# Programming Assignment-1

Abhinav Ujjawal (2021120) & Nikhil Suri (2021268)

## Monoalphabetic Substitution Cipher (Project-0)

Mono-alphabetic cipher is a substitution cipher where each letter of the plain text is replaced with another letter of the alphabet. In our case, the universe contains only three letters that are ‘A’, ‘B’ and ‘C’.

Also, the encryption key that we have used comprises 9 entries in a map where every entry corresponds to a relation between a pair of characters and another pair of characters.

Key used for the assignment:

| Mapped from | Mapped to |
|---|---|
| ‘AB’ | ‘BC’ |
| 'AC' | 'CB' |
| 'BA' | 'CA' |
| 'BC' | 'CC' |
| 'CA' | 'BB' |
| 'CB' | 'AA' |
| 'AA' | 'AB' |
| 'BB' | 'AC' |
| 'CC' | 'BA' |

## Encryption and decryption

### Encryption

Encrypt the PlainText into CipherText, using the encryption key. The encryption function is called on the entire PlainText, which consists of originalText + Hash(of originalText). The key is obtained in the code by reading a dictionary which consists of key-value pairs.

Encryption is simply an O(n) procedure, where n = size of PlainText.

## Decryption

Decrypt the CipherText into PlainText, using the decryption key. The key is obtained in the code by reading a dictionary which consists of key-value pairs, but this time, in the reverse order.

Decryption is also an O(n) procedure, where n = size of CipherText.

# Hashing and Recognisability

## Hashing

We use this hash function to construct recognisable plaintexts, i.e., those that satisfy the property: p = (s, Hash(s)).

1) Initialize the initial Hash Value = “0000000000000000” (16-bit string of 0s)
2) Divide the input plaintext into blocks of N-character segments, where N is a constant specified in the implementation – in our implementation, N = 16
3) For each block of characters, do the following:
    - Rotate the current hash value to the left by one bit
    - XOR the block with the hash value, and store the result as the new Hash Value
4) Encode the bits in hash in this criteria so that it becomes encryptable:
    - 0 - A
    - 1 - B
5) Return the Hash Value

**<u>Note</u>**: Hash Value will be used in the form of a string of characters, instead of 0s and 1s. This is because we have to make the hash value encryptable as well.
(So we map the 0 to A and 1 to B after calculating the hash)

## Recognisability

The function “is_recognizable” checks whether a particular PlainText is recognizable or not. The function works as follows:

1) Calculate the hash value of the Binary(candidate PlainText) by calling the hash function, where Binary(.) denotes the binary equivalent of the PlainText, as per ASCII values.
2) Compare the calculated hash value (after converting into a string) with the expected hash value, and check for equality.
3) If the two values are equal, return "True", indicating that the candidate PlainText is recognizable. Else, return "False", indicating that the candidate PlainText is not recognizable.

## Brute-force solution

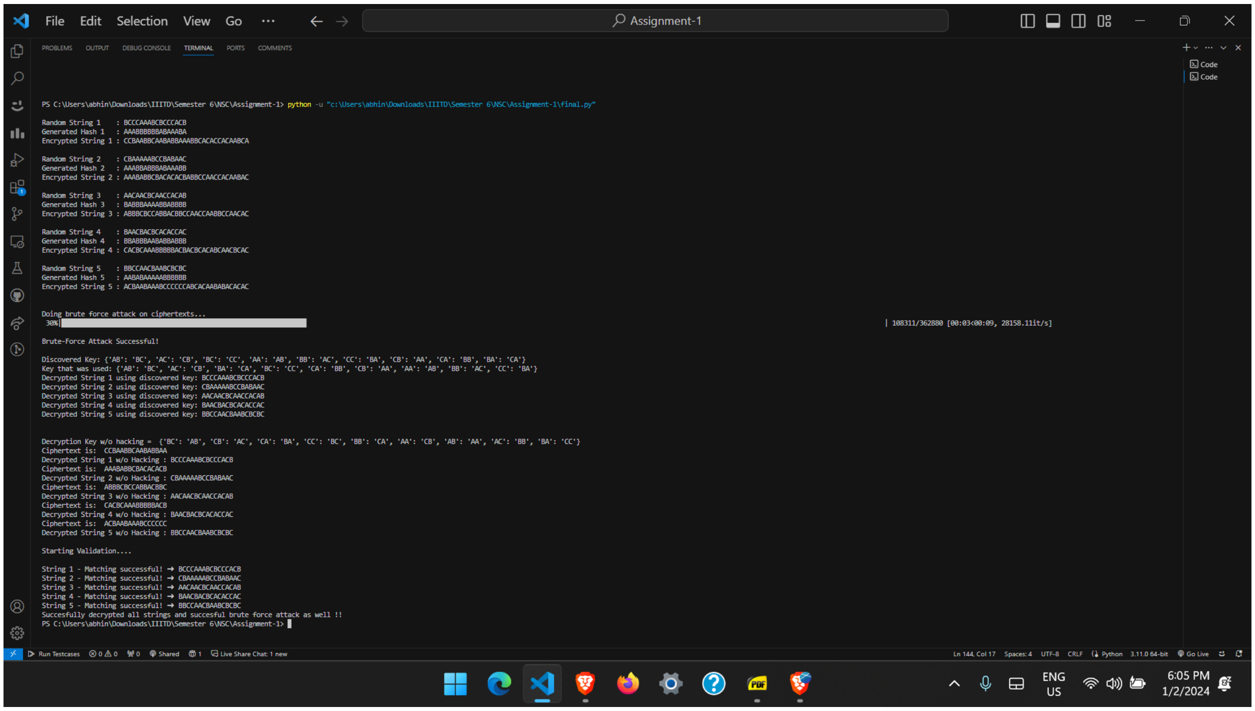
- We get all the possible combinations of the key (9!=362,880). Iterate through these combinations and try to use them to decrypt the first CipherText.
- If it is recognisable after decrypting, try this candidate key over other CipherTexts.
- The key is found when all the 5 CipherTexts are correctly decrypted using the candidate key. Return this candidate key.

