=1$, $p_{19}=0$
Therefore, Trudy will need 2m, or 40 in this case, bits to compute the secret coefficients.

1c.
Yes, Trudy can still perform the same attack as in part 1b, if he has enough of the plaintext. If Trudy has access to the plaintext and ciphertext, he can XOR the bits to get the keystream using the formula:
$s_i = x_i + y_i$ mod 2
where x and y are the ciphertext and plaintext, respectively.
Once Trudy has the 2m keybits, as shown in part 1b, he can get the secret coefficients.

2.
Yes, an affine cipher is vulnerable to the frequency analysis attack. Since the cipher is a monoalphabetic substitution cipher, each character in the original text is mapped to a character of the alphabet in the ciphertext. This means that after being encrypted, the frequency distribution of the letters does not change. If the attacker knows the alphabet and the frequency distribution of the plaintext, they are still able to match it with the frequency distribution of the ciphertext to get a mapping between characters. Starting by assuming from the most frequent character, the attacker is able to use the affine cipher formula to solve for possible (a, b) pairs:
e(k) = (ax + b) mod m

m is known as it is the length of the alphabet, so the attacker can assume the most frequent character in plaintext k, is substituted by (ax+b) mod m. From here, the attacker can solve for all pairs of (a,b). They can then test if any of these pairs create a sensible plaintext when used to decode the ciphertext. This means the attacker has broken the cipher.

Full Steps:

1a.

a20= 0 + 0 + 1 + 1 + 1 + 1 + 1 + 1 + 1 (mod 2) = 1
a21= a1 + a2 + a4 + a6 + a10 + a14 + a15 + a17 + a19 (mod 2)
a21= 0 + 1 + 0 + 1 + 0 + 1 + 0 + 1 + 1 (mod 2) = 1
a22= a2 + a3 + a5 + a7 + a11 + a15 + a16 + a18 + a20 (mod 2)
a22= 1 + 1 + 1 + 0 + 1 + 0 + 1 + 1 + 1 (mod 2) = 1
a23= a3 + a4 + a6 + a8 + a12 + a16 + a17 + a19 + a21 (mod 2)
a23= 1 + 0 + 1 + 0 + 0 + 1 + 1 + 1 + 1 (mod 2) = 0
a24= a4 + a5 + a7 + a9 + a13 + a17 + a18 + a20 + a22 (mod 2)
a24= 0 + 1 + 0 + 1 + 1 + 1 + 1 + 1 + 1 (mod 2) = 1
a25= a5 + a6 + a8 + a10 + a14 + a18 + a19 + a21 + a23 (mod 2)
a25= 1 + 1 + 0 + 0 + 1 + 1 + 1 + 1 + 0 (mod 2) = 0
a26= a6 + a7 + a9 + a11 + a15 + a19 + a20 + a22 + a24 (mod 2)
a26= 1 + 0 + 1 + 1 + 0 + 1 + 1 + 1 + 1 (mod 2) = 1
a27= a7 + a8 + a10 + a12 + a16 + a20 + a21 + a23 + a25 (mod 2)
a27= 0 + 0 + 0 + 0 + 1 + 1 + 1 + 0 + 0 (mod 2) = 1
a28= a8 + a9 + a11 + a13 + a17 + a21 + a22 + a24 + a26 (mod 2)
a28= 0 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 (mod 2) = 0
a29= a9 + a10 + a12 + a14 + a18 + a22 + a23 + a25 + a27 (mod 2)
a29= 1 + 0 + 0 + 1 + 1 + 1 + 0 + 0 + 1 (mod 2) = 1
a30= a10 + a11 + a13 + a15 + a19 + a23 + a24 + a26 + a28 (mod 2)
a30= 0 + 1 + 1 + 0 + 1 + 0 + 1 + 1 + 0 (mod 2) = 1
a31= a11 + a12 + a14 + a16 + a20 + a24 + a25 + a27 + a29 (mod 2)
a31= 1 + 0 + 1 + 1 + 1 + 1 + 0 + 1 + 1 (mod 2) = 1
a32= a12 + a13 + a15 + a17 + a21 + a25 + a26 + a28 + a30 (mod 2)
a32= 0 + 1 + 0 + 1 + 1 + 0 + 1 + 0 + 1 (mod 2) = 1
a33= a13 + a14 + a16 + a18 + a22 + a26 + a27 + a29 + a31 (mod 2)
a33= 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 (mod 2) = 1
a34= a14 + a15 + a17 + a19 + a23 + a27 + a28 + a30 + a32 (mod 2)
a34= 1 + 0 + 1 + 1 + 0 + 1 + 0 + 1 + 1 (mod 2) = 0
a35= a15 + a16 + a18 + a20 + a24 + a28 + a29 + a31 + a33 (mod 2)
a35= 0 + 1 + 1 + 1 + 1 + 0 + 1 + 1 + 1 (mod 2) = 1
a36= a16 + a17 + a19 + a21 + a25 + a29 + a30 + a32 + a34 (mod 2)
a36= 1 + 1 + 1 + 1 + 0 + 1 + 1 + 1 + 0 (mod 2) = 1
a37= a17 + a18 + a20 + a22 + a26 + a30 + a31 + a33 + a35 (mod 2)
a37= 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 (mod 2) = 1
a38= a18 + a19 + a21 + a23 + a27 + a31 + a32 + a34 + a36 (mod 2)
a38= 1 + 1 + 1 + 0 + 1 + 1 + 1 + 0 + 1 (mod 2) = 1
a39= a19 + a20 + a22 + a24 + a28 + a32 + a33 + a35 + a37 (mod 2)
a39= 1 + 1 + 1 + 1 + 0 + 1 + 1 + 1 + 1 (mod 2) = 0
a40= a20 + a21 + a23 + a25 + a29 + a33 + a34 + a36 + a38 (mod 2)
a40= 1 + 1 + 0 + 0 + 1 + 1 + 0 + 1 + 1 (mod 2) = 0
a41= a21 + a22 + a24 + a26 + a30 + a34 + a35 + a37 + a39 (mod 2)

