IS - Lab Assignment -1
Topic : Implement Symmetric Encryption

Algorithm Details :
Encryption :
Key = (starting_char, Dimensions of matrix, clockwise/counterclockwise, rotating_degree, rotation_direction, swapping_of_rows, shift, shift_direction)

As there are 256 characters in total, we need 256 places to fit them all.
2D matrix of dimensions : (1, 256), (2, 128), (4, 64), (8, 32), (16, 16), (32, 8), (64, 4), (128, 2), (256, 1) will be needed to accommodate all characters.

We are storing ascii values of characters in the matrix spirally inwards (counterclockwise / clockwise) depending on the key.

We can rotate by 90/180 degrees (clockwise / counterclockwise).

If swap = 1, we swap the rows of the matrix.

After the matrix containing ascii values is created, each character(ascii) is mapped to a unique row and column number , which indeed is the code for that character.

Code = row number, column number
Largest row/col number = 256, hence length of each row & column number to be made to 3 (by adding 0s).

Now we encode the string message with numbers using matrix codes.

If the shift value is non-zero, We circularly shift the message counterclockwise (left-shift) /clockwise (right-shift) depending on the key.

Decryption :
Using key the same matrix is formed, and the codes are obtained. Shifting is undone using the shift, shift_direction from the key. Parsing through the encoded message 3 at a time we get row and column number where the character is present in the matrix. In this way we decrypt the message.

Example :
Key = (starting_char = 8, dim = (16, 16), counterclockwise, degree = 90, clockwise, swap=1, shift = 3, shift_direction = clockwise)

Matrix will be formed in following way :

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	8	67	66	65	64	63	62	61	60	59	58	57	56	55	54	53
1	9	68	119	118	117	116	115	114	113	112	111	110	109	108	107	52
2	10	69	120	162	161	161	160	159	158	157	156	155	154	153	106	51
3	11	70	121	163	198	197	196	195	194	193	192	191	190	152	105	50
4	12	71	122	164	199	227	226	224	223	222	221	220	189	151	104	49
5	13	72	123	165	200	228	247	246	245	244	243	219	188	150	103	48
6	14	73	124	166	201	229	248	3	2	1	242	218	187	149	102	47
7	15	74	125	167	202	230	249	4	7	0	241	217	186	148	101	46
8	16	75	126	168	203	231	250	5	6	255	240	216	185	147	100	45
9	17	76	127	169	204	232	251	252	253	254	239	215	184	146	99	44
10	18	77	128	170	205	233	234	235	236	237	238	214	183	145	98	43
11	19	78	129	171	206	207	208	209	210	211	212	213	181	144	97	42
12	20	79	130	172	173	174	175	176	177	178	179	180	181	143	96	41
13	21	80	131	132	133	134	135	136	137	138	139	140	141	142	95	40
14	22	81	82	83	84	85	86	87	88	89	90	91	92	93	94	39
15	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38

After rotating 90 degrees counterclockwise :
(16x16 matrix) :

53…….	……..	…….	……...38
.			.
60…….	……..	……..	……....31
61…….	……..	……..	……....30
.			.
8……..	……..	…….	……23

After swapping rows :

[[8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23]
[67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 24]
[66 119 120 121 122 123 124 125 126 127 128 129 130 131 82 25]
[65 118 163 164 165 166 167 168 169 170 171 172 173 132 83 26]
[64 117 162 199 200 201 202 203 204 205 206 207 174 133 84 27]
[63 116 161 198 227 228 229 230 231 232 233 208 175 134 85 28]
[62 115 160 197 226 247 248 249 250 251 234 209 176 135 86 29]
[61 114 159 196 225 246 3 4 5 252 235 210 177 136 87 30]
[60 113 158 195 224 245 2 7 6 253 236 211 178 137 88 31]
[59 112 157 194 223 244 1 0 255 254 237 212 179 138 89 32]
[58 111 156 193 222 243 242 241 240 239 238 213 180 139 90 33]
[57 110 155 192 221 220 219 218 217 216 215 214 181 140 91 34]
[56 109 154 191 190 189 188 187 186 185 184 183 182 141 92 35]
[55 108 153 152 151 150 149 148 147 146 145 144 143 142 93 36]
[54 107 106 105 104 103 102 101 100 99 98 97 96 95 94 37]
[53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38]]

Message to be encrypted : “sample”

Number	Row	Column
115(s)	006	001
97(a)	014	011
109(m)	012	001
112(p)	009	001
108(l)	013	001
101(e)	014	007

Encrypted message : 006001014011012001009001013001014007

Shift = 3, clockwise(right shift)
Message to be sent to decryptor :
007006001014011012001009001013001014

Decryptor :

1. Create a matrix based on the key.
2. Shift the received message counterclockwise(left shift) by 3.
3. Get the positions from where to refer the character.

Received message : 007006001014011012001009001013001014
After shifting counterclockwise by 3:
006001014011012001009001013001014007
006001014011012001009001013001014007

Row	Column	Number	Character
6	1	115	s
14	11	97	a
12	1	109	m
9	1	112	p
13	1	108	l
14	7	101	e

Decrypted message : “sample"

Analysis :

Key = (starting_char, Dimensions of matrix, clockwise/counterclockwise, rotating_degree, rotation_direction, swapping_of_rows, shift, shift_direction)
Each character is represented by 6 numbers
Brute-force :
To guess the key -

starting_char : 256 alternatives
Dimensions : 9 choices - (1, 256), (2, 128), (4, 64), (8, 32), (16, 16), (32, 8), (64, 4), (128, 2), (256, 1)
clockwise/counterclockwise : 2 choices
rotating degree-direction : 3 choices - (90, clockwise), (90, counterclockwise), (180)
Swapping_of_rows enabled : 2 choices - 0/1
Shift : 6 x len(file) choices - 0, 1, 2, ….6xlen(file)
Shift_direction : 2 choices - clockwise/counterclockwise

To guess length of code -

Possible choices : 6 x len(file) choices

~1990656 x len(file)^2 trails needed to decrypt the message correctly

Instructions to run the encryptor-decryptor :
Decryptor : server, Encryptor : client, sample.txt : testing file(12kb)
python3 decryptor.py
python3 encryptor.py sample.txt
Output of decryptor (decrypted message” is saved in “decrypted.txt”
Check using : diff sample.txt decrypted.txt