Problem 1

a) Yes, we can see the coefficients of the power of x below has 2^8-$1 = 255$ different combinations, so it's a primitive polynomial. I use LFSR(8) to calculate the coefficients. When the first bit is 0, simply shift the bits left; while the first bit is 1, shift the bits left then xor the sequence [100011101] (from the polynomial $x^8 + x^4 + x^3 + x^2 + 1 = 0$).

```
1: [0, 0, 0, 0, 0, 0, 0, 1]
2: [0, 0, 0, 0, 0, 0, 1, 0]
3: [0, 0, 0, 0, 0, 1, 0, 0]
4: [0, 0, 0, 0, 1, 0, 0, 0]
5: [0, 0, 0, 1, 0, 0, 0, 0]
6: [0, 0, 1, 0, 0, 0, 0, 0]
7: [0, 1, 0, 0, 0, 0, 0, 0]
8: [1, 0, 0, 0, 0, 0, 0, 0]
9: [0, 0, 0, 1, 1, 1, 0, 1]
10: [0, 0, 1, 1, 1, 0, 1, 0]
11: [0, 1, 1, 1, 0, 1, 0, 0]
12: [1, 1, 1, 0, 1, 0, 0, 0]
13: [1, 1, 0, 0, 1, 1, 0, 1]
14: [1, 0, 0, 0, 0, 1, 1, 1]
15: [0, 0, 0, 1, 0, 0, 1, 1]
16: [0, 0, 1, 0, 0, 1, 1, 0]
17: [0, 1, 0, 0, 1, 1, 0, 0]
18: [1, 0, 0, 1, 1, 0, 0, 0]
19: [0, 0, 1, 0, 1, 1, 0, 1]
20: [0, 1, 0, 1, 1, 0, 1, 0]
21: [1, 0, 1, 1, 0, 1, 0, 0]
22: [0, 1, 1, 1, 0, 1, 0, 1]
23: [1, 1, 1, 0, 1, 0, 1, 0]
24: [1, 1, 0, 0, 1, 0, 0, 1]
25: [1, 0, 0, 0, 1, 1, 1, 1]
26: [0, 0, 0, 0, 0, 0, 1, 1]
27: [0, 0, 0, 0, 0, 1, 1, 0]
28: [0, 0, 0, 0, 1, 1, 0, 0]
29: [0, 0, 0, 1, 1, 0, 0, 0]
30: [0, 0, 1, 1, 0, 0, 0, 0]
31: [0, 1, 1, 0, 0, 0, 0, 0]
32: [1, 1, 0, 0, 0, 0, 0, 0]
33: [1, 0, 0, 1, 1, 1, 0, 1]
34: [0, 0, 1, 0, 0, 1, 1, 1]
35: [0, 1, 0, 0, 1, 1, 1, 0]
36: [1, 0, 0, 1, 1, 1, 0, 0]
37: [0, 0, 1, 0, 0, 1, 0, 1]
38: [0, 1, 0, 0, 1, 0, 1, 0]
39: [1, 0, 0, 1, 0, 1, 0, 0]
40: [0, 0, 1, 1, 0, 1, 0, 1]
41: [0, 1, 1, 0, 1, 0, 1, 0]
42: [1, 1, 0, 1, 0, 1, 0, 0]
43: [1, 0, 1, 1, 0, 1, 0, 1]
44: [0, 1, 1, 1, 0, 1, 1, 1]
45: [1, 1, 1, 0, 1, 1, 1, 0]
46: [1, 1, 0, 0, 0, 0, 0, 1]
47: [1, 0, 0, 1, 1, 1, 1, 1]
48: [0, 0, 1, 0, 0, 0, 1, 1]
49: [0, 1, 0, 0, 0, 1, 1, 0]
50: [1, 0, 0, 0, 1, 1, 0, 0]
51: [0, 0, 0, 0, 0, 1, 0, 1]
52: [0, 0, 0, 0, 1, 0, 1, 0]
53: [0, 0, 0, 1, 0, 1, 0, 0]
54: [0, 0, 1, 0, 1, 0, 0, 0]
55: [0, 1, 0, 1, 0, 0, 0, 0]
56: [1, 0, 1, 0, 0, 0, 0, 0]