a41= 1 + 1 + 1 + 1 + 1 + 0 + 1 + 1 + 0 (mod 2) = 1
a42= a22 + a23 + a25 + a27 + a31 + a35 + a36 + a38 + a40 (mod 2)
a42= 1 + 0 + 0 + 1 + 1 + 1 + 1 + 1 + 0 (mod 2) = 0
a43= a23 + a24 + a26 + a28 + a32 + a36 + a37 + a39 + a41 (mod 2)
a43= 0 + 1 + 1 + 0 + 1 + 1 + 1 + 0 + 1 (mod 2) = 0
a44= a24 + a25 + a27 + a29 + a33 + a37 + a38 + a40 + a42 (mod 2)
a44= 1 + 0 + 1 + 1 + 1 + 1 + 1 + 0 + 0 (mod 2) = 0
a45= a25 + a26 + a28 + a30 + a34 + a38 + a39 + a41 + a43 (mod 2)
a45= 0 + 1 + 0 + 1 + 0 + 1 + 0 + 1 + 0 (mod 2) = 0
a46= a26 + a27 + a29 + a31 + a35 + a39 + a40 + a42 + a44 (mod 2)
a46= 1 + 1 + 1 + 1 + 1 + 0 + 0 + 0 + 0 (mod 2) = 1
a47= a27 + a28 + a30 + a32 + a36 + a40 + a41 + a43 + a45 (mod 2)
a47= 1 + 0 + 1 + 1 + 1 + 0 + 1 + 0 + 0 (mod 2) = 1
a48= a28 + a29 + a31 + a33 + a37 + a41 + a42 + a44 + a46 (mod 2)
a48= 0 + 1 + 1 + 1 + 1 + 1 + 0 + 0 + 1 (mod 2) = 0
a49= a29 + a30 + a32 + a34 + a38 + a42 + a43 + a45 + a47 (mod 2)
a49= 1 + 1 + 1 + 0 + 1 + 0 + 0 + 0 + 1 (mod 2) = 1

