Indian Institute of Information Technology Surat

Lab Report on

Information Security (CS 602)

Practical Submitted by

Aditya Kumar(UI22CS03)

Course Faculty
Dr. Reema Patel

Department of Computer Science and Engineering
Indian Institute of Information Technology Surat
Gujarat-394190, India

January-2025

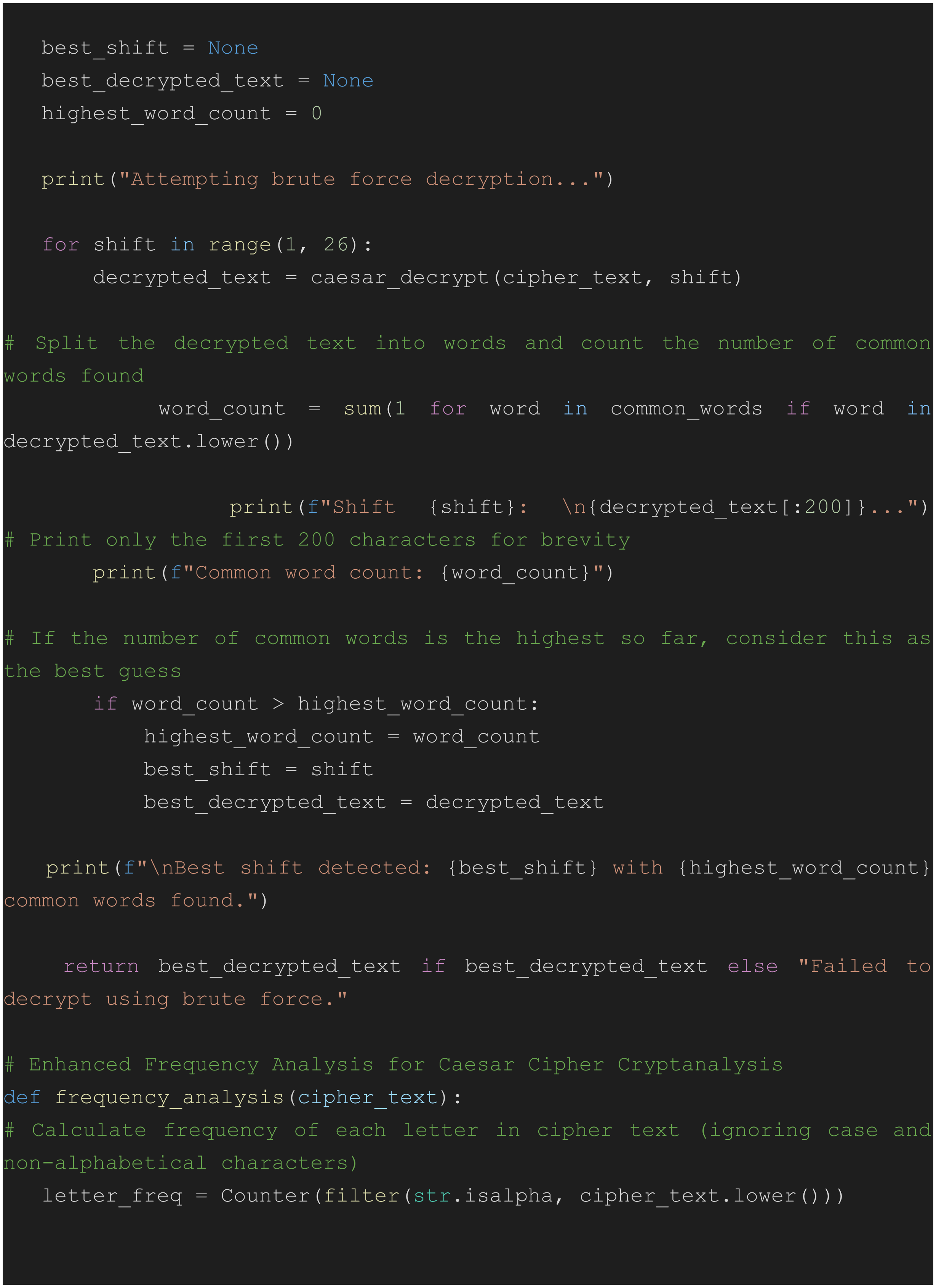
Table of Contents

Caesar Cipher Encryption and Decryption in Python

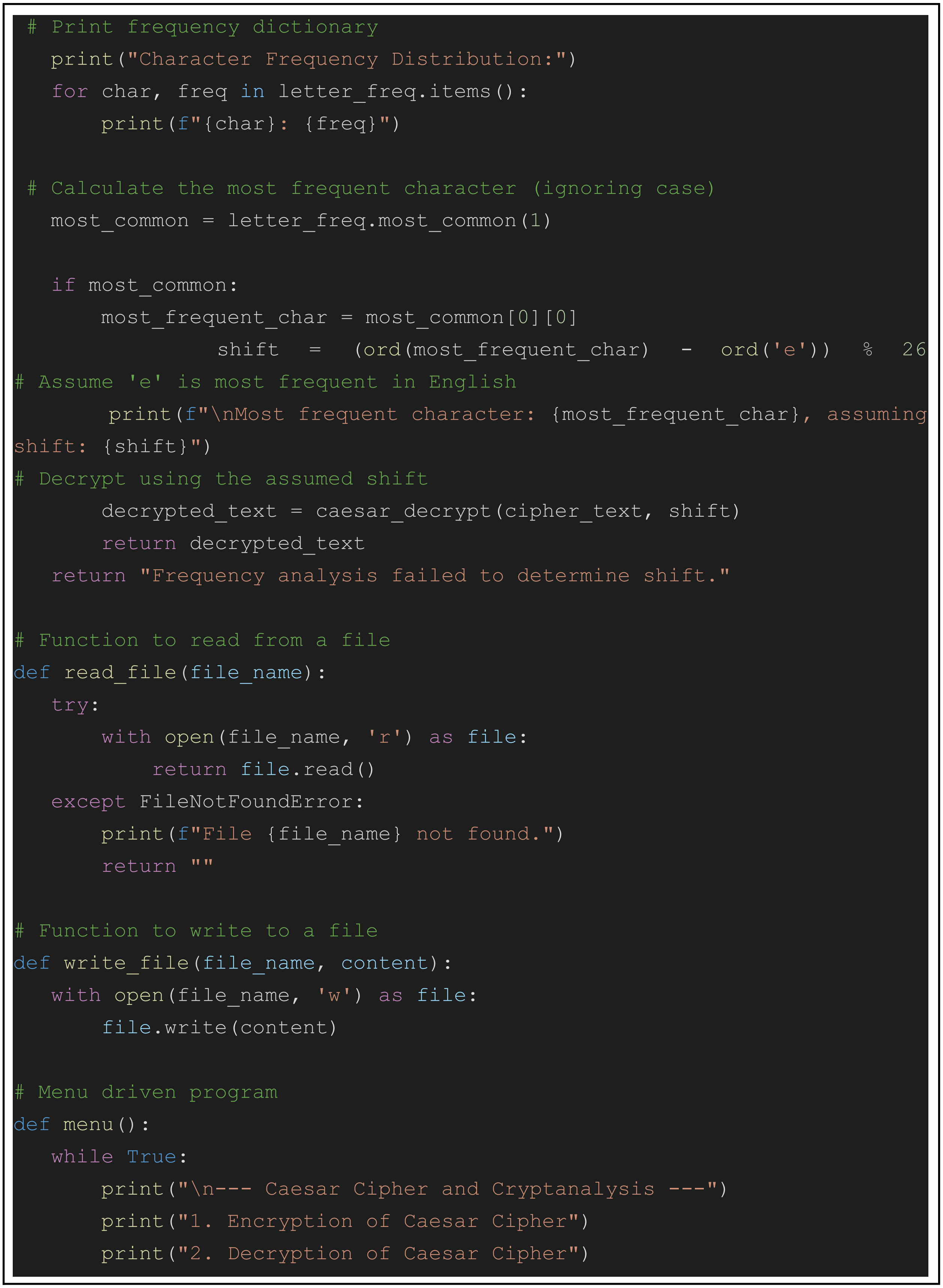
1. Write the Menu driven Program for following Cipher and Cryptanalysis.
    a. 1. Encryption and Decryption of Ceaser Cipher
    b. 2. Cryptanalysis of Ceaser Cipher – Brute Force attack, Frequency Analysis.
        i. Input: File - Large Plaintext (file.txt)
        ii. Output: File - Encoded Text (Cipher.txt)
        iii. Input File Name: Plaintext.txt
        iv. Encrypted File Name: Cipher.txt
        v. Decrypted Filename: Recover.txt