57: [0, 1, 0, 1, 1, 1, 0, 1]
58: [1, 0, 1, 1, 1, 0, 1, 0]
59: [0, 1, 1, 0, 1, 0, 0, 1]
60: [1, 1, 0, 1, 0, 0, 1, 0]
61: [1, 0, 1, 1, 1, 0, 0, 1]
62: [0, 1, 1, 0, 1, 1, 1, 1]
63: [1, 1, 0, 1, 1, 1, 1, 0]
64: [1, 0, 1, 0, 0, 0, 0, 1]
65: [0, 1, 0, 1, 1, 1, 1, 1]
66: [1, 0, 1, 1, 1, 1, 1, 0]
67: [0, 1, 1, 0, 0, 0, 0, 1]
68: [1, 1, 0, 0, 0, 0, 1, 0]
69: [1, 0, 0, 1, 1, 0, 0, 1]
70: [0, 0, 1, 0, 1, 1, 1, 1]
71: [0, 1, 0, 1, 1, 1, 1, 0]
72: [1, 0, 1, 1, 1, 1, 0, 0]
73: [0, 1, 1, 0, 0, 1, 0, 1]
74: [1, 1, 0, 0, 1, 0, 1, 0]
75: [1, 0, 0, 0, 1, 0, 0, 1]
76: [0, 0, 0, 0, 1, 1, 1, 1]
77: [0, 0, 0, 1, 1, 1, 1, 0]
78: [0, 0, 1, 1, 1, 1, 0, 0]
79: [0, 1, 1, 1, 1, 0, 0, 0]
80: [1, 1, 1, 1, 0, 0, 0, 0]
81: [1, 1, 1, 1, 1, 1, 0, 1]
82: [1, 1, 1, 0, 0, 1, 1, 1]
83: [1, 1, 0, 1, 0, 0, 1, 1]
84: [1, 0, 1, 1, 1, 0, 1, 1]
85: [0, 1, 1, 0, 1, 0, 1, 1]
86: [1, 1, 0, 1, 0, 1, 1, 0]
87: [1, 0, 1, 1, 0, 0, 0, 1]
88: [0, 1, 1, 1, 1, 1, 1, 1]
89: [1, 1, 1, 1, 1, 1, 1, 0]
90: [1, 1, 1, 0, 0, 0, 0, 1]
91: [1, 1, 0, 1, 1, 1, 1, 1]
92: [1, 0, 1, 0, 0, 0, 1, 1]
93: [0, 1, 0, 1, 1, 0, 1, 1]
94: [1, 0, 1, 1, 0, 1, 1, 0]
95: [0, 1, 1, 1, 0, 0, 0, 1]
96: [1, 1, 1, 0, 0, 0, 1, 0]
97: [1, 1, 0, 1, 1, 0, 0, 1]
98: [1, 0, 1, 0, 1, 1, 1, 1]
99: [0, 1, 0, 0, 0, 0, 1, 1]
100: [1, 0, 0, 0, 0, 1, 1, 0]
101: [0, 0, 0, 1, 0, 0, 0, 1]
102: [0, 0, 1, 0, 0, 0, 1, 0]
103: [0, 1, 0, 0, 0, 1, 0, 0]
104: [1, 0, 0, 0, 1, 0, 0, 0]
105: [0, 0, 0, 0, 1, 1, 0, 1]
106: [0, 0, 0, 1, 1, 0, 1, 0]
107: [0, 0, 1, 1, 0, 1, 0, 0]
108: [0, 1, 1, 0, 1, 0, 0, 0]
109: [1, 1, 0, 1, 0, 0, 0, 0]
110: [1, 0, 1, 1, 1, 1, 0, 1]
111: [0, 1, 1, 0, 0, 1, 1, 1]
112: [1, 1, 0, 0, 1, 1, 1, 0]
113: [1, 0, 0, 0, 0, 0, 0, 1]
114: [0, 0, 0, 1, 1, 1, 1, 1]
115: [0, 0, 1, 1, 1, 1, 1, 0]
116: [0, 1, 1, 1, 1, 1, 0, 0]
117: [1, 1, 1, 1, 1, 0, 0, 0]
118: [1, 1, 1, 0, 1, 1, 0, 1]
119: [1, 1, 0, 0, 0, 1, 1, 1]
120: [1, 0, 0, 1, 0, 0, 1, 1]
121: [0, 0, 1, 1, 1, 0, 1, 1]
122: [0, 1, 1, 1, 0, 1, 1, 0]
123: [1, 1, 1, 0, 1, 1, 0, 0]