XORing generated bits with plaintext bits
bit 0 of ciphertext = 0 XOR 0 = 0
bit 1 of ciphertext = 0 XOR 0 = 0
bit 2 of ciphertext = 1 XOR 1 = 0
bit 3 of ciphertext = 1 XOR 1 = 0
bit 4 of ciphertext = 0 XOR 1 = 1
bit 5 of ciphertext = 1 XOR 1 = 0
bit 6 of ciphertext = 1 XOR 0 = 1
bit 7 of ciphertext = 0 XOR 1 = 1
bit 8 of ciphertext = 0 XOR 0 = 0
bit 9 of ciphertext = 1 XOR 1 = 0
bit 10 of ciphertext = 0 XOR 1 = 1
bit 11 of ciphertext = 1 XOR 1 = 0
bit 12 of ciphertext = 0 XOR 0 = 0
bit 13 of ciphertext = 1 XOR 0 = 1
bit 14 of ciphertext = 1 XOR 1 = 0
bit 15 of ciphertext = 0 XOR 1 = 1
bit 16 of ciphertext = 1 XOR 1 = 0
bit 17 of ciphertext = 1 XOR 0 = 1
bit 18 of ciphertext = 1 XOR 0 = 1
bit 19 of ciphertext = 1 XOR 0 = 1
bit 20 of ciphertext = 1 XOR 0 = 1
bit 21 of ciphertext = 1 XOR 0 = 1
bit 22 of ciphertext = 1 XOR 0 = 1
bit 23 of ciphertext = 0 XOR 1 = 1
bit 24 of ciphertext = 1 XOR 1 = 0

bit 25 of ciphertext = 0 XOR 1 = 1
bit 26 of ciphertext = 1 XOR 0 = 1
bit 27 of ciphertext = 1 XOR 0 = 1
bit 28 of ciphertext = 0 XOR 1 = 1
bit 29 of ciphertext = 1 XOR 1 = 0
bit 30 of ciphertext = 1 XOR 0 = 1
bit 31 of ciphertext = 1 XOR 0 = 1
bit 32 of ciphertext = 1 XOR 0 = 1
bit 33 of ciphertext = 1 XOR 1 = 0
bit 34 of ciphertext = 0 XOR 0 = 0
bit 35 of ciphertext = 1 XOR 0 = 1
bit 36 of ciphertext = 1 XOR 1 = 0
bit 37 of ciphertext = 1 XOR 1 = 0
bit 38 of ciphertext = 1 XOR 0 = 1
bit 39 of ciphertext = 0 XOR 1 = 1
bit 40 of ciphertext = 0 XOR 0 = 0
bit 41 of ciphertext = 1 XOR 1 = 0
bit 42 of ciphertext = 0 XOR 1 = 1
bit 43 of ciphertext = 0 XOR 0 = 0
bit 44 of ciphertext = 0 XOR 1 = 1
bit 45 of ciphertext = 0 XOR 1 = 1
bit 46 of ciphertext = 1 XOR 0 = 1
bit 47 of ciphertext = 1 XOR 1 = 0
bit 48 of ciphertext = 0 XOR 0 = 0
bit 49 of ciphertext = 1 XOR 0 = 1

XORing generated bits with ciphertext bits:
bit 0 of plaintext = 0 XOR 0 = 0
bit 1 of plaintext = 0 XOR 0 = 0
bit 2 of plaintext = 1 XOR 0 = 1
bit 3 of plaintext = 1 XOR 0 = 1
bit 4 of plaintext = 0 XOR 1 = 1
bit 5 of plaintext = 1 XOR 0 = 1
bit 6 of plaintext = 1 XOR 1 = 0
bit 7 of plaintext = 0 XOR 1 = 1
bit 8 of plaintext = 0 XOR 0 = 0
bit 9 of plaintext = 1 XOR 0 = 1
bit 10 of plaintext = 0 XOR 1 = 1
bit 11 of plaintext = 1 XOR 0 = 1
bit 12 of plaintext = 0 XOR 0 = 0
bit 13 of plaintext = 1 XOR 1 = 0
bit 14 of plaintext = 1 XOR 0 = 1
bit 15 of plaintext = 0 XOR 1 = 1
bit 16 of plaintext = 1 XOR 0 = 1

