4CS001 –

Introductory Programming

&

Problem-Solving

Coursework Report

Name: Abhises Yogal

Group: L4SG1

University Student ID: 2514056

College ID: NP03CY4A240093

Course: BSc (Hons) Cybersecurity

Caesar Cipher

The Caesar Cipher, named after **Julius Caesar** who reportedly used it for military communications, is one of the simplest and oldest methods of encryption. It's a type of substitution cipher in which each letter in the plaintext is shifted a certain number of places up or down the alphabet, allowing for both encryption and decryption of the text.

The Caesar Cipher is easy to understand and implement, it's not very secure by modern standards. Since there are only 25 possible shifts (ignoring the trivial shift of 0), it's vulnerable to brute-force attacks, where an attacker tries all possible shifts to decode the message. Nonetheless, it's a great way to learn the basics of encryption and provides a historical glimpse into the world of cryptography.

1. What are the most challenging aspects of the coursework task?
The Caesar Cipher is a classic puzzle that may seem dishonestly simple. However, coding it can present some fun and complicated challenges. One of these challenges involves handling non-alphabetic characters. You need to decide whether to ignore spaces, punctuation, and numbers, keep them in their original positions, or remove them altogether. Another challenge is the cover-around effect. When a letter at the end of the alphabet (such as "Z") is shifted, it must cover around to the beginning (to "A"). This is where modular arithmetic comes into play, and it can become confusing.

Case sensitivity also adds to the difficulty. Managing both uppercase and lowercase letters while ensuring that the cipher behaves consistently for each can be tricky. Additionally, error handling is essential. You must ensure that users cannot break your program by entering invalid shifts or messages, which requires thorough input validation.

While the Caesar Cipher itself is not computationally intensive, efficiently handling large files can be challenging, especially when reading from and writing to them. Finally, creating clear prompts and error messages is an unexpectedly significant part of the coding process. It's important to design these elements carefully so that users understand how to use the program smoothly. These challenges made working on the Caesar Cipher more intricate and rewarding other than that, the most difficult part was changing user-input to red color.

2. How did you go about completing the task?

To complete the task, I followed a structured approach that began with understanding the requirements and goals of the Caesar Cipher, particularly focusing on its encryption, decryption, and the role of the shift number.

Next, I wrote the code, starting with the encryption and decryption functions. I then integrated user input functions, ensuring robust error handling throughout. Thorough testing and debugging followed to ensure that each function, as well as the overall workflow, operated correctly.

I also aimed to enhance the user experience by filtering the prompts and error messages. Additionally, I merged features such as reading from either a file or the console and writing the results to a file. Finally, I reviewed and improved the code for readability, efficiency, and maintainability, resulting in a functional and user-friendly Caesar Cipher program.

3. What have you learned over the course of completing this coursework task?
Completing the coursework task involving the Caesar Cipher has provided me with many valuable lessons. Firstly, I have gained a deeper understanding of how to handle non-alphabetic characters and the importance of determining how to treat spaces, punctuation, and numbers during the encryption and decryption process.

Additionally, learning about the wrap-around effect when shifting letters has given me practical insight into modular arithmetic, which is crucial for ensuring that characters like "Z" correctly loop back to "A." Managing both uppercase and lowercase letters has emphasized the importance of case sensitivity and the need for consistent behavior across different cases.

Error handling has also been a significant area of learning. I have realized the importance of thoroughly validating user inputs to prevent invalid shifts or messages from breaking the program. This task has highlighted the necessity for efficient file handling, especially when working with large files, to ensure smooth reading and writing processes.

Lastly, I have learned that creating clear and user-friendly prompts and error messages is vital for a positive user experience. It is essential to design prompts that allow users to easily understand and interact with the program. Overall, this coursework task has deepened my appreciation for the intricacies of coding, error handling, and user experience design.

Programs Preview:

*Caesar_Cipher_AbhisesYogal_2514056.py - C:\Users\abish\OneDrive\Desktop\Herald\Assignment\Python\Coursework-1\Caesar_Cipher_AbhisesYogal_251405