124: [1, 1, 0, 0, 0, 1, 0, 1]
125: [1, 0, 0, 1, 0, 1, 1, 1]
126: [0, 0, 1, 1, 0, 0, 1, 1]
127: [0, 1, 1, 0, 0, 1, 1, 0]
128: [1, 1, 0, 0, 1, 1, 0, 0]
129: [1, 0, 0, 0, 0, 1, 0, 1]
130: [0, 0, 0, 1, 0, 1, 1, 1]
131: [0, 0, 1, 0, 1, 1, 1, 0]
```

```
132: [0, 1, 0, 1, 1, 1, 0, 0]
133: [1, 0, 1, 1, 1, 0, 0, 0]
134: [0, 1, 1, 0, 1, 1, 0, 1]
135: [1, 1, 0, 1, 1, 0, 1, 0]
136: [1, 0, 1, 0, 1, 0, 0, 1]
137: [0, 1, 0, 0, 1, 1, 1, 1]
138: [1, 0, 0, 1, 1, 1, 1, 0]
139: [0, 0, 1, 0, 0, 0, 0, 1]
140: [0, 1, 0, 0, 0, 0, 1, 0]
141: [1, 0, 0, 0, 0, 1, 0, 0]
142: [0, 0, 0, 1, 0, 1, 0, 1]
143: [0, 0, 1, 0, 1, 0, 1, 0]
144: [0, 1, 0, 1, 0, 1, 0, 0]
145: [1, 0, 1, 0, 1, 0, 0, 0]
146: [0, 1, 0, 0, 1, 1, 0, 1]
147: [1, 0, 0, 1, 1, 0, 1, 0]
148: [0, 0, 1, 0, 1, 0, 0, 1]
149: [0, 1, 0, 1, 0, 0, 1, 0]
150: [1, 0, 1, 0, 0, 1, 0, 0]
151: [0, 1, 0, 1, 0, 1, 0, 1]
152: [1, 0, 1, 0, 1, 0, 1, 0]
153: [0, 1, 0, 0, 1, 0, 0, 1]
154: [1, 0, 0, 1, 0, 0, 1, 0]
155: [0, 0, 1, 1, 1, 0, 0, 1]
156: [0, 1, 1, 1, 0, 0, 1, 0]
157: [1, 1, 1, 0, 0, 1, 0, 0]
158: [1, 1, 0, 1, 0, 1, 0, 1]
159: [1, 0, 1, 1, 0, 1, 1, 1]
160: [0, 1, 1, 1, 0, 0, 1, 1]
161: [1, 1, 1, 0, 0, 1, 1, 0]
162: [1, 1, 0, 1, 0, 0, 0, 1]
163: [1, 0, 1, 1, 1, 1, 1, 1]
164: [0, 1, 1, 0, 0, 0, 1, 1]
165: [1, 1, 0, 0, 0, 1, 1, 0]
166: [1, 0, 0, 1, 0, 0, 0, 1]
167: [0, 0, 1, 1, 1, 1, 1, 1]
168: [0, 1, 1, 1, 1, 1, 1, 0]
169: [1, 1, 1, 1, 1, 1, 0, 0]
170: [1, 1, 1, 0, 0, 1, 0, 1]
171: [1, 1, 0, 1, 0, 1, 1, 1]
172: [1, 0, 1, 1, 0, 0, 1, 1]
173: [0, 1, 1, 1, 1, 0, 1, 1]
174: [1, 1, 1, 1, 0, 1, 1, 0]
175: [1, 1, 1, 1, 0, 0, 0, 1]
176: [1, 1, 1, 1, 1, 1, 1, 1]
177: [1, 1, 1, 0, 0, 0, 1, 1]
178: [1, 1, 0, 1, 1, 0, 1, 1]
179: [1, 0, 1, 0, 1, 0, 1, 1]
180: [0, 1, 0, 0, 1, 0, 1, 1]
181: [1, 0, 0, 1, 0, 1, 1, 0]
182: [0, 0, 1, 1, 0, 0, 0, 1]
183: [0, 1, 1, 0, 0, 0, 1, 0]
184: [1, 1, 0, 0, 0, 1, 0, 0]
185: [1, 0, 0, 1, 0, 1, 0, 1]
186: [0, 0, 1, 1, 0, 1, 1, 1]
187: [0, 1, 1, 0, 1, 1, 1, 0]
188: [1, 1, 0, 1, 1, 1, 0, 0]
189: [1, 0, 1, 0, 0, 1, 0, 1]