bit 17 of plaintext = 1 XOR 1 = 0
bit 18 of plaintext = 1 XOR 1 = 0
bit 19 of plaintext = 1 XOR 1 = 0
bit 20 of plaintext = 1 XOR 1 = 0
bit 21 of plaintext = 1 XOR 1 = 0
bit 22 of plaintext = 1 XOR 1 = 0
bit 23 of plaintext = 0 XOR 1 = 1
bit 24 of plaintext = 1 XOR 0 = 1
bit 25 of plaintext = 0 XOR 1 = 1
bit 26 of plaintext = 1 XOR 1 = 0
bit 27 of plaintext = 1 XOR 1 = 0
bit 28 of plaintext = 0 XOR 1 = 1
bit 29 of plaintext = 1 XOR 0 = 1
bit 30 of plaintext = 1 XOR 1 = 0
bit 31 of plaintext = 1 XOR 1 = 0
bit 32 of plaintext = 1 XOR 1 = 0
bit 33 of plaintext = 1 XOR 0 = 1
bit 34 of plaintext = 0 XOR 0 = 0
bit 35 of plaintext = 1 XOR 1 = 0
bit 36 of plaintext = 1 XOR 0 = 1
bit 37 of plaintext = 1 XOR 0 = 1
bit 38 of plaintext = 1 XOR 1 = 0
bit 39 of plaintext = 0 XOR 1 = 1
bit 40 of plaintext = 0 XOR 0 = 0
bit 41 of plaintext = 1 XOR 0 = 1
bit 42 of plaintext = 0 XOR 1 = 1
bit 43 of plaintext = 0 XOR 0 = 0
bit 44 of plaintext = 0 XOR 1 = 1
bit 45 of plaintext = 0 XOR 1 = 1
bit 46 of plaintext = 1 XOR 1 = 0
bit 47 of plaintext = 1 XOR 0 = 1
bit 48 of plaintext = 0 XOR 0 = 0
bit 49 of plaintext = 1 XOR 1 = 0

1b.
Full system of equations:
s20 = s0p0 + s1p1 + s2p2 + s3p3 + s4p4 + s5p5 + s6p6 + s7p7 + s8p8 + s9p9 + s10p10 + s11p11 + s12p12 + s13p13 + s14p14 + s15p15 + s16p16 + s17p17 + s18p18 + s19p19 (mod 2)
s21 = s1p0 + s2p1 + s3p2 + s4p3 + s5p4 + s6p5 + s7p6 + s8p7 + s9p8 + s10p9 + s11p10 + s12p11 + s13p12 + s14p13 + s15p14 + s16p15 + s17p16 + s18p17 + s19p18 + s20p19 (mod 2)
s22 = s2p0 + s3p1 + s4p2 + s5p3 + s6p4 + s7p5 + s8p6 + s9p7 + s10p8 + s11p9 + s12p10 + s13p11 + s14p12 + s15p13 + s16p14 + s17p15 + s18p16 + s19p17 + s20p18 + s21p19 (mod 2)
s23 = s3p0 + s4p1 + s5p2 + s6p3 + s7p4 + s8p5 + s9p6 + s10p7 + s11p8 + s12p9 + s13p10 + s14p11 + s15p12 + s16p13 + s17p14 + s18p15 + s19p16 + s20p17 + s21p18 + s22p19 (mod 2)

s24 = s4p0 + s5p1 + s6p2 + s7p3 + s8p4 + s9p5 + s10p6 + s11p7 + s12p8 + s13p9 + s14p10 + s15p11 + s16p12 + s17p13 + s18p14 + s19p15 + s20p16 + s21p17 + s22p18 + s23p19 (mod 2)

s25 = s5p0 + s6p1 + s7p2 + s8p3 + s9p4 + s10p5 + s11p6 + s12p7 + s13p8 + s14p9 + s15p10 + s16p11 + s17p12 + s18p13 + s19p14 + s20p15 + s21p16 + s22p17 + s23p18 + s24p19 (mod 2)

s26 = s6p0 + s7p1 + s8p2 + s9p3 + s10p4 + s11p5 + s12p6 + s13p7 + s14p8 + s15p9 + s16p10 + s17p11 + s18p12 + s19p13 + s20p14 + s21p15 + s22p16 + s23p17 + s24p18 + s25p19 (mod 2)