Code:

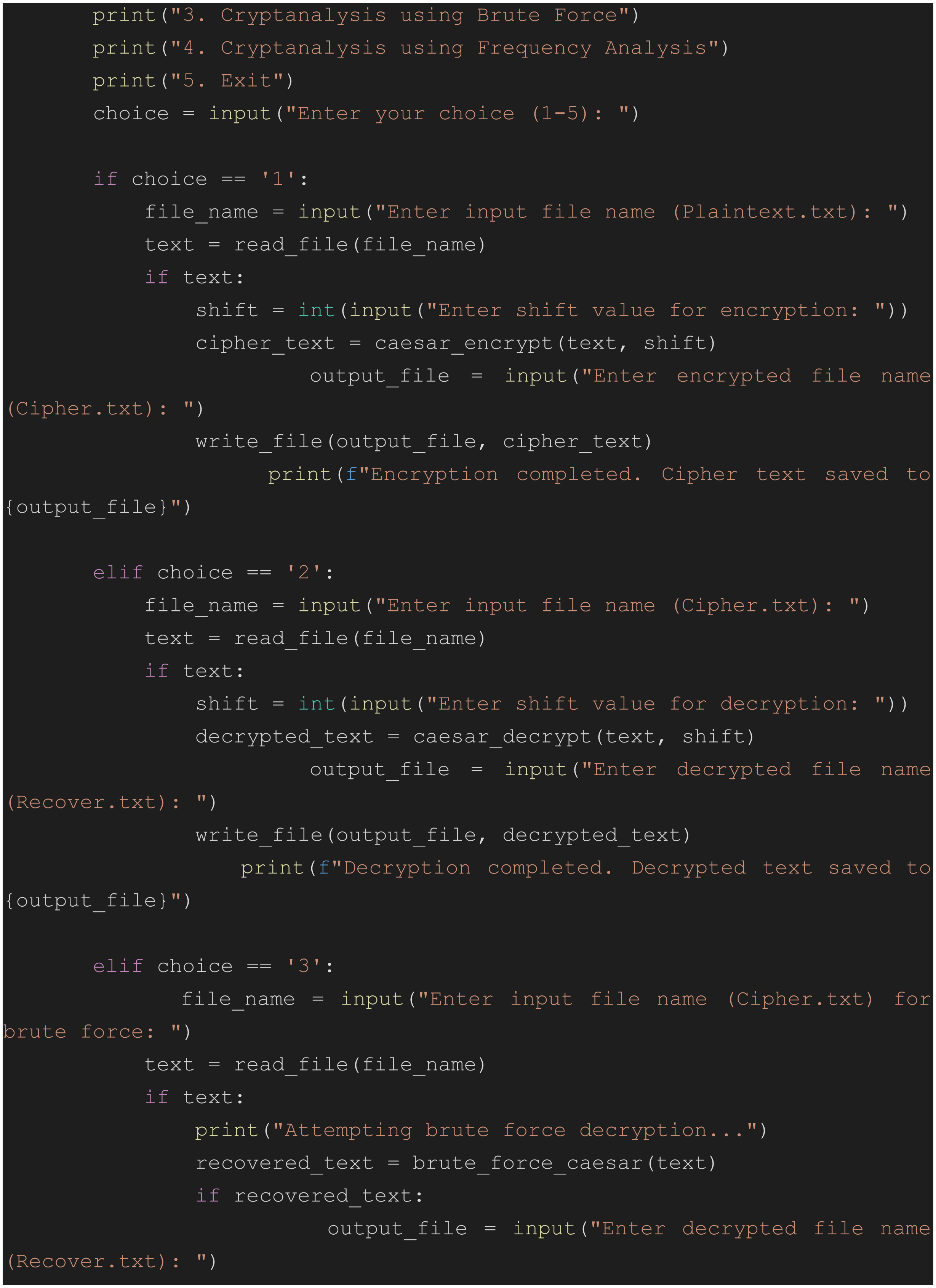
```
import string
from collections import Counter

# Function for Caesar Cipher Encryption
def caesar_encrypt(text, shift):
   result = ""
   for char in text:
       if char.isalpha():
           shift_base = 65 if char.isupper() else 97
           result += chr((ord(char) - shift_base + shift) % 26 + shift_base)
       else:
           result += char
   return result

# Function for Caesar Cipher Decryption
def caesar_decrypt(text, shift):
   return caesar_encrypt(text, -shift)

# Brute Force Cryptanalysis with automatic analysis
def brute_force_caesar(cipher_text):
    common_words = ['the', 'and', 'of', 'to', 'is', 'in', 'it', 'you', 'that', 'for']  # Common words in English
```