190: [0, 1, 0, 1, 0, 1, 1, 1]
191: [1, 0, 1, 0, 1, 1, 1, 0]
192: [0, 1, 0, 0, 0, 0, 0, 1]
193: [1, 0, 0, 0, 0, 0, 1, 0]
194: [0, 0, 0, 1, 1, 0, 0, 1]
195: [0, 0, 1, 1, 0, 0, 1, 0]
196: [0, 1, 1, 0, 0, 1, 0, 0]
197: [1, 1, 0, 0, 1, 0, 0, 0]
198: [1, 0, 0, 0, 1, 1, 0, 1]
199: [0, 0, 0, 0, 0, 1, 1, 1]
200: [0, 0, 0, 0, 1, 1, 1, 0]
201: [0, 0, 0, 1, 1, 1, 0, 0]
202: [0, 0, 1, 1, 1, 0, 0, 0]
203: [0, 1, 1, 1, 0, 0, 0, 0]
204: [1, 1, 1, 0, 0, 0, 0, 0]
205: [1, 1, 0, 1, 1, 1, 0, 1]
206: [1, 0, 1, 0, 0, 1, 1, 1]
207: [0, 1, 0, 1, 0, 0, 1, 1]
208: [1, 0, 1, 0, 0, 1, 1, 0]
209: [0, 1, 0, 1, 0, 0, 0, 1]
210: [1, 0, 1, 0, 0, 0, 1, 0]
211: [0, 1, 0, 1, 1, 0, 0, 1]
212: [1, 0, 1, 1, 0, 0, 1, 0]
213: [0, 1, 1, 1, 1, 0, 0, 1]
214: [1, 1, 1, 1, 0, 0, 1, 0]
215: [1, 1, 1, 1, 1, 0, 0, 1]
216: [1, 1, 1, 0, 1, 1, 1, 1]
217: [1, 1, 0, 0, 0, 0, 1, 1]
218: [1, 0, 0, 1, 1, 0, 1, 1]
219: [0, 0, 1, 0, 1, 0, 1, 1]
220: [0, 1, 0, 1, 0, 1, 1, 0]
221: [1, 0, 1, 0, 1, 1, 0, 0]
222: [0, 1, 0, 0, 0, 1, 0, 1]
223: [1, 0, 0, 0, 1, 0, 1, 0]
224: [0, 0, 0, 0, 1, 0, 0, 1]
225: [0, 0, 0, 1, 0, 0, 1, 0]
226: [0, 0, 1, 0, 0, 1, 0, 0]
227: [0, 1, 0, 0, 1, 0, 0, 0]
228: [1, 0, 0, 1, 0, 0, 0, 0]
229: [0, 0, 1, 1, 1, 1, 0, 1]
230: [0, 1, 1, 1, 1, 0, 1, 0]
231: [1, 1, 1, 1, 0, 1, 0, 0]
232: [1, 1, 1, 1, 0, 1, 0, 1]
233: [1, 1, 1, 1, 0, 1, 1, 1]
234: [1, 1, 1, 1, 0, 0, 1, 1]
235: [1, 1, 1, 1, 1, 0, 1, 1]
236: [1, 1, 1, 0, 1, 0, 1, 1]
237: [1, 1, 0, 0, 1, 0, 1, 1]
238: [1, 0, 0, 0, 1, 0, 1, 1]
239: [0, 0, 0, 0, 1, 0, 1, 1]
240: [0, 0, 0, 1, 0, 1, 1, 0]
241: [0, 0, 1, 0, 1, 1, 0, 0]
242: [0, 1, 0, 1, 1, 0, 0, 0]
243: [1, 0, 1, 1, 0, 0, 0, 0]
244: [0, 1, 1, 1, 1, 1, 0, 1]
245: [1, 1, 1, 1, 1, 0, 1, 0]
246: [1, 1, 1, 0, 1, 0, 0, 1]
247: [1, 1, 0, 0, 1, 1, 1, 1]
248: [1, 0, 0, 0, 0, 0, 1, 1]
249: [0, 0, 0, 1, 1, 0, 1, 1]
250: [0, 0, 1, 1, 0, 1, 1, 0]
251: [0, 1, 1, 0, 1, 1, 0, 0]
252: [1, 1, 0, 1, 1, 0, 0, 0]
253: [1, 0, 1, 0, 1, 1, 0, 1]
254: [0, 1, 0, 0, 0, 1, 1, 1]
255: [1, 0, 0, 0, 1, 1, 1, 0]
```

