The Caesar Code

An Online **quiz game**

Developed by **Adam Melhem**

Country: Israel

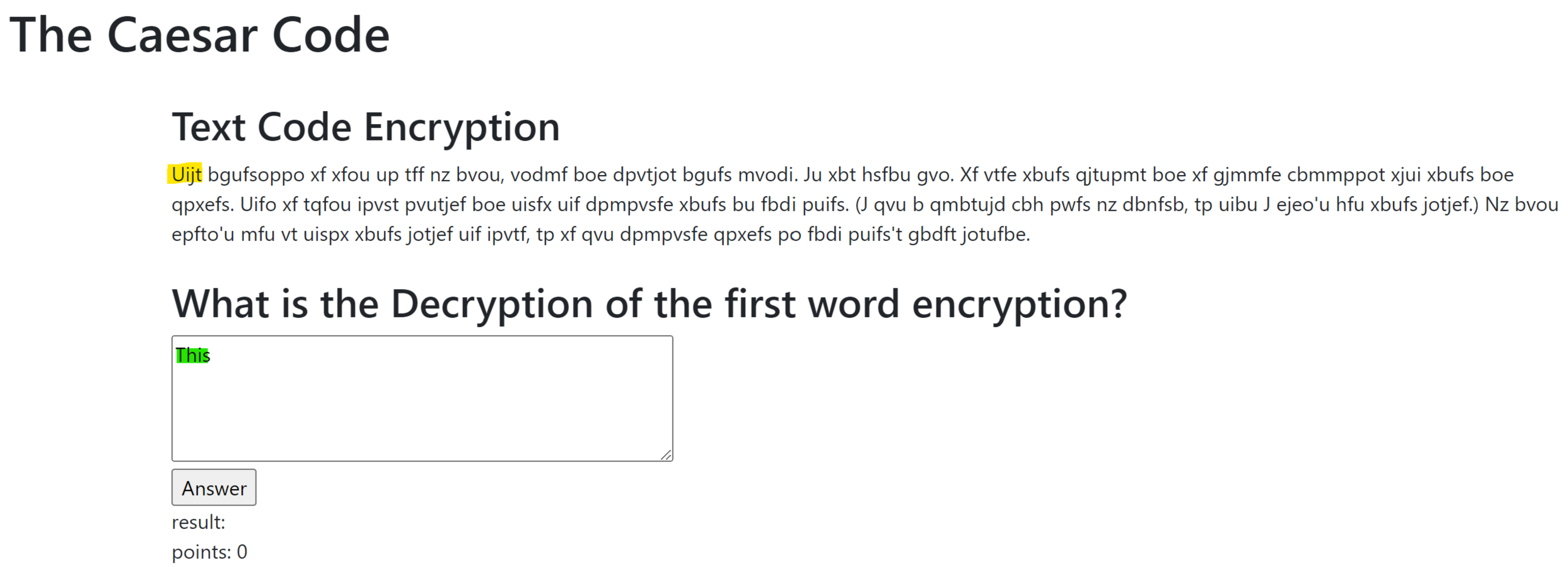
Phone: +972548100372

Email: fm_post@hotmail.com

LinkedIn: https://www.linkedin.com/in/adam-melhem

The game display Encrypted text to the player, and the player is expected to Decrypt the first word of the text and submit it as an answer.

Each question has different score value points, the score value points depends on the question readability, the more simple the question is the less points the player gain for answered the question.



The Caesar Code

Text Code Encryption

Uijt bgufsoppo xf xfou up tff nz bvou, vodmf boe dpvtjot bgufs mvodi. Ju xbt hsfbu gvo. Xf vtfe xbufs qjtupmt boe xf gjmmfe cbmmppot xjui xbufs boe qpxefs. Uifo xf tqfou ipvst pvutjef boe uisfx uif dpmpvsfe xbufs bu fbdi puifs. (J qvu b qmbtujd cbh pwfs nz dbnfsb, tp uibu J ejeo'u hfu xbufs jotjef.) Nz bvou epfto'u mfu vt uispx xbufs jotjef uif ipvtf, tp xf qvu dpmpvsfe qpxefs po fbdi puifs't gbdft jotufbe.