s27 = s7p0 + s8p1 + s9p2 + s10p3 + s11p4 + s12p5 + s13p6 + s14p7 + s15p8 + s16p9 + s17p10 + s18p11 + s19p12 + s20p13 + s21p14 + s22p15 + s23p16 + s24p17 + s25p18 + s26p19 (mod 2)

s28 = s8p0 + s9p1 + s10p2 + s11p3 + s12p4 + s13p5 + s14p6 + s15p7 + s16p8 + s17p9 + s18p10 + s19p11 + s20p12 + s21p13 + s22p14 + s23p15 + s24p16 + s25p17 + s26p18 + s27p19 (mod 2)

s29 = s9p0 + s10p1 + s11p2 + s12p3 + s13p4 + s14p5 + s15p6 + s16p7 + s17p8 + s18p9 + s19p10 + s20p11 + s21p12 + s22p13 + s23p14 + s24p15 + s25p16 + s26p17 + s27p18 + s28p19 (mod 2)

s30 = s10p0 + s11p1 + s12p2 + s13p3 + s14p4 + s15p5 + s16p6 + s17p7 + s18p8 + s19p9 + s20p10 + s21p11 + s22p12 + s23p13 + s24p14 + s25p15 + s26p16 + s27p17 + s28p18 + s29p19 (mod 2)

s31 = s11p0 + s12p1 + s13p2 + s14p3 + s15p4 + s16p5 + s17p6 + s18p7 + s19p8 + s20p9 + s21p10 + s22p11 + s23p12 + s24p13 + s25p14 + s26p15 + s27p16 + s28p17 + s29p18 + s30p19 (mod 2)

s32 = s12p0 + s13p1 + s14p2 + s15p3 + s16p4 + s17p5 + s18p6 + s19p7 + s20p8 + s21p9 + s22p10 + s23p11 + s24p12 + s25p13 + s26p14 + s27p15 + s28p16 + s29p17 + s30p18 + s31p19 (mod 2)

s33 = s13p0 + s14p1 + s15p2 + s16p3 + s17p4 + s18p5 + s19p6 + s20p7 + s21p8 + s22p9 + s23p10 + s24p11 + s25p12 + s26p13 + s27p14 + s28p15 + s29p16 + s30p17 + s31p18 + s32p19 (mod 2)

s34 = s14p0 + s15p1 + s16p2 + s17p3 + s18p4 + s19p5 + s20p6 + s21p7 + s22p8 + s23p9 + s24p10 + s25p11 + s26p12 + s27p13 + s28p14 + s29p15 + s30p16 + s31p17 + s32p18 + s33p19 (mod 2)

s35 = s15p0 + s16p1 + s17p2 + s18p3 + s19p4 + s20p5 + s21p6 + s22p7 + s23p8 + s24p9 + s25p10 + s26p11 + s27p12 + s28p13 + s29p14 + s30p15 + s31p16 + s32p17 + s33p18 + s34p19 (mod 2)

s36 = s16p0 + s17p1 + s18p2 + s19p3 + s20p4 + s21p5 + s22p6 + s23p7 + s24p8 + s25p9 + s26p10 + s27p11 + s28p12 + s29p13 + s30p14 + s31p15 + s32p16 + s33p17 + s34p18 + s35p19 (mod 2)

s37 = s17p0 + s18p1 + s19p2 + s20p3 + s21p4 + s22p5 + s23p6 + s24p7 + s25p8 + s26p9 + s27p10 + s28p11 + s29p12 + s30p13 + s31p14 + s32p15 + s33p16 + s34p17 + s35p18 + s36p19 (mod 2)

s38 = s18p0 + s19p1 + s20p2 + s21p3 + s22p4 + s23p5 + s24p6 + s25p7 + s26p8 + s27p9 + s28p10 + s29p11 + s30p12 + s31p13 + s32p14 + s33p15 + s34p16 + s35p17 + s36p18 + s37p19 (mod 2)

$s39 = s19p0 + s20p1 + s21p2 + s22p3 + s23p4 + s24p5 + s25p6 + s26p7 + s27p8 + s28p9 + s29p10 + s30p11 + s31p12 + s32p13 + s33p14 + s34p15 + s35p16 + s36p17 + s37p18 + s38p19 \pmod 2$