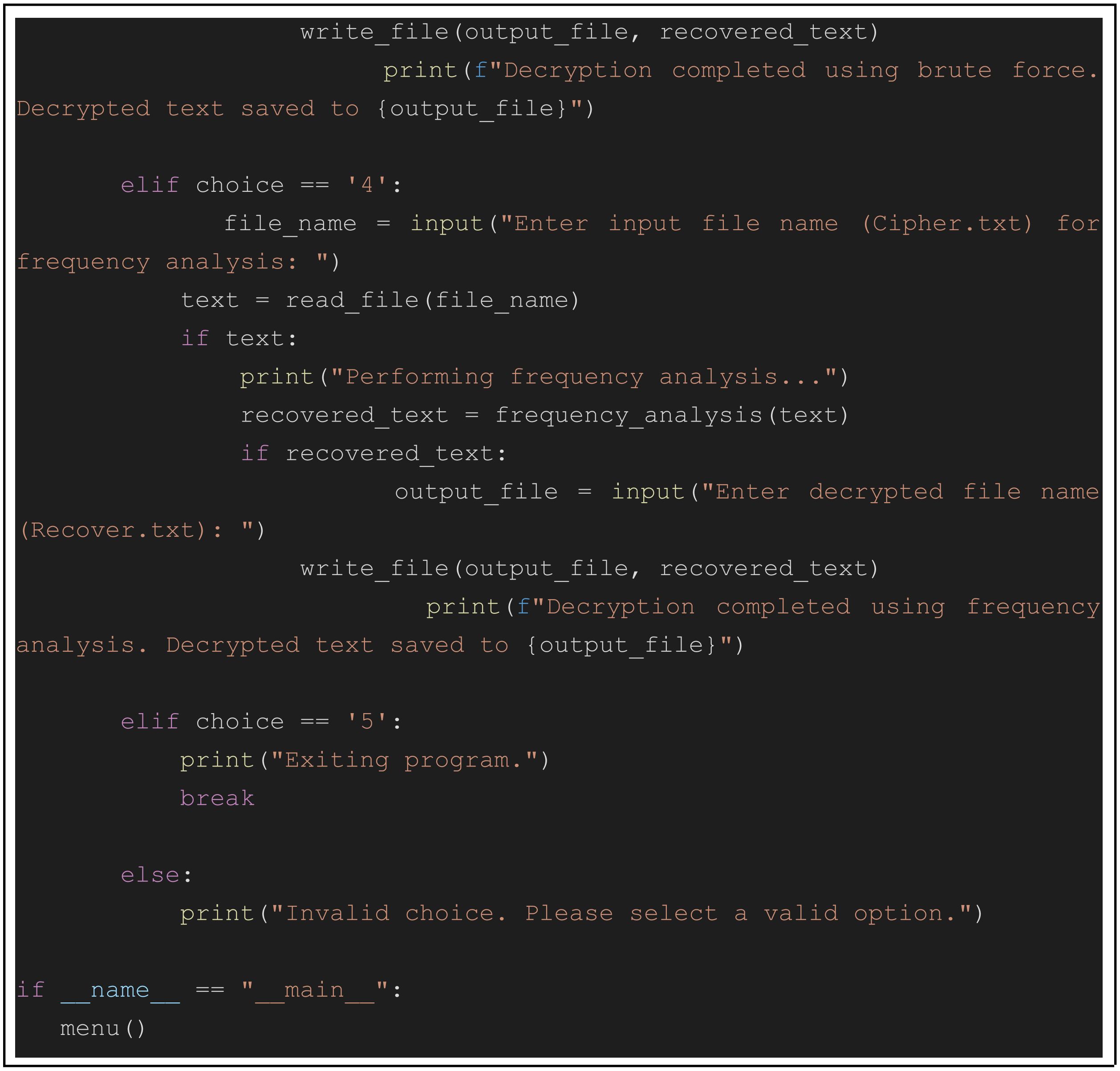
```
   best_shift = None
   best_decrypted_text = None
   highest_word_count = 0

   print("Attempting brute force decryption...")

   for shift in range(1, 26):
       decrypted_text = caesar_decrypt(cipher_text, shift)

#  Split  the  decrypted  text  into  words  and  count  the  number  of  common
words found
            word_count  =  sum(1  for  word  in  common_words  if  word  in
decrypted_text.lower())

                  print(f"Shift  {shift}:  \n{decrypted_text[:200]}...")
# Print only the first 200 characters for brevity
       print(f"Common word count: {word_count}")

# If the number of common words is the highest so far, consider this as
the best guess
       if word_count > highest_word_count:
           highest_word_count = word_count
           best_shift = shift
           best_decrypted_text = decrypted_text

   print(f"\nBest shift detected: {best_shift} with {highest_word_count}
common words found.")

     return  best_decrypted_text  if  best_decrypted_text  else  "Failed  to
decrypt using brute force."

# Enhanced Frequency Analysis for Caesar Cipher Cryptanalysis
def frequency_analysis(cipher_text):
#  Calculate  frequency  of  each  letter  in  cipher  text  (ignoring  case  and
non-alphabetical characters)
   letter_freq = Counter(filter(str.isalpha, cipher_text.lower()))
```



```
 # Print frequency dictionary
   print("Character Frequency Distribution:")
   for char, freq in letter_freq.items():
       print(f"{char}: {freq}")

 # Calculate the most frequent character (ignoring case)
   most_common = letter_freq.most_common(1)

   if most_common:
       most_frequent_char = most_common[0][0]
                shift  =  (ord(most_frequent_char)  -  ord('e'))  %  26
# Assume 'e' is most frequent in English
        print(f"\nMost frequent character: {most_frequent_char}, assuming
shift: {shift}")
# Decrypt using the assumed shift
       decrypted_text = caesar_decrypt(cipher_text, shift)
       return decrypted_text
   return "Frequency analysis failed to determine shift."

# Function to read from a file
def read_file(file_name):
   try:
       with open(file_name, 'r') as file:
           return file.read()
   except FileNotFoundError:
       print(f"File {file_name} not found.")
       return ""

# Function to write to a file
def write_file(file_name, content):
   with open(file_name, 'w') as file:
       file.write(content)

# Menu driven program
def menu():
   while True:
       print("\n--- Caesar Cipher and Cryptanalysis ---")
       print("1. Encryption of Caesar Cipher")
       print("2. Decryption of Caesar Cipher")
```

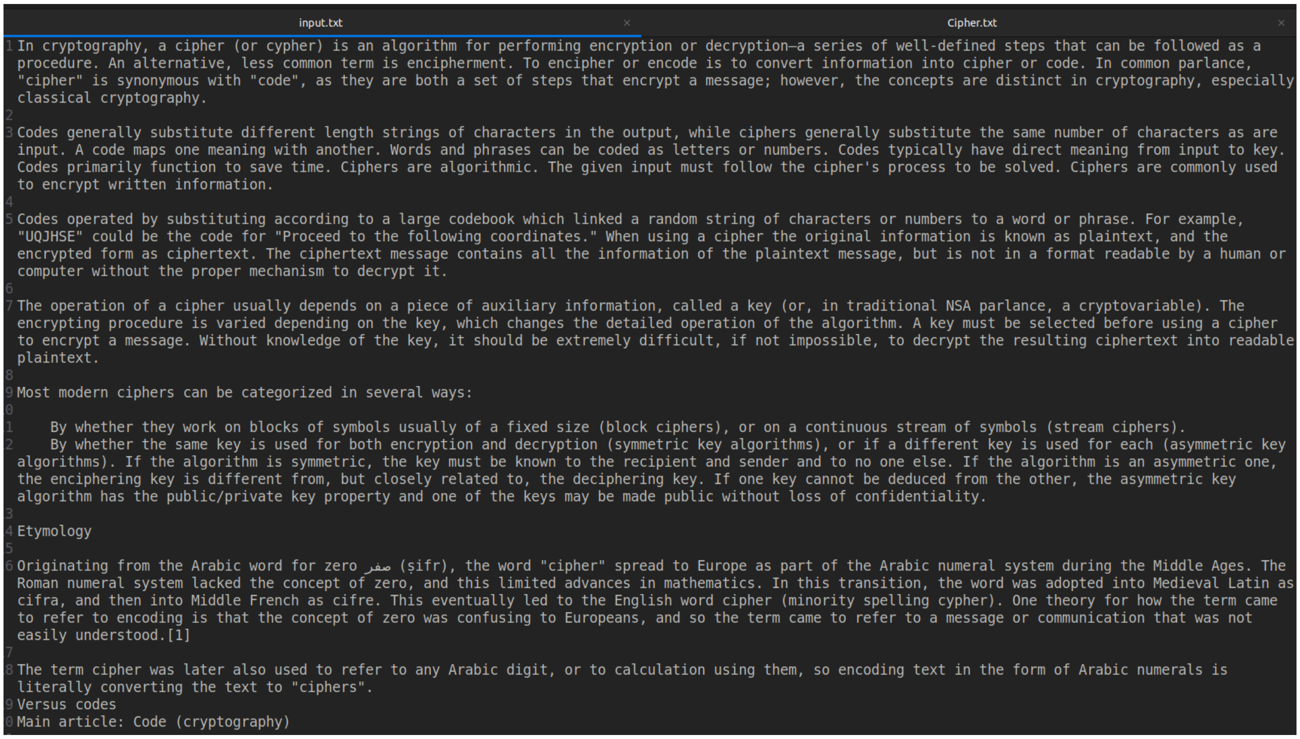


```
        print("3. Cryptanalysis using Brute Force")
        print("4. Cryptanalysis using Frequency Analysis")
        print("5. Exit")
        choice = input("Enter your choice (1-5): ")

        if choice == '1':
            file_name = input("Enter input file name (Plaintext.txt): ")
            text = read_file(file_name)
            if text:
                shift = int(input("Enter shift value for encryption: "))
                cipher_text = caesar_encrypt(text, shift)
                        output_file  =  input("Enter  encrypted  file  name
(Cipher.txt): ")
                write_file(output_file, cipher_text)
                     print(f"Encryption  completed.  Cipher  text  saved  to
{output_file}")

        elif choice == '2':
            file_name = input("Enter input file name (Cipher.txt): ")
            text = read_file(file_name)
            if text:
                shift = int(input("Enter shift value for decryption: "))
                decrypted_text = caesar_decrypt(text, shift)
                        output_file  =  input("Enter  decrypted  file  name
(Recover.txt): ")
                write_file(output_file, decrypted_text)
                   print(f"Decryption  completed.  Decrypted  text  saved  to
{output_file}")

        elif choice == '3':
               file_name  =  input("Enter  input  file  name  (Cipher.txt)  for
brute force: ")
            text = read_file(file_name)
            if text:
                print("Attempting brute force decryption...")
                recovered_text = brute_force_caesar(text)
                if recovered_text:
                         output_file  =  input("Enter  decrypted  file  name
(Recover.txt): ")
```