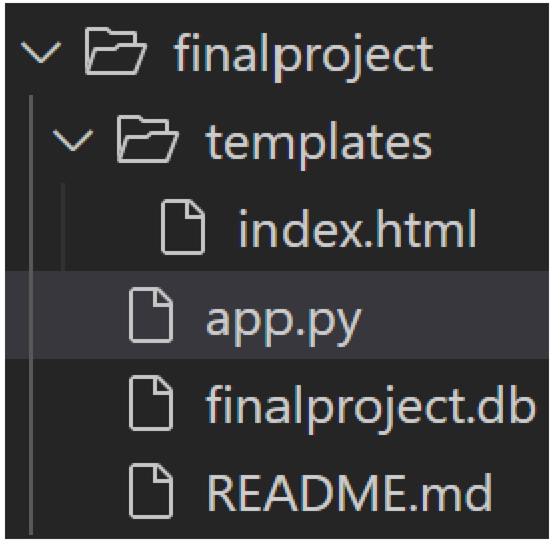
What is the Decryption of the first word encryption?

This

result:

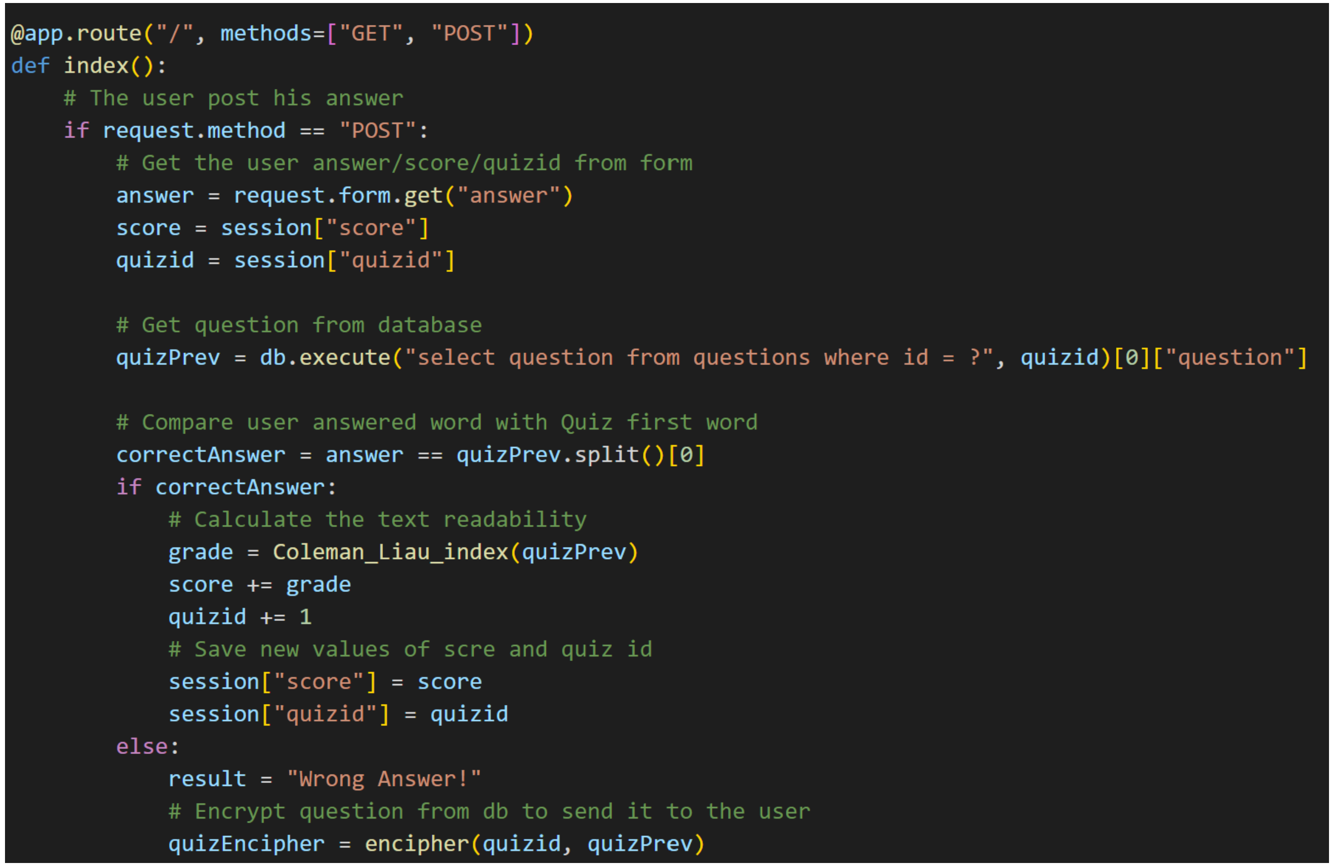
points: 0

For this question above for example, the word Uijt (Highlighted yellow), is the