The asymptotic time complexity of discovering the key via brute force is **O(9!)** in our implementation. More generally, Brute-force is O((n^2)!), where n = number of symbols in the universe.

## Constraints

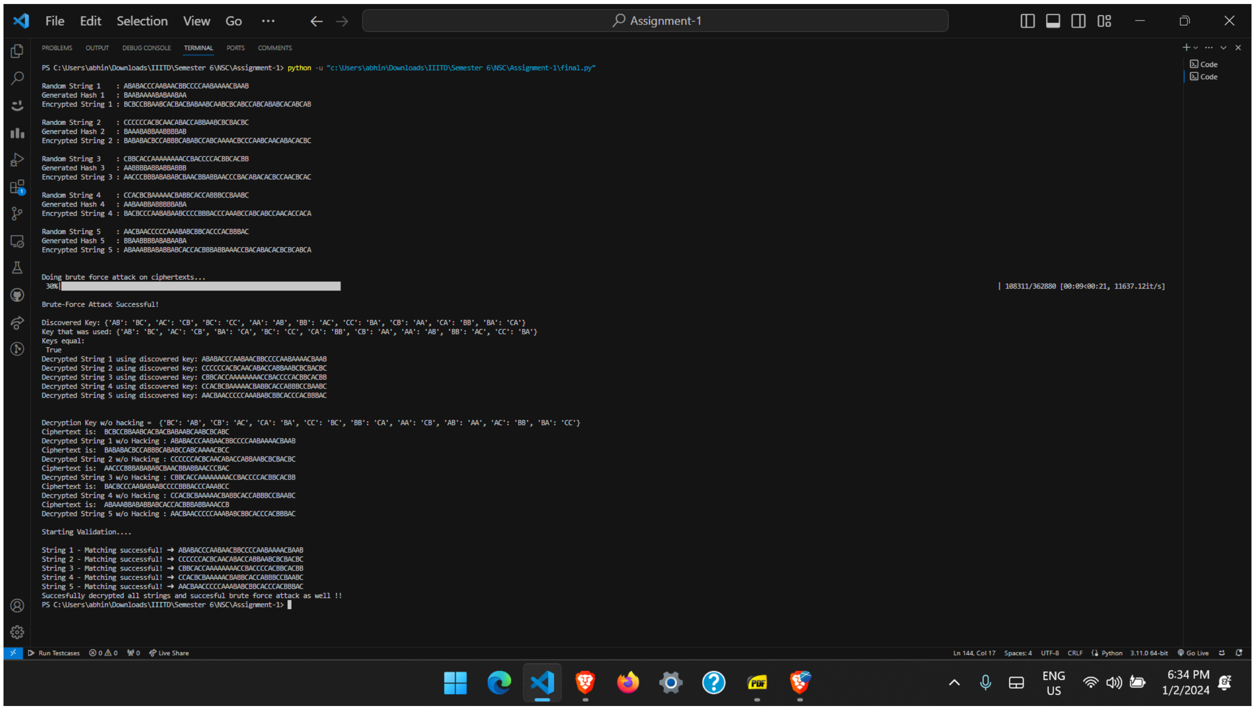
- Plaintext characters must belong to the set {A, B, C}
- The plaintext must be of even length for encryption to work.
- The plaintext must satisfy the property $\pi$ = (s, Hash(s))
- Size of key = 9

## Sample Inputs & Outputs

1. 