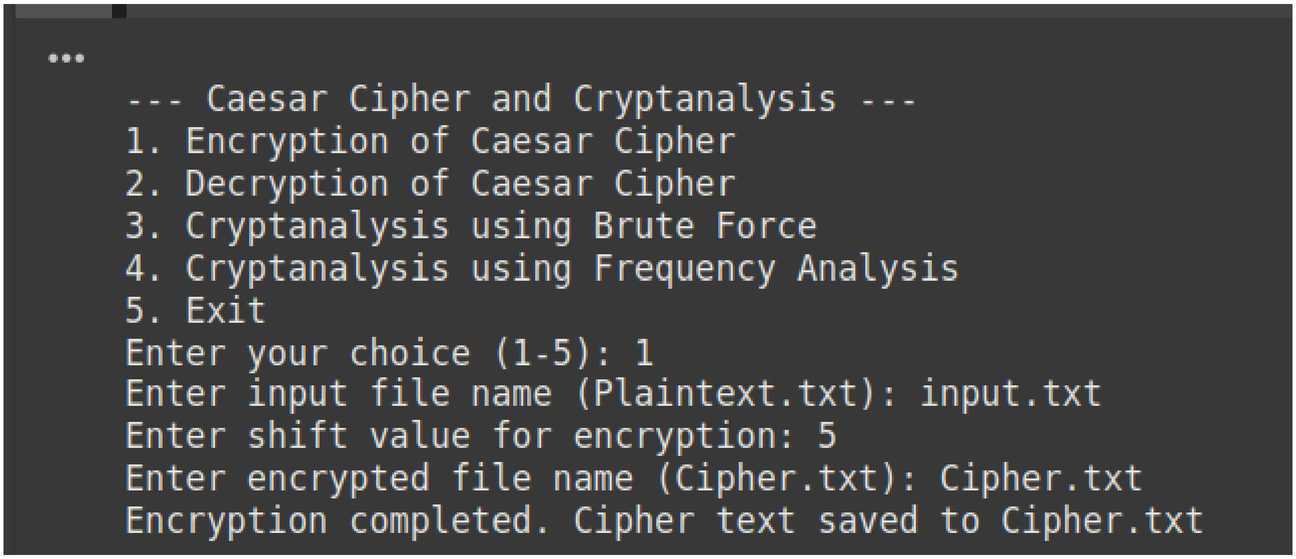
```python
                    write_file(output_file, recovered_text)
                    print(f"Decryption completed using brute force. Decrypted text saved to {output_file}")

        elif choice == '4':
            file_name = input("Enter input file name (Cipher.txt) for frequency analysis: ")
            text = read_file(file_name)
            if text:
                print("Performing frequency analysis...")
                recovered_text = frequency_analysis(text)
                if recovered_text:
                    output_file = input("Enter decrypted file name (Recover.txt): ")
                    write_file(output_file, recovered_text)
                    print(f"Decryption completed using frequency analysis. Decrypted text saved to {output_file}")

        elif choice == '5':
            print("Exiting program.")
            break

        else:
            print("Invalid choice. Please select a valid option.")

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

Output:



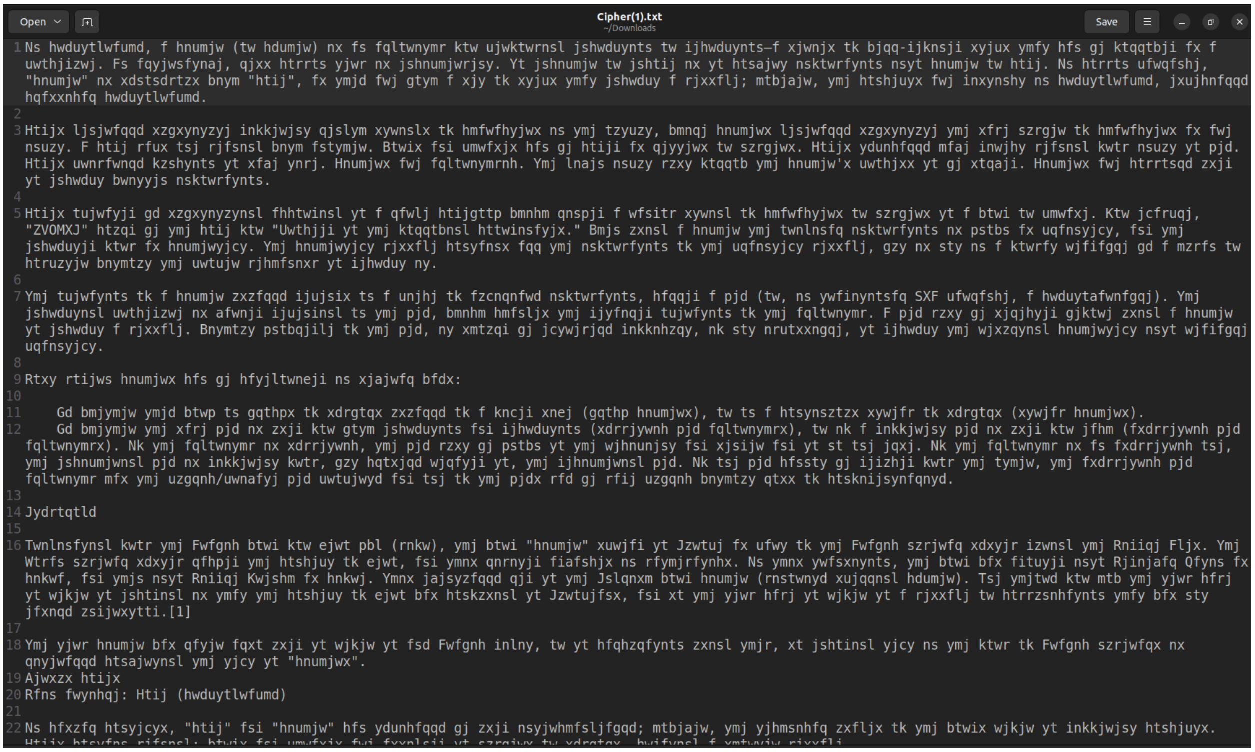
```
In cryptography, a cipher (or cypher) is an algorithm for performing encryption or decryption—a series of well-defined steps that can be followed as a
procedure. An alternative, less common term is encipherment. To encipher or encode is to convert information into cipher or code. In common parlance,
"cipher" is synonymous with "code", as they are both a set of steps that encrypt a message; however, the concepts are distinct in cryptography, especially
classical cryptography.

Codes generally substitute different length strings of characters in the output, while ciphers generally substitute the same number of characters as are
input. A code maps one meaning with another. Words and phrases can be coded as letters or numbers. Codes typically have direct meaning from input to key.
Codes primarily function to save time. Ciphers are algorithmic. The given input must follow the cipher's process to be solved. Ciphers are commonly used
to encrypt written information.

Codes operated by substituting according to a large codebook which linked a random string of characters or numbers to a word or phrase. For example,
"UQJHSE" could be the code for "Proceed to the following coordinates." When using a cipher the original information is known as plaintext, and the
encrypted form as ciphertext. The ciphertext message contains all the information of the plaintext message, but is not in a format readable by a human or
computer without the proper mechanism to decrypt it.

The operation of a cipher usually depends on a piece of auxiliary information, called a key (or, in traditional NSA parlance, a cryptovariable). The
encrypting procedure is varied depending on the key, which changes the detailed operation of the algorithm. A key must be selected before using a cipher
to encrypt a message. Without knowledge of the key, it should be extremely difficult, if not impossible, to decrypt the resulting ciphertext into readable
plaintext.

Most modern ciphers can be categorized in several ways:

    By whether they work on blocks of symbols usually of a fixed size (block ciphers), or on a continuous stream of symbols (stream ciphers).
    By whether the same key is used for both encryption and decryption (symmetric key algorithms), or if a different key is used for each (asymmetric key
algorithms). If the algorithm is symmetric, the key must be known to the recipient and sender and to no one else. If the algorithm is an asymmetric one,
the enciphering key is different from, but closely related to, the deciphering key. If one key cannot be deduced from the other, the asymmetric key
algorithm has the public/private key property and one of the keys may be made public without loss of confidentiality.