File Edit Format Run Options Window Help

```
""" importing string """
import string

def welcome():
    """ printing welcome message """
    print("Welcome to the Caesar Cipher.")
    print("This program encrypts and decrypts text with the Caesar Cipher.")

def enter_message():
    """ Asks for mode and message to encrypt or decrypt """
    while True:
        mode=input("Would you like to encrypt(e) or decrypt(d)?:")
        if mode in ["e","d"]:
            break
        print("Invalid mode.")
        if mode=="e":
            operation = "encrypt"
        else:
            operation = "decrypt"
    message=input(f"What message would you like to {operation}?:")
    while True:
        try:
            shift=int(input("What is the shift number?:"))
            break
        except ValueError:
            print("Invalid shift")
        message=message.upper()
        return mode,message,shift
```

```
def encrypt(message,shift):
    """ code to encrypt message through shift """
    outcome=""
    for char in message:
        if char in string.ascii_uppercase:
            index=(string.ascii_uppercase.index(char)+shift)%26
            outcome+=string.ascii_uppercase[index]
        else:
            outcome+=char
    return outcome

def decrypt(message,shift):
    """ decrypts message through reversing shift """
    return encrypt(message,-shift)

def process_file(filename, mode, shift):
    """ opens file and encrypts or decrypts each line """
    messages=[]
    with open(filename, encoding="utf-8") as f:
        for line in f:
            message=line.strip().upper()
            if mode=="e":
                messages.append(encrypt(message, shift))
            else:
                messages.append(decrypt(message, shift))
    return messages
```

```
def is_file(filename):
    """ checks if the file exists or not """
    try:
        with open(filename, encoding="utf-8"):
            return True
    except FileNotFoundError:
        return False

def write_messages(messages):
    """ writes messages to file 'results.txt' """
    with open("results.txt", "w", encoding="utf-8") as f:
        for message in messages:
            f.write(message+"\n")
```

```
def message_or_file():
    """ asks user to select the mode """
    while True:
        mode=input("Would you like to encrypt(e) or decrypt(d)?:")
        if mode in ["e","d"]:
            break
        print("Invalid mode")
    while True:
        source=input("Would you like to read from a file(f) or the console(c)?:")
        if source in ["f","c"]:
            break
        print("Invalid source")
    if source=="c":
        message=input("What message would you like to {}?:".format("encrypt" if mode=="e" else "decrypt"))
        message=message.upper()
        return mode, message, None
    else:
        while True:
            filename=input("Enter a filename:")
            if is_file(filename):
                break
            print("Invalid filename")
        return mode, None, filename
```

```
def main():
    """ Main function to welcome the user, handle encryption or decryption of messages or
    files, and loop until the user decides to exit. """
    welcome()
    while True:
        mode, message, filename=message_or_file()
        if filename:
            while True:
                try:
                    shift=int(input("What is the shift number?:"))
                    break
                except ValueError:
                    print("Invalid shift")
            messages=process_file(filename, mode, shift)
            write_messages(messages)
            print("Output written to results.txt")
        else:
            shift=int(input("What is the shift number?:"))
            outcome=encrypt(message,shift) if mode=="e" else decrypt(message,shift)
            print(outcome)
        while True:
            answer=input("Would you like to encrypt or decrypt another message?(y/n):")
            if answer in ["y","n"]:
                break
            print("Invalid input")
        if answer=="n":
            print("Thanks for using the program, goodbye!")
            break

if __name__=="__main__":
    main()
```

Program Outcome:

From console (only encryption):

```
Welcome to the Caesar Cipher.
This program encrypts and decrypts text with the Caesar Cipher.
Would you like to encrypt(e) or decrypt(d)?:e
Would you like to read from a file(f) or the console(c)?:c
What message would you like to encrypt?:Abhises Yogal
What is the shift number?:7
HIOPZLZ FVNHS
Would you like to encrypt or decrypt another message?(y/n):n
Thanks for using the program, goodbye!
```

From console (only decryption):

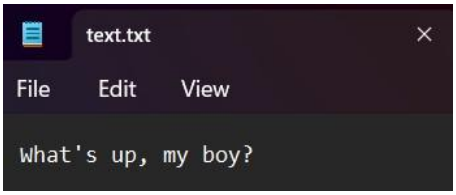
```
= RESTART: C:\Users\abish\OneDrive\Desktop\Herald\Assignment\Pytl
r_AbhisesYogal_2514056.py
Welcome to the Caesar Cipher.
This program encrypts and decrypts text with the Caesar Cipher.
Would you like to encrypt(e) or decrypt(d)?:d
Would you like to read from a file(f) or the console(c)?:c
What message would you like to decrypt?:hiopzlz fvnhs
What is the shift number?:7
ABHISES YOGAL
Would you like to encrypt or decrypt another message?(y/n):n
Thanks for using the program, goodbye!
```

From console (both encryption and decryption):

```
= RESTART: C:\Users\abish\OneDrive\Desktop\Herald\Assignment\Pytl
Welcome to the Caesar Cipher.
This program encrypts and decrypts text with the Caesar Cipher.
Would you like to encrypt(e) or decrypt(d)?:e
Would you like to read from a file(f) or the console(c)?:c
What message would you like to encrypt?:herald college
What is the shift number?:5
MJWFQI HTQQJLJ
Would you like to encrypt or decrypt another message?(y/n):y
Would you like to encrypt(e) or decrypt(d)?:d
Would you like to read from a file(f) or the console(c)?:c
What message would you like to decrypt?:mjwfqi htqqjlj
What is the shift number?:5
HERALD COLLEGE
Would you like to encrypt or decrypt another message?(y/n):n
Thanks for using the program, goodbye!
```