question word the user expected to decrypt , and the expected answer is the word "This"(Highlighted green).



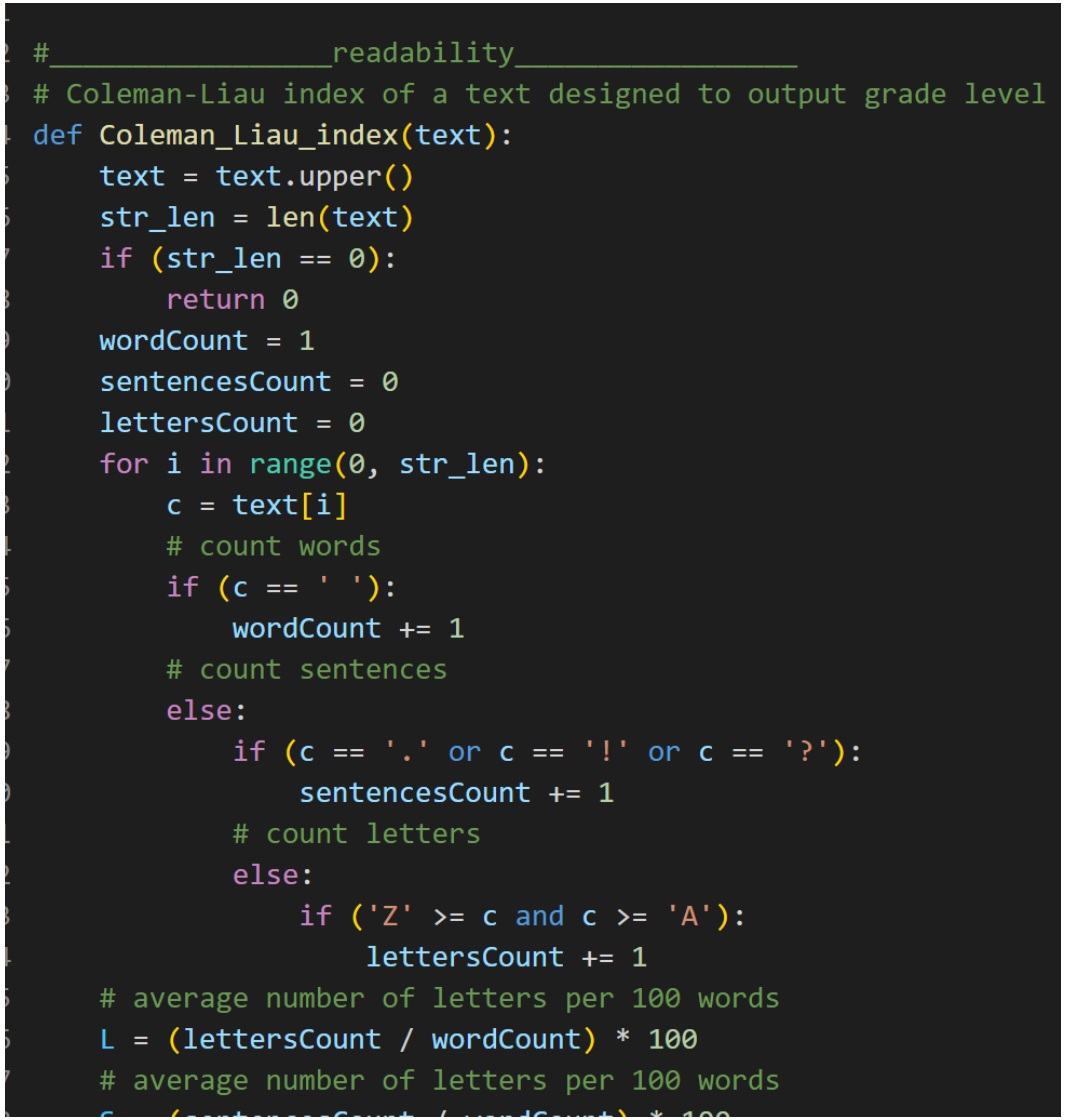
```
finalproject
  templates
    index.html
  app.py
  finalproject.db
  README.md
```

