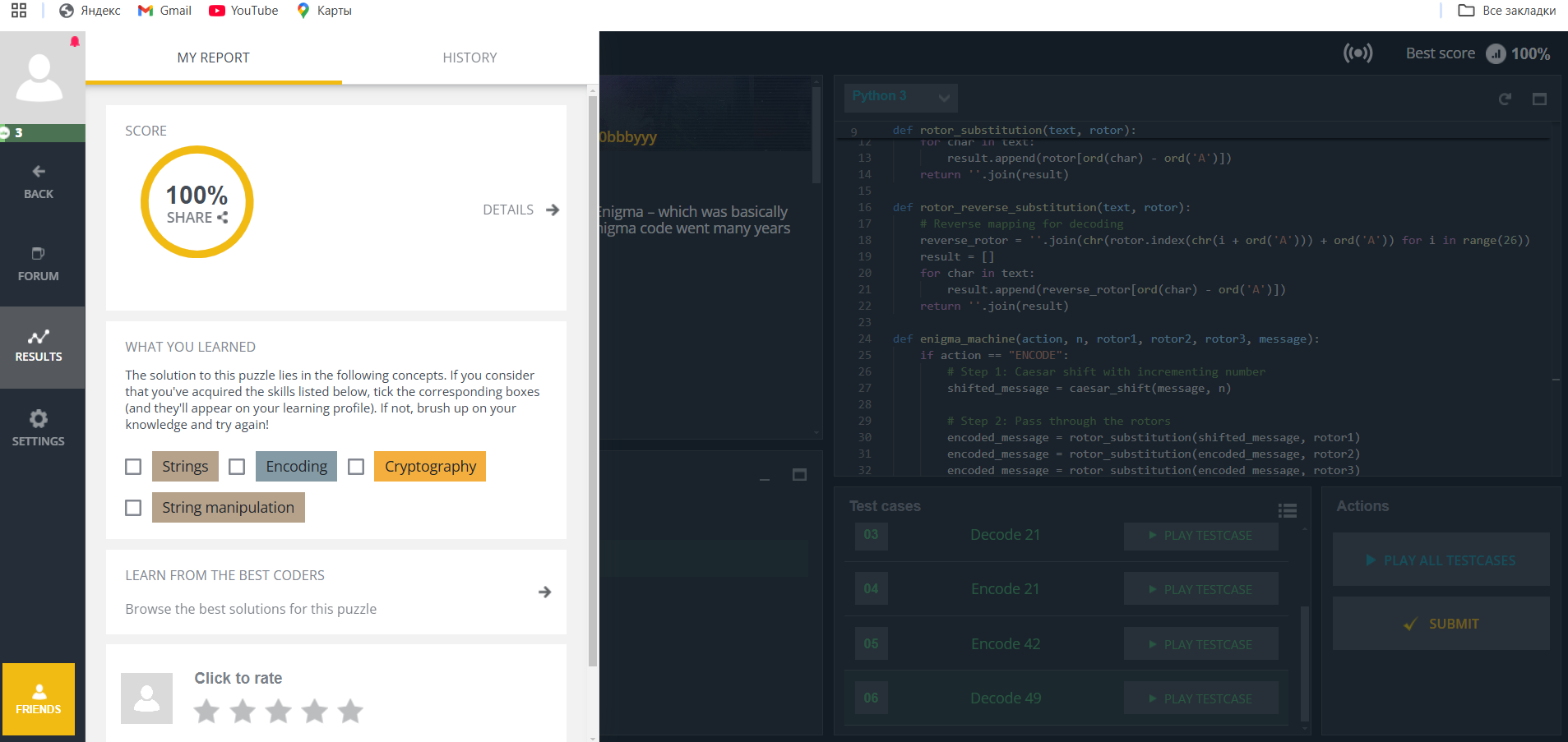
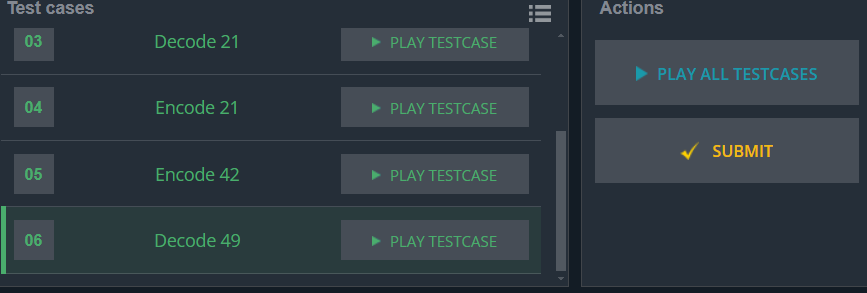
**Єлін Євген РІ-41 мн**

# Лабораторна робота №1 Enigma Machine

Посилання на гітхаб:

<https://github.com/YelinYevgen/lab-1.git>





**Код програми на Python:**

def caesar\_shift(text, shift):

# Incremental Caesar shift

result = []

for i, char in enumerate(text):

new\_char = chr((ord(char) - ord('A') + shift + i) % 26 + ord('A'))

result.append(new\_char)

return ''.join(result)

def rotor\_substitution(text, rotor):

# Substitutes each character in text with its rotor mapping

result = []

for char in text:

result.append(rotor[ord(char) - ord('A')])

return ''.join(result)

def rotor\_reverse\_substitution(text, rotor):

# Reverse mapping for decoding

reverse\_rotor = ''.join(chr(rotor.index(chr(i + ord('A'))) + ord('A')) for i in range(26))

result = []

for char in text:

result.append(reverse\_rotor[ord(char) - ord('A')])

return ''.join(result)

def enigma\_machine(action, n, rotor1, rotor2, rotor3, message):

if action == "ENCODE":

# Step 1: Caesar shift with incrementing number

shifted\_message = caesar\_shift(message, n)

# Step 2: Pass through the rotors

encoded\_message = rotor\_substitution(shifted\_message, rotor1)

encoded\_message = rotor\_substitution(encoded\_message, rotor2)

encoded\_message = rotor\_substitution(encoded\_message, rotor3)

return encoded\_message

elif action == "DECODE":

# Reverse through the rotors for decoding

decoded\_message = rotor\_reverse\_substitution(message, rotor3)

decoded\_message = rotor\_reverse\_substitution(decoded\_message, rotor2)

decoded\_message = rotor\_reverse\_substitution(decoded\_message, rotor1)

# Reverse Caesar shift

reversed\_caesar\_message = []

for i, char in enumerate(decoded\_message):

shift\_value = n + i

new\_char = chr((ord(char) - ord('A') - shift\_value + 26) % 26 + ord('A'))

reversed\_caesar\_message.append(new\_char)

return ''.join(reversed\_caesar\_message)

# Reading input

action = input().strip() # Either "ENCODE" or "DECODE"

n = int(input().strip()) # Starting shift N

rotor1 = input().strip() # First rotor substitution sequence

rotor2 = input().strip() # Second rotor substitution sequence

rotor3 = input().strip() # Third rotor substitution sequence

message = input().strip() # Message to encode or decode

# Running the Enigma machine

result = enigma\_machine(action, n, rotor1, rotor2, rotor3, message)

print(result) # Output the result