From file (only encryption):

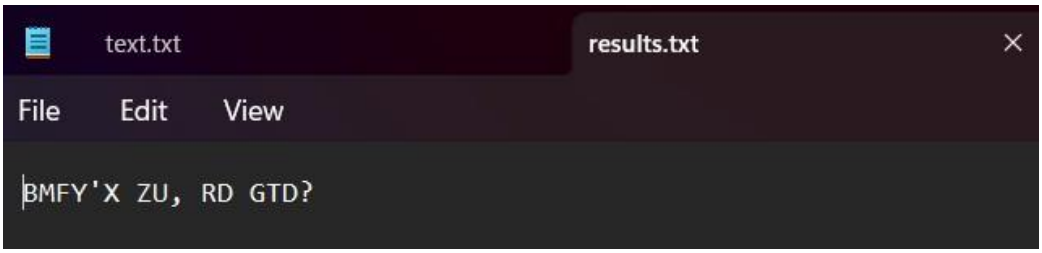
Text file before encryption:



Program:

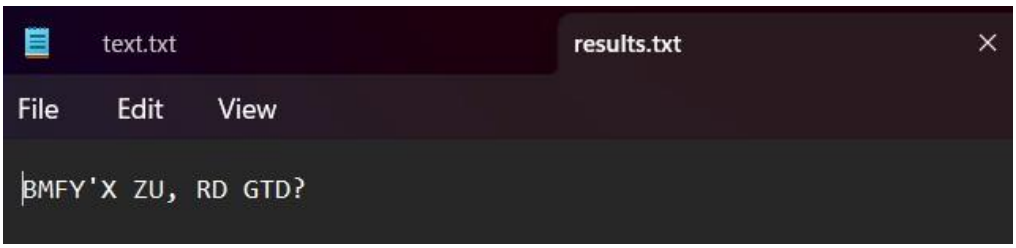
```
============ RESTART: C:\Users\abish\OneDrive\Desktop\Herald\As
Welcome to the Caesar Cipher.
This program encrypts and decrypts text with the Caesar Cipher.
Would you like to encrypt(e) or decrypt(d)?:e
Would you like to read from a file(f) or the console(c)?:f
Enter a filename:text.txt
What is the shift number?:5
Output written to results.txt
Would you like to encrypt or decrypt another message?(y/n):n
Thanks for using the program, goodbye!
```

Text file after encryption:



From file (only decryption):

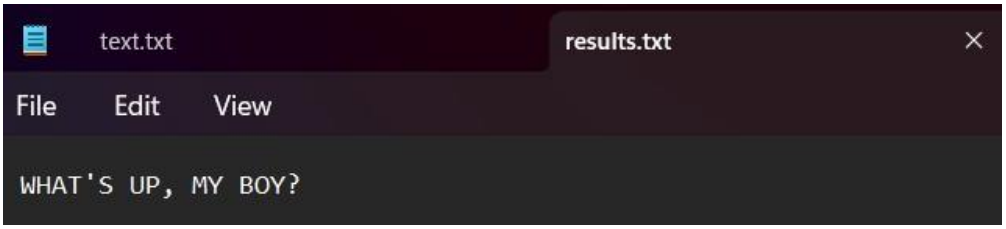
Text file before decryption:



Program:

```
============= RESTART: C:\Users\abish\OneDrive\Desktop\Herald\As
Welcome to the Caesar Cipher.
This program encrypts and decrypts text with the Caesar Cipher.
Would you like to encrypt(e) or decrypt(d)?:d
Would you like to read from a file(f) or the console(c)?:f
Enter a filename:results.txt
What is the shift number?:5
Output written to results.txt
Would you like to encrypt or decrypt another message?(y/n):n
Thanks for using the program, goodbye!
```

Text file after decryption:



From file (both encryption and decryption):

```
============= RESTART: C:\Users\abish\OneDrive\Desktop\Herald\As
Welcome to the Caesar Cipher.
This program encrypts and decrypts text with the Caesar Cipher.
Would you like to encrypt(e) or decrypt(d)?:e
Would you like to read from a file(f) or the console(c)?:f
Enter a filename:text.txt
What is the shift number?:5
Output written to results.txt
Would you like to encrypt or decrypt another message?(y/n):y
Would you like to encrypt(e) or decrypt(d)?:d
Would you like to read from a file(f) or the console(c)?:f
Enter a filename:results.txt
What is the shift number?:5
Output written to results.txt
Would you like to encrypt or decrypt another message?(y/n):n
Thanks for using the program, goodbye!
```