Matveeva Lada Ivanna AIS ID 99198

Report lab 4

Exercise 1:
File ex1.py

To use the Polybius square in the program, we created a list called polybius_square.

```
polybius_square = [
    ['A', 'B', 'C', 'D', 'E', 'F'],
    ['G', 'H', 'I', 'J', 'K', 'L'],
    ['M', 'N', 'O', 'P', 'Q', 'R'],
    ['S', 'T', 'U', 'V', 'W', 'X'],
    ['Y', 'Z', '0', '1', '2', '3'],
    ['4', '5', '6', '7', '8', '9']
]
```

After starting the program, the user can enter a message.

```
Please enter message:
```

a) message “ENCRYPT ME 2 DAY”

We enter the message 'ENCRYPT ME 2 DAY' and then we get the encoded message. Since the Polybius square does not contain a space character, each word is written on a new line. The Polybius square contains only uppercase letters, we use the upper() function for the input message.

```
Please enter message: ENCRYPT ME 2 DAY
15 32 13 36 51 34 42
31 15
55
14 11 51
```

a) message “MATVEEVA”

We enter the message 'MATVEEVA' and then we get the encoded message.

```
Please enter message: MATVEEVA
31 11 42 44 15 15 44 11
```

Exercise 2:

File ex2.py

The '^' operation represents the XOR operation.

XOR Truth Table:

A	B	A^B
0	0	0
0	1	1
1	0	1
1	1	0

a^b^c^a^b

a = 1011 b = 0110 c = 0100

a^b = 1101
a^b^c = 1001
a^b^c^a = 0010
a^b^c^a^b = **0100**

a = 0101 b = 1110 c = 1101

a^b = 1011
a^b^c = 0110
a^b^c^a = 0011
a^b^c^a^b = **1101**

a = 0001 b = 0101 c = 1010

a^b = 0100
a^b^c = 1110
a^b^c^a = 1111
a^b^c^a^b = **1010**

In ex2.py, we sequentially enter a, b, and c, and then we get the result.

For a = 1011, b = 0110, c = 0100, we get

```
Please enter a: 1011
Please enter b: 0110
Please enter c: 0100
0100
```

For a = 0101, b = 1110, c = 1101

```
Please enter a: 0101
Please enter b: 1110
Please enter c: 1101
1101
```

For a = 0001 b = 0101 c = 1010

```
Please enter a: 0001
Please enter b: 0101
Please enter c: 1010
1010
```

Exercise 3:

File ex3.py
Entropy is calculated using the formula

H=log2(N)

Where:

- H — entropy of the system in bits
- N — the number of possible states

H = log2(8) = 3 bits
H = log2(128) = 7 bits
H = log2(256) = 8 bits

ex3.py

We declared variables with state values where we calculate the entropy. To perform the calculation, we use the math library.

```
log2_8 = int(math.log2(8))
log2_128 = int(math.log2(128))
log2_256 = int(math.log2(256))
```

And we get the result:

```
For 8 states entropy equals 3
For 128 states entropy equals 7
For 256 states entropy equals 8
```