Etymology

Originating from the Arabic word for zero صفر (ṣifr), the word "cipher" spread to Europe as part of the Arabic numeral system during the Middle Ages. The
Roman numeral system lacked the concept of zero, and this limited advances in mathematics. In this transition, the word was adopted into Medieval Latin as
cifra, and then into Middle French as cifre. This eventually led to the English word cipher (minority spelling cypher). One theory for how the term came
to refer to encoding is that the concept of zero was confusing to Europeans, and so the term came to refer to a message or communication that was not
easily understood.[1]

The term cipher was later also used to refer to any Arabic digit, or to calculation using them, so encoding text in the form of Arabic numerals is
literally converting the text to "ciphers".
Versus codes
Main article: Code (cryptography)
```

input.txt



```
--- Caesar Cipher and Cryptanalysis ---
1. Encryption of Caesar Cipher
2. Decryption of Caesar Cipher
3. Cryptanalysis using Brute Force
4. Cryptanalysis using Frequency Analysis
5. Exit
Enter your choice (1-5): 1
Enter input file name (Plaintext.txt): input.txt
Enter shift value for encryption: 5
Enter encrypted file name (Cipher.txt): Cipher.txt
Encryption completed. Cipher text saved to Cipher.txt
```

Option 1: Encryption, with key=5



Ns hwduytlwfumd, f hnumjw (tw hdumjw) nx fs fqltwnymr ktw ujwktwrnsl jshwduynts tw ijhwduynts—f xjwnjx tk bjqq-ijknsji xyjux ymfy hfs gj ktqqtbji fx f uwthjizwj. Fs fqyjwsfynaj, qjxx htrrts yjwr nx jshnumjwrjsy. Yt jshnumjw tw jshtij nx yt htsajwy nsktwrfynts nsyt hnumjw tw htij. Ns htrrts ufwqfshj, "hnumjw" nx xdstsdrtzx bnym "htij", fx ymjd fwj gtym f xjy tk xyjux ymfy jshwduy f rjxxflj; mtbjajw, ymj htshjuyx fwj inxynshy ns hwduytlwfumd, jxujhnfqqd hqfxxnhfq hwduytlwfumd.

Htijx ljsjwfqqd xzgxynyzyj inkkjwjsy qjslym xywnslx tk hmfwfhyjwx ns ymj tzyuzy, bmnqj hnumjwx ljsjwfqqd xzgxynyzyj ymj xfrj szrgjw tk hmfwfhyjwx fx fwj nsuzy. F htij rfux tsj rjfsnsl bnym fstymjw. Btwix fsi umwfxjx hfs gj htiji fx qjyyjwx tw szrgjwx. Htijx ydunhfqqd mfaj inwjhy rjfsnsl kwtr nsuzy yt pjd. Htijx uwnrfwnqd kzshynts yt xfaj ynrj. Hnumjwx fwj fqltwnymrnh. Ymj lnajs nsuzy rzxy ktqqtb ymj hnumjw'x uwthjxx yt gj xtqaji. Hnumjwx fwj htrrtsqd zxji yt jshwduy bwnyyjs nsktwrfynts.

Htijx tujwfyji gd xzgxynyzynsl fhhtwinsl yt f qfwlj htijgttp bmnhm qnspji f wfsitr xywnsl tk hmfwfhyjwx tw szrgjwx yt f btwi tw umwfxj. Ktw jcfruqj, "ZVOMXJ" htzqi gj ymj htij ktw "Uwthjji yt ymj ktqqtbnsl httwinsfyjx." Bmjs zxnsl f hnumjw ymj twnlnsfq nsktwrfynts nx pstbs fx uqfnsyjcy, fsi ymj jshwduyji ktwr fx hnumjwyjcy. Ymj hnumjwyjcy rjxxflj htsyfnsx fqq ymj nsktwrfynts tk ymj uqfnsyjcy rjxxflj, gzy nx sty ns f ktwrfy wjfifgqj gd f mzrfs tw htruzyjw bnymtzy ymj uwtujw rjhmfsnxr yt ijhwduy ny.

Ymj tujwfynts tk f hnumjw zxzfqqd ijujsix ts f unjhj tk fzcnqnfwd nsktwrfynts, hfqqji f pjd (tw, ns ywfinyntsfq SXF ufwqfshj, f hwduytafwnfgqj). Ymj jshwduynsl uwthjizwj nx afwnji ijujsinsl ts ymj pjd, bmnhm hmfsljx ymj ijyfnqji tujwfynts tk ymj fqltwnymr. F pjd rzxy gj xjqjhyji gjktwj zxnsl f hnumjw yt jshwduy f rjxxflj. Bnymtzy pstbqjilj tk ymj pjd, ny xmtzqi gj jcywjrjqd inkknhzqy, nk sty nrutxxngqj, yt ijhwduy ymj wjxzqynsl hnumjwyjcy nsyt wjfifgqj uqfnsyjcy.

Rtxy rtijws hnumjwx hfs gj hfyjltwneji ns xjajwfq bfdx:

    Gd bmjymjw ymjd btwp ts gqthpx tk xdrgtqx zxzfqqd tk f kncji xnej (gqthp hnumjwx), tw ts f htsynsztzx xywjfr tk xdrgtqx (xywjfr hnumjwx).
    Gd bmjymjw ymj xfrj pjd nx zxji ktw gtym jshwduynts fsi ijhwduynts (xdrrjywnh pjd fqltwnymrx), tw nk f inkkjwjsy pjd nx zxji ktw jfhm (fxdrrjywnh pjd fqltwnymrx). Nk ymj fqltwnymr nx xdrrjywnh, ymj pjd rzxy gj pstbs yt ymj wjhnunjsy fsi xjsijw fsi yt st tsj jqxj. Nk ymj fqltwnymr nx fs fxdrrjywnh tsj, ymj jshnumjwnsl pjd nx inkkjwjsy kwtr, gzy hqtxjqd wjqfyji yt, ymj ijhnumjwnsl pjd. Nk tsj pjd hfssty gj ijizhji kwtr ymj tymjw, ymj fxdrrjywnh pjd fqltwnymr mfx ymj uzgqnh/uwnafyj pjd uwtujwyd fsi tsj tk ymj pjdx rfd gj rfij uzgqnh bnymtzy qtxx tk htsknijsynfqnyd.

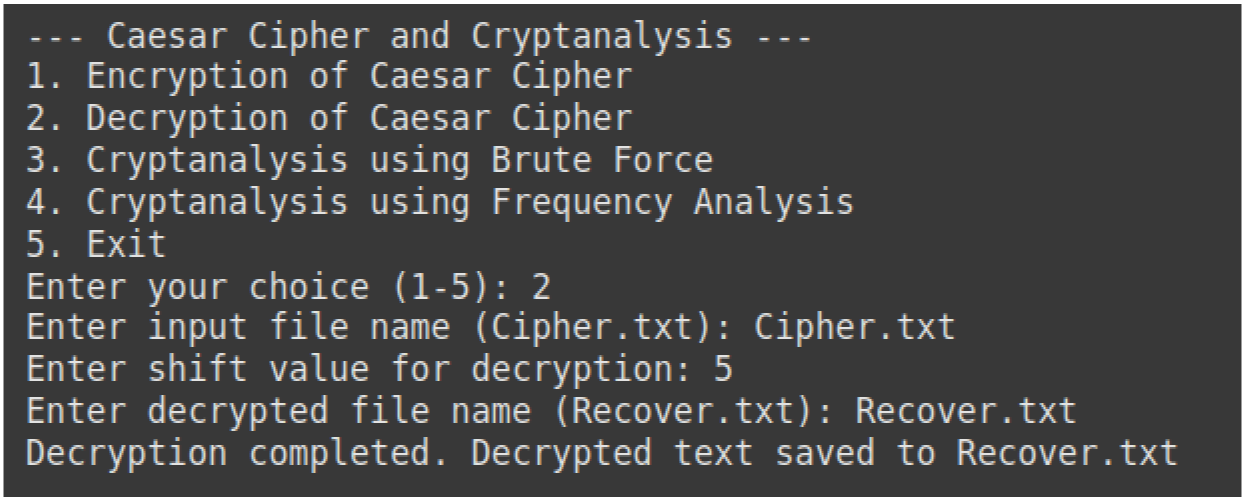
Jydrtqtld

Twnlnsfynsl kwtr ymj Fwfgnh btwi ktw ejwt pbl (rnkw), ymj btwi "hnumjw" xuwjfi yt Jzwtuj fx ufwy tk ymj Fwfgnh szrjwfq xdxyjr izwnsl ymj Rniiqj Fljx. Ymj Wtrfs szrjwfq xdxyjr qfhpji ymj htshjuy tk ejwt, fsi ymnx qnrnyji fiafshjx ns rfymjrfynhx. Ns ymnx ywfsxnynts, ymj btwi bfx fituyji nsyt Rjinjafq Qfyns fx hnkwf, fsi ymjs nsyt Rniiqj Kwjshm fx hnkwj. Ymnx jajsyzfqqd qji yt ymj Jslqnxm btwi hnumjw (rnstwnyd xujqqnsl hdumjw). Tsj ymjtwd ktw mtb ymj yjwr hfrj yt wjkjw yt jshtinsl nx ymfy ymj htshjuy tk ejwt bfx htskzxnsl yt Jzwtujfsx, fsi xt ymj yjwr hfrj yt wjkjw yt f rjxxflj tw htrrzsnhfynts ymfy bfx sty jfxnqd zsijwxytti.[1]

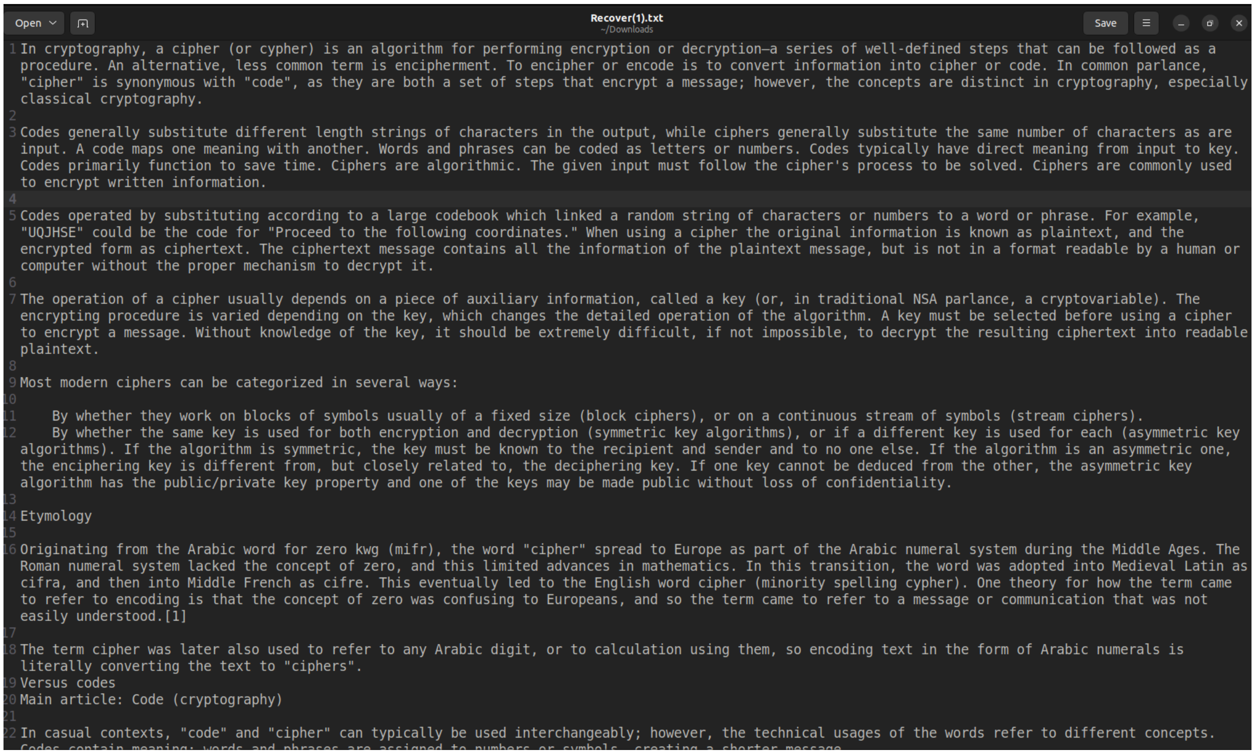
Ymj yjwr hnumjw bfx qfyjw fqxt zxji yt wjkjw yt fsd Fwfgnh inlny, tw yt hfqhzqfynts zxnsl ymjr, xt jshtinsl yjcy ns ymj ktwr tk Fwfgnh szrjwfqx nx qnyjwfqqd htsajwynsl ymj yjcy yt "hnumjwx".
Ajwxzx htijx
Rfns fwynhqj: Htij (hwduytlwfumd)

Ns hfxzfq htsyjcyx, "htij" fsi "hnumjw" hfs ydunhfqqd gj zxji nsyjwhmfsljfgqd; mtbjajw, ymj yjhmsnhfq zxfljx tk ymj btwix wjkjw yt inkkjwjsy htshjuyx.
Htijx htsyfns rjfsnsl: btwix fsi umwfxjx fwj fxxnlsji yt szrgjwx tw xdrgtqx, hwjfynsl f xmtwyjw rjxxflj

Encrypted.txt