b) 255, as it's a primitive polynomial, the result is shown above.

c) No, as the generator must be x, where is x is the root of the primitive polynomial = 0. For example, in this case x is the root of the polynomial $x^8 + x^4 + x^3 + x^2 + 1 = 0$, otherwise it won't form 255 different results.

Problem 2

a) We can use LFSR(8) (as mentioned in problem 1) to calculate the coefficients of each power of the generator x, once we calculate the coefficients, we can know how to calculate the i^{th} bit of the key stream. For example, x^{10} = [00111010], then we xor the initial key [00000001], we can derive the 10^{th}

bit of the key stream is 0, and so on. After the keystream is calculated the same long as the plaintext, we can calculate the ciphertext in binary (as shown below). We can use the same way to decrypt the ciphertext into binary plaintext, then finally convert into Ascii.

```
Problem 2-1:
Plaintext: ATNYCUWEARESTRIVINGTOBEAGREATUNIVERSITYTHATTRANSCENDSDISCIPLINARYDIVIDESTOSOLVETHEINCREASINGLYCOMPLEXPROBLEMSTHATTHEWORLDFACESWEWILLCONTINUETOBEGUIDED
BYTHEIDEATHATWECANACHIEVESOMETHINGMUCHGREATERTOGETHERTHANWECANINDIVIDUALLYAFTERALLTHATWASTHEIDEATHATLEDTOTHECREATIONOFOURUNIVERSITYINTHEFIRSTPLACE
--------------------------------------
Ciphertext in binary: 1100000111011010011010111001100110001010011000100111011111101000111011011110001010111110001010011011110011010100100011101100101010001100111
0101000010101111100110011100100111001101100011000110000000001010111001110111010100100010111011010101101011001110001000101011101011001000110011101001011011011 0011
1010000110000000011110001000100100000101000101010000110000011010011001100000101101010110010000000110111101011000010101011111110110011101000001100100111001111 01010
1000000011110000110010110101010110010011011100100100001011011011111100110110101010101101101100110001100011111100101111001101000110010111110101001001110111001 0011
0101011011001001110101111011010100000000111101001110011010101111011001011001110100010010110111001111100001110100110110101110100001000111010111101101001001001 0000
1011000100100001100010001111010010101011100100010001001111110010111001111010110100001011101100010011101001010011101111001011101101101110101111110111001001001 0000
1111010000100110011101000011001100011110011010110000101010100011111010110011010111111010101100010000000100000001101000000111101010001001000111010000001001000 0011
1100010111010111101110000100011001110111000100100011101000100000110001100111100100010111111011000011101000110100110100101000101111110110001100111011011101001 0000
1110011000010001011001100001011110110001001110011000111010011101010100111111111010111010111000010011100010100010101011000111010110110110010111111110111100011 1000
1000110111010100001110011011111010111111011111000110010000000001111111111010101111000100100001011010011100001011101111100110000100010011001100000000101000111 10110
1110001111111011001000111101110001100111011100110100111110000101010010110011000101100010101111110100010010011110101111011001101110111000110001101010111100011 10111
1010001011100111110110010011010111100011001000000110010101001011010000100000010011101000100011010100110100000110000110101111111100001000000000101000010001010 0111
