# 08 - Brute Force Program for Alphabet Cipher Hacking Documentation

Author: Muzaffar Ali

Version: 1.0

Copyright: (c) Muzaffar Ali

License: Public

## Purpose

This project is designed for the Advanced Python Class by Muzaffar Ali. The Brute Force Program for Alphabet Cipher Hacking attempts to decode a message encoded with a Caesar cipher by trying all possible shift values from 0 to 99.

## Requirements

- No additional libraries are required as the program uses Python's built-in functionality for string manipulation.

## Environment Setup

No special environment setup is needed for this program as it only uses Python's standard library.

## Code

The main code for the Brute Force Program for Alphabet Cipher Hacking is as follows:  
```python  
# Create an alphabet list containing 'a' to 'z', repeated four times  
alphabet = [chr(i) for i in range(97, 123)] \* 4  
  
def caesar(start\_text, shift\_amount, cipher\_direction):  
 """  
 This function encodes or decodes the given text using the Caesar cipher.  
   
 Parameters:  
 start\_text (str): The text to encode or decode.  
 shift\_amount (int): The number of positions to shift each character.  
 cipher\_direction (str): Either 'encode' to encrypt or 'decode' to decrypt.  
   
 Returns:  
 str: The encoded or decoded text.  
 """  
 end\_text = ""  
 if cipher\_direction == "decode":  
 # Reverse the shift amount for decoding  
 shift\_amount \*= -1  
 for char in start\_text:  
 if char in alphabet:  
 # Find the current position of the character  
 position = alphabet.index(char)  
 # Calculate the new position after shifting  
 new\_position = position + shift\_amount  
 # Append the new character to the result text  
 end\_text += alphabet[new\_position]  
 else:  
 # If the character is not in the alphabet, leave it unchanged  
 end\_text += char  
 return end\_text  
  
def brute\_force\_caesar(cipher\_text):  
 """  
 This function attempts to decode the given cipher text using all possible shift amounts (0-99).  
   
 Parameters:  
 cipher\_text (str): The text to decode.  
 """  
 print("Brute Force Results:")  
 for shift in range(100):  
 # Decode the text with the current shift amount  
 decoded\_text = caesar(start\_text=cipher\_text, shift\_amount=shift, cipher\_direction="decode")  
 # Print the result for the current shift amount  
 print(f"Shift amount {shift}: {decoded\_text}")  
  
# Get the cipher text from the user  
cipher\_text = input("Enter the cipher text to brute force:\n").lower()  
# Perform brute force decoding on the input cipher text  
brute\_force\_caesar(cipher\_text)  
```

## Execution

1. Ensure Python is installed on your system.  
2. Save the provided code in a file named `brute\_force\_cipher.py`.  
3. Open a terminal or command prompt.  
4. Navigate to the directory where `brute\_force\_cipher.py` is saved.  
5. Run the program using the command:  
```  
python brute\_force\_cipher.py  
```

## How It Works

1. Alphabet List:  
 - A repeated list of alphabet letters to handle character shifting and wrapping.  
  
2. Function `caesar(start\_text, shift\_amount, cipher\_direction)`:  
 - Parameters: Takes the text to encode or decode, the shift amount, and the direction (encode or decode).  
 - Direction Check: Reverses the shift amount for decoding.  
 - Character Shifting: Shifts each character by the specified amount, leaving non-alphabet characters unchanged.  
 - Return: Returns the encoded or decoded text.  
  
3. Function `brute\_force\_caesar(cipher\_text)`:  
 - Parameters: Takes the cipher text to decode.  
 - Brute Force: Attempts to decode the text using all possible shift amounts (0-99).  
 - Output: Prints the decoded text for each shift amount.  
  
4. User Interaction:  
 - Prompts the user to enter the cipher text.  
 - Performs brute force decoding on the input text.

## Output

The program attempts to decode the provided cipher text using all possible shift values from 0 to 99, displaying the results for each shift amount.