The index.html file , is the front-end view page of the project, while the app.py file, is the project back-end logic , and the finalproject.db file is the data base of the project.



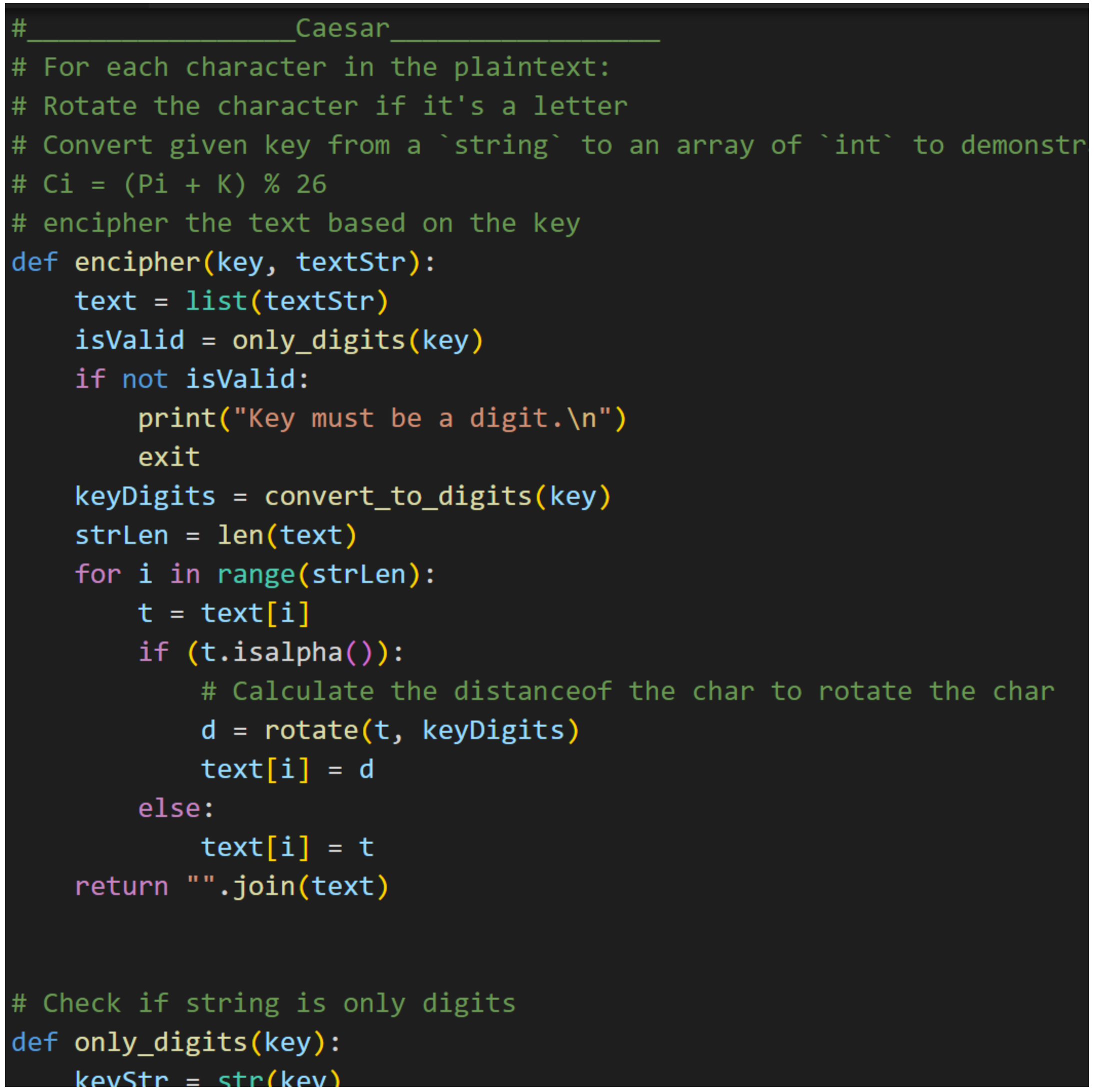
```python
@app.route("/", methods=["GET", "POST"])
def index():
    # The user post his answer
    if request.method == "POST":
        # Get the user answer/score/quizid from form
        answer = request.form.get("answer")
        score = session["score"]
        quizid = session["quizid"]

        # Get question from database
        quizPrev = db.execute("select question from questions where id = ?", quizid)[0]["question"]

        # Compare user answered word with Quiz first word
        correctAnswer = answer == quizPrev.split()[0]
        if correctAnswer:
            # Calculate the text readability
            grade = Coleman_Liau_index(quizPrev)
            score += grade
            quizid += 1
            # Save new values of scre and quiz id
            session["score"] = score
            session["quizid"] = quizid
        else:
            result = "Wrong Answer!"
            # Encrypt question from db to send it to the user
            quizEncipher = encipher(quizid, quizPrev)
```