0100110110100110100110110000011010111111110111100010111011001100101110101101100000000011011111101100000110111011010111111100010011101010011000110011000000100 0001
1100110110111001001100001111110011110010111111111110110101001101011010111010100110010010110011101001000011101010000110111111100010011001100101001101001111000 0100
0001001001010111111100010101010110101110110110110010100101100101101001000010111000011000110101011100001000100010000000000000110000001 1100
--------------------------------------
Decrypted plaintext in binary: 010000010101010001001110010110010100001101010101010101110100010101000001010100100100010101010011010101000101001001001001010101011001
00100101001110010001110101010001001111010000100100010101000001010001110101001001000101010000010101010001010101010011100100100101010110010001010101001001010011010
01001010101000101100101010100010010000100000101010100010101000101001001000001010011100101001101000011010001010100111001000100010100110100010001001001010100110100
00110100100101010000010011000100100101001110010000010101001001011001010001000100100101010110010010100010001000101010100110101010001001111010100110100111101001
10001010110010001010101010001001000010001010100100101001110010000110101001010001010100000010101001101001001010011100100011101001100010110010100001101001111010011
01010100000010011000100010101011000010100000101001001001111010000100100110010001010100110101010011010101000100100001000001010101000101010001001000010001010101011
10100111101010010010011000100010001000110010000001010000110100010101010011001011101000101010101110100100101001100010011000100001101001111010011100101010001001001
01001110010101010100010101010100010011110100001001000101010001110101010101001001010001000100010101000100010000100101100101010100010010000100010101001001010001000
10001010100000010101010001001000010000010101010001010111010001010100001101000010100111001000001010000110100100001001001010001010101011001000101010100110100111101
00110101000101010101000100100001001001010011100100011101001101010101010100001110100100001000111010100100100010101000001010101000100010101010010010101000100111010
00111010001010101010001001000010001010101001001010100010010000100000101001110010101110100010101000011010000010100111001001001010011100100010001001001010101100100
10010100010001010101010100000101001100010011000101100101000001010001100101001000101010100100100000101001100010011000101010001001000010000001010101000101011101000
00101010011010101010001001000010001010100100101000100010001010100000010101010010000100000010101010001001100010001010100010001010100010011110101010001001000010001
01010000110101001001000101010000010101010001001001010011110100111001001111000110010011110101010101010010010101010101001110010010010101011001000101010100100101001
1010010010101010001011001010010010100111001010100010010000100010101000110010010101001001010100110101010001010000010011000100000101000011 01000101
--------------------------------------
Decrypted plaintext: ATNYCUWEARESTRIVINGTOBEAGREATUNIVERSITYTHATTRANSCENDSDISCIPLINARYDIVIDESTOSOLVETHEINCREASINGLYCOMPLEXPROBLEMSTHATTHEWORLDFACESWEWILLCONTINUE
TOBEGUIDEDBYTHEIDEATHATWECANACHIEVESOMETHINGMUCHGREATERTOGETHERTHANWECANINDIVIDUALLYAFTERALLTHATWASTHEIDEATHATLEDTOTHECREATIONOFOURUNIVERSITYINTHEFIRSTPLACE
--------------------------------------
```