```
--- Caesar Cipher and Cryptanalysis ---
1. Encryption of Caesar Cipher
2. Decryption of Caesar Cipher
3. Cryptanalysis using Brute Force
4. Cryptanalysis using Frequency Analysis
5. Exit
Enter your choice (1-5): 2
Enter input file name (Cipher.txt): Cipher.txt
Enter shift value for decryption: 5
Enter decrypted file name (Recover.txt): Recover.txt
Decryption completed. Decrypted text saved to Recover.txt
```

Option 2: Decrypt Text with key =5



In cryptography, a cipher (or cypher) is an algorithm for performing encryption or decryption—a series of well-defined steps that can be followed as a procedure. An alternative, less common term is encipherment. To encipher or encode is to convert information into cipher or code. In common parlance, "cipher" is synonymous with "code", as they are both a set of steps that encrypt a message; however, the concepts are distinct in cryptography, especially classical cryptography.

Codes generally substitute different length strings of characters in the output, while ciphers generally substitute the same number of characters as are input. A code maps one meaning with another. Words and phrases can be coded as letters or numbers. Codes typically have direct meaning from input to key. Codes primarily function to save time. Ciphers are algorithmic. The given input must follow the cipher's process to be solved. Ciphers are commonly used to encrypt written information.

Codes operated by substituting according to a large codebook which linked a random string of characters or numbers to a word or phrase. For example, "UQJHSE" could be the code for "Proceed to the following coordinates." When using a cipher the original information is known as plaintext, and the encrypted form as ciphertext. The ciphertext message contains all the information of the plaintext message, but is not in a format readable by a human or computer without the proper mechanism to decrypt it.

The operation of a cipher usually depends on a piece of auxiliary information, called a key (or, in traditional NSA parlance, a cryptovariable). The encrypting procedure is varied depending on the key, which changes the detailed operation of the algorithm. A key must be selected before using a cipher to encrypt a message. Without knowledge of the key, it should be extremely difficult, if not impossible, to decrypt the resulting ciphertext into readable plaintext.

Most modern ciphers can be categorized in several ways:

    By whether they work on blocks of symbols usually of a fixed size (block ciphers), or on a continuous stream of symbols (stream ciphers).
    By whether the same key is used for both encryption and decryption (symmetric key algorithms), or if a different key is used for each (asymmetric key algorithms). If the algorithm is symmetric, the key must be known to the recipient and sender and to no one else. If the algorithm is an asymmetric one, the enciphering key is different from, but closely related to, the deciphering key. If one key cannot be deduced from the other, the asymmetric key algorithm has the public/private key property and one of the keys may be made public without loss of confidentiality.

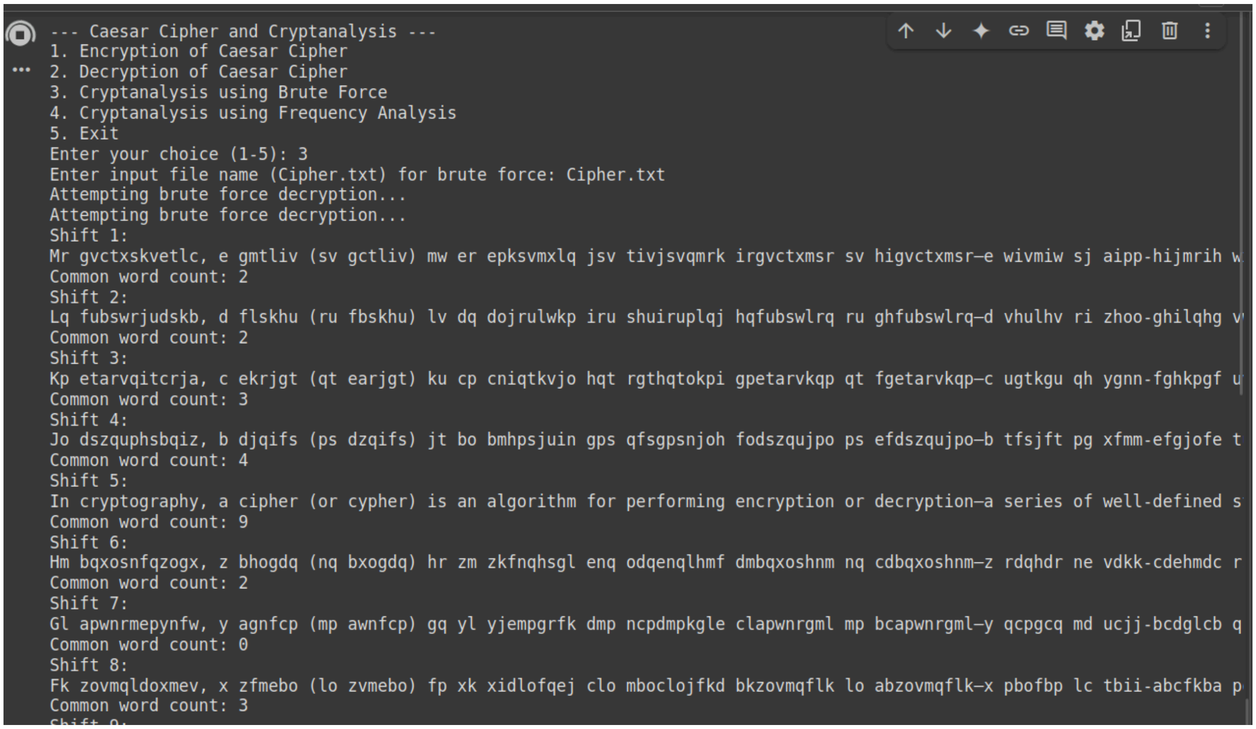
Etymology

Originating from the Arabic word for zero kwg (mifr), the word "cipher" spread to Europe as part of the Arabic numeral system during the Middle Ages. The Roman numeral system lacked the concept of zero, and this limited advances in mathematics. In this transition, the word was adopted into Medieval Latin as cifra, and then into Middle French as cifre. This eventually led to the English word cipher (minority spelling cypher). One theory for how the term came to refer to encoding is that the concept of zero was confusing to Europeans, and so the term came to refer to a message or communication that was not easily understood.[1]

The term cipher was later also used to refer to any Arabic digit, or to calculation using them, so encoding text in the form of Arabic numerals is literally converting the text to "ciphers".
Versus codes
Main article: Code (cryptography)

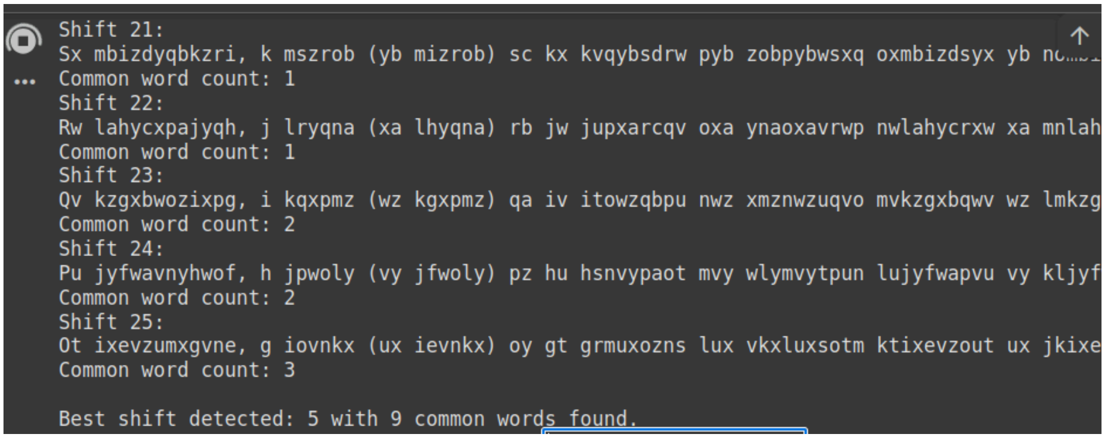
In casual contexts, "code" and "cipher" can typically be used interchangeably; however, the technical usages of the words refer to different concepts. Codes contain meaning; words and phrases are assigned to numbers or symbols, creating a shorter message.

Decrypted.txt

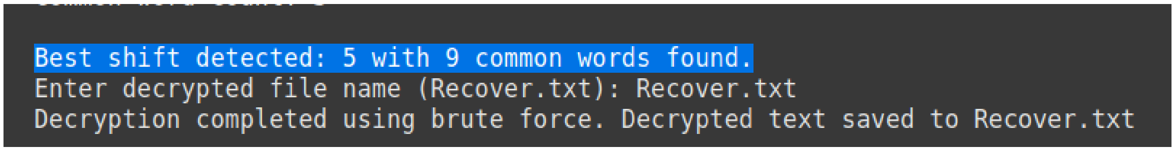


```
--- Caesar Cipher and Cryptanalysis ---
1. Encryption of Caesar Cipher
2. Decryption of Caesar Cipher
3. Cryptanalysis using Brute Force
4. Cryptanalysis using Frequency Analysis
5. Exit
Enter your choice (1-5): 3
Enter input file name (Cipher.txt) for brute force: Cipher.txt
Attempting brute force decryption...
Attempting brute force decryption...
Shift 1:
Mr gvctxskvetlc, e gmtliv (sv gctliv) mw er epksvmxlq jsv tivjsvqmrk irgvctxmsr sv higvctxmsr—e wivmiw sj aipp-hijmrih w
Common word count: 2
Shift 2:
Lq fubswrjudskb, d flskhu (ru fbskhu) lv dq dojrulwkp iru shuiruplqj hqfubswlrq ru ghfubswlrq—d vhulhv ri zhoo-ghilqhg v
Common word count: 2
Shift 3:
Kp etarvqitcrja, c ekrjgt (qt earjgt) ku cp cniqtkvjo hqt rgthqtokpi gpetarvkqp qt fgetarvkqp—c ugtkgu qh ygnn-fghkpgf u
Common word count: 3
Shift 4:
Jo dszquphsbqiz, b djqifs (ps dzqifs) jt bo bmhpsjuin gps qfsgpsnjoh fodszqujpo ps efdszqujpo—b tfsjft pg xfmm-efgjofe t
Common word count: 4
Shift 5:
In cryptography, a cipher (or cypher) is an algorithm for performing encryption or decryption—a series of well-defined s
Common word count: 9
Shift 6:
Hm bqxosnfqzogx, z bhogdq (nq bxogdq) hr zm zkfnqhsgl enq odqenqlhmf dmbqxoshnm nq cdbqxoshnm—z rdqhdr ne vdkk-cdehmdc r
Common word count: 2
Shift 7:
Gl apwnrmepynfw, y agnfcp (mp awnfcp) gq yl yjempgrfk dmp ncpdmpkgle clapwnrgml mp bcapwnrgml—y qcpgcq md ucjj-bcdglcb q
Common word count: 0
Shift 8:
Fk zovmqldoxmev, x zfmebo (lo zvmebo) fp xk xidlofqej clo mboclojfkd bkzovmqflk lo abzovmqflk—x pbofbp lc tbii-abcfkba p
Common word count: 3
```

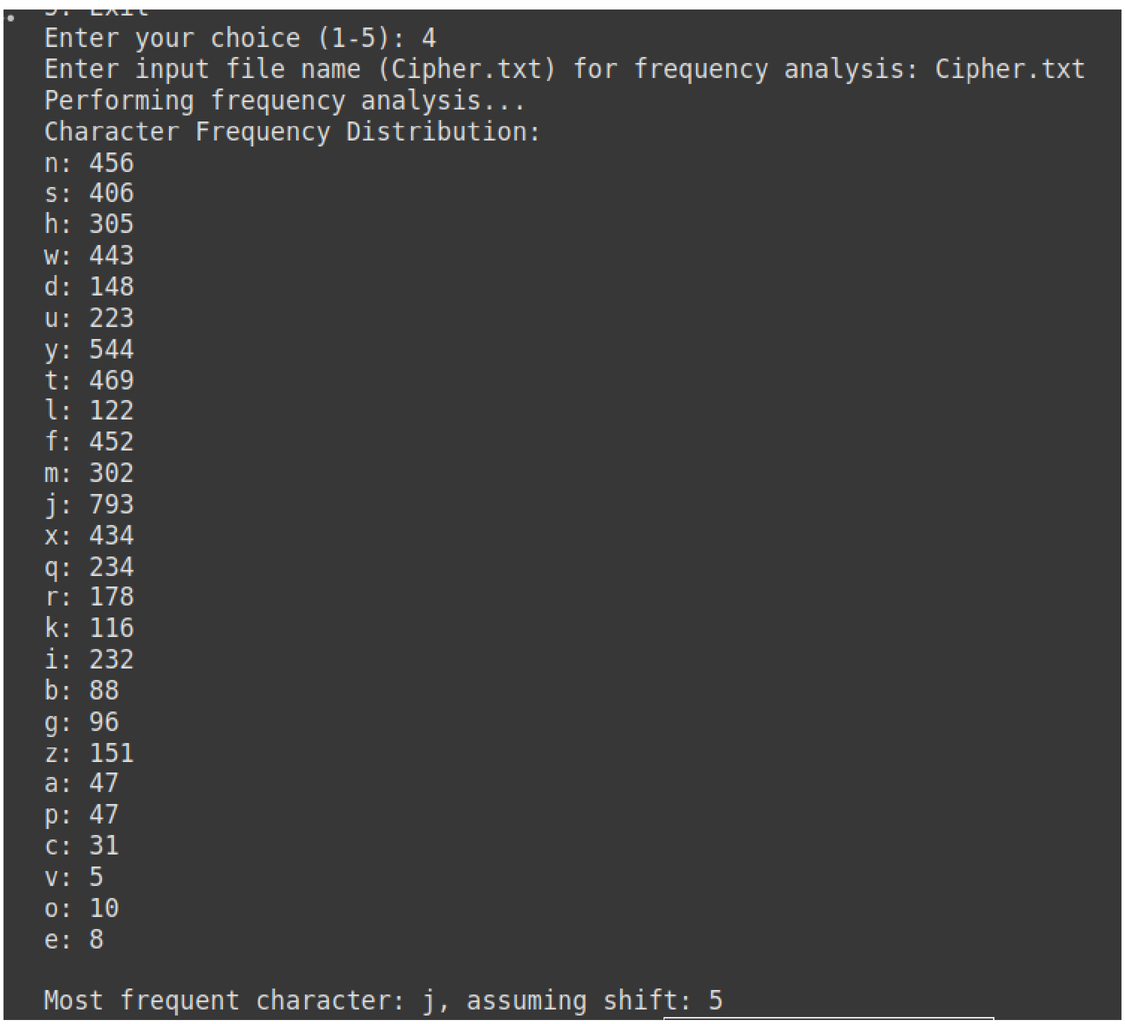
Option 3: Brute Force on common words and matching with original text



```
Shift 21:
Sx mbizdyqbkzri, k mszrob (yb mizrob) sc kx kvqybsdrw pyb zobpybwsxq oxmbizdsyx yb no
Common word count: 1
Shift 22:
Rw lahycxpajyqh, j lryqna (xa lhyqna) rb jw jupxarcqv oxa ynaoxavrwp nwlahycrxw xa mnlah
Common word count: 1
Shift 23:
Qv kzgxbwozixpg, i kqxpmz (wz kgxpmz) qa iv itowzqbpu nwz xmznwzuqvo mvkzgxbqwv wz lmkzg
Common word count: 2
Shift 24:
Pu jyfwavnyhwof, h jpwoly (vy jfwoly) pz hu hsnvypaot mvy wlymvytpun lujyfwapvu vy kljyf
Common word count: 2
Shift 25:
Ot ixevzumxgvne, g iovnkx (ux ievnkx) oy gt grmuxozns lux vkxluxsotm ktixevzout ux jkixe
Common word count: 3

Best shift detected: 5 with 9 common words found.
```

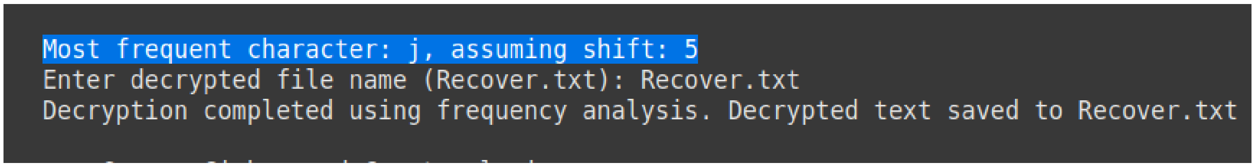


```
Common word count: 3

Best shift detected: 5 with 9 common words found.
Enter decrypted file name (Recover.txt): Recover.txt
Decryption completed using brute force. Decrypted text saved to Recover.txt
```



```
5. Exit
Enter your choice (1-5): 4
Enter input file name (Cipher.txt) for frequency analysis: Cipher.txt
Performing frequency analysis...
Character Frequency Distribution:
n: 456
s: 406
h: 305
w: 443
d: 148
u: 223
y: 544
t: 469
l: 122
f: 452
m: 302
j: 793
x: 434
q: 234
r: 178
k: 116
i: 232
b: 88
g: 96
z: 151
a: 47
p: 47
c: 31
v: 5
o: 10
e: 8

Most frequent character: j, assuming shift: 5
```

Option 4: Print out of Most Frequent character in “Cipher.txt” and then compared and found shift



```
Most frequent character: j, assuming shift: 5
Enter decrypted file name (Recover.txt): Recover.txt
Decryption completed using frequency analysis. Decrypted text saved to Recover.txt
```

Conclusion : In this assignment, I created a program to encrypt and decrypt text using the Caesar cipher and also built methods for cracking the cipher through brute force and frequency analysis. The Caesar cipher is a classic encryption technique that shifts letters in the alphabet by a set number, and I implemented functions to handle both encryption and decryption by shifting the letters forward or backward.

For cryptanalysis, I used two approaches. The brute force method involves trying all possible shifts (from 1 to 25) to decrypt a message. Instead of requiring manual input, the program automatically detects the most likely shift by checking for the presence of common English words, making the process faster and more efficient. The frequency analysis method assumes that the most frequent letter in a ciphertext corresponds to 'e', the most common letter in English. By examining the frequency of characters in the ciphertext, the program can figure out the correct shift and decrypt the text.

I also integrated file handling, so users can easily input text from files, perform encryption or decryption, and save the results into output files. This added flexibility, allowing the program to handle larger pieces of text efficiently. I made sure the program was user-friendly by providing a clear menu and keeping the process as automatic as possible.

Overall, this assignment taught me a lot about both cryptography and cryptanalysis. It showed me how classical ciphers like the Caesar cipher work and how they can be cracked using straightforward methods. It also highlighted the power of automation in cryptanalysis, making the process quicker and more accurate. This experience was a great way to dive deeper into the world of encryption and decryption.