The index web api get both post and get requests, when the user first load the page he triggers the get method , while when he post an answer he triggers the post method, to provide the answer and get the result.



```python
#_________________readability_________________
# Coleman-Liau index of a text designed to output grade level
def Coleman_Liau_index(text):
    text = text.upper()
    str_len = len(text)
    if (str_len == 0):
        return 0
    wordCount = 1
    sentencesCount = 0
    lettersCount = 0
    for i in range(0, str_len):
        c = text[i]
        # count words
        if (c == ' '):
            wordCount += 1
        # count sentences
        else:
            if (c == '.' or c == '!' or c == '?'):
                sentencesCount += 1
            # count letters
            else:
                if ('Z' >= c and c >= 'A'):
                    lettersCount += 1
    # average number of letters per 100 words
    L = (lettersCount / wordCount) * 100
    # average number of letters per 100 words
    S = (sentencesCount / wordCount) * 100
```

This method calculates the readability of the given text and return grade score level.



```
#_________________Caesar_________________
# For each character in the plaintext:
# Rotate the character if it's a letter
# Convert given key from a `string` to an array of `int` to demonstr
# Ci = (Pi + K) % 26
# encipher the text based on the key
def encipher(key, textStr):
    text = list(textStr)
    isValid = only_digits(key)
    if not isValid:
        print("Key must be a digit.\n")
        exit
    keyDigits = convert_to_digits(key)
    strLen = len(text)
    for i in range(strLen):
        t = text[i]
        if (t.isalpha()):
            # Calculate the distanceof the char to rotate the char
            d = rotate(t, keyDigits)
            text[i] = d
        else:
            text[i] = t
    return "".join(text)


# Check if string is only digits
def only_digits(key):
    keyStr = str(key)
```

The encipher method converts the text to Caesar code text, that the player expected to figure out.

# The Caesar Code

The application has collection of questions, each question is encrypted into Caesar code , and the user must guess

what is the first word in the text , the system will check the user answer and calculate the readability of the

question if the answer is correct then the user will add this grade to the score.

The index.html has the form that display the questions and score and answer result , he also contain a form to get the user answer.

The app.py file has the backend code that contains 4 main parts , post function , get function , readability calculating function , and decryption function .

the get function returns the first question and store new score value and qustion id in the session , while the encryption function encrypt the text of the question to send it to the user ,, the decryption function do exactly the opposite , the readability function calculate how easy or hard the text was and provides a level as a grade.

if the user answer correct then the system add up the score and ask the user the next question.