b) After extracting the msbs, we can try to find the length of cycle to determine the possible key length by following the pseudocode below:

```
for i in msbs:
  if i != cycled_posistion:
        tmp.push(i)
        while tmp not meet the first len(bit) in cycle:
              cycle.append(tmp[0])
              tmp.pop(0)
              cycled_position = len(tmp)-1
```

else:

cycled_position++;

tmp.append(i)

if cycled_position>=len(cycle): // cycle more than once

cycled_position -= len(cycle)

After the execution, we find the cycle length is 255, which means the bits is circulated in the multiple of 255 times. However, we know that 255 = $2^8 - 1$, so it is possibly that it is cycled in 255 bits, and the characteristic polynomial has degree 8. That is, we let $x^8 = ax^7 + bx^6 + \cdots + gx + 1$. And we already have the value of x^8, x^{16}, x^{24} ..., then we expand them into the max degree of 7 and solve the linear equations with 255 equations (actually, to solve deg(7), 7 equations are enough). So clearly it is possible to find out the characteristic polynomial by solving the linear equations.

```
Problem 2-2:
Msbs: ['1', '1', '0', '1', '1', '0', '0', '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1', '0', '1', '0', '0', '1', '1', '0', '0', '1', '1', '0', '1', '0',
 '1', '0', '0', '0', '1', '1', '0', '0', '0', '0', '0', '1', '1', '1', '0', '1', '0', '1', '0', '1', '0', '1', '1', '1', '1', '1', '0', '0', '1', '0', '1', '0',
'0', '0', '0', '1', '0', '0', '1', '1', '1', '1', '1', '1', '1', '1', '0', '0', '0', '0', '1', '0', '1', '1', '1', '1', '0', '0', '0', '1', '1', '0', '1', '0', '
0', '0', '0', '0', '0', '0', '1', '0', '0', '0', '1', '1', '1', '0', '0', '0', '1', '0', '0', '1', '0', '1', '1', '1', '0', '0', '0', '0', '0', '0', '1', '1', '0
', '0', '1', '0', '0', '1', '0', '0', '1', '1', '0', '1', '1', '1', '0', '0', '1', '0', '0', '0', '0', '0', '1', '0', '1', '0', '1', '1', '0', '1', '1', '0', '1'
, '0', '1', '1', '0', '0', '1', '0', '1', '1', '0', '0', '0', '0', '1', '1', '1', '1', '1', '0', '1', '1', '0', '1', '1', '1', '1', '0', '1', '0', '1', '1', '1',
 '0', '1', '0', '0', '0', '1', '0', '0', '0', '0', '1', '1', '0', '1', '1', '0', '0', '0', '1', '1', '1', '1', '0', '0', '1', '1', '1', '0', '0', '1', '1', '0',
'0', '0', '1', '0', '1', '1', '0', '1', '0', '0', '1', '0', '0', '0', '1', '0', '1', '0', '0', '1', '0', '1', '0', '1', '0', '0', '1', '1', '1', '0', '1', '1', '
1', '0', '1', '1', '0', '0', '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1', '0', '1', '0', '0', '1', '1', '0', '0', '1', '1', '0', '1', '0', '1', '0', '0
', '0', '1', '1', '0', '0', '0', '0']
Cycle length: 255
```

Problem 3

a) Follow the pseudocode in the spec, I create two dictionary to save the shuffle cards in each iteration. Below is the result:

```
Naive algorithm:
(1, 2, 3, 4): 38591
(1, 2, 4, 3): 38772
(1, 3, 2, 4): 39558
(1, 3, 4, 2): 54544
(1, 4, 2, 3): 43277
(1, 4, 3, 2): 35272
(2, 1, 3, 4): 38914
(2, 1, 4, 3): 58574
(2, 3, 1, 4): 54758
(2, 3, 4, 1): 54405
(2, 4, 1, 3): 43289
(2, 4, 3, 1): 43229
(3, 1, 2, 4): 43371
(3, 1, 4, 2): 42884
(3, 2, 1, 4): 35229
(3, 2, 4, 1): 42901
(3, 4, 1, 2): 42740
(3, 4, 2, 1): 38978
(4, 1, 2, 3): 31177
(4, 1, 3, 2): 34932
(4, 2, 1, 3): 34928
(4, 2, 3, 1): 31479
(4, 3, 1, 2): 39022
(4, 3, 2, 1): 39152
```

```
Fisher-Yates algorithm:
(1, 2, 3, 4): 41853
(1, 2, 4, 3): 41351
(1, 3, 2, 4): 41486
(1, 3, 4, 2): 41735
(1, 4, 2, 3): 41681
(1, 4, 3, 2): 41737
(2, 1, 3, 4): 41450
(2, 1, 4, 3): 42031
(2, 3, 1, 4): 41727
(2, 3, 4, 1): 41718
(2, 4, 1, 3): 41745
(2, 4, 3, 1): 41548
(3, 1, 2, 4): 41923
(3, 1, 4, 2): 41559
(3, 2, 1, 4): 41782
(3, 2, 4, 1): 41521
(3, 4, 1, 2): 41712
(3, 4, 2, 1): 41729
(4, 1, 2, 3): 41613
(4, 1, 3, 2): 41857
(4, 2, 1, 3): 41585
(4, 2, 3, 1): 41497
(4, 3, 1, 2): 41864
(4, 3, 2, 1): 41272
```

b) I think Fisher-Yates algorithm is better, as each combination has more equal times to appear.

c) Naïve algorithm has worse distribution, so it may not achieve the purpose of generating a uniform random function, which is unfair.

PS: You may simply use python3 <problemxxx.py> to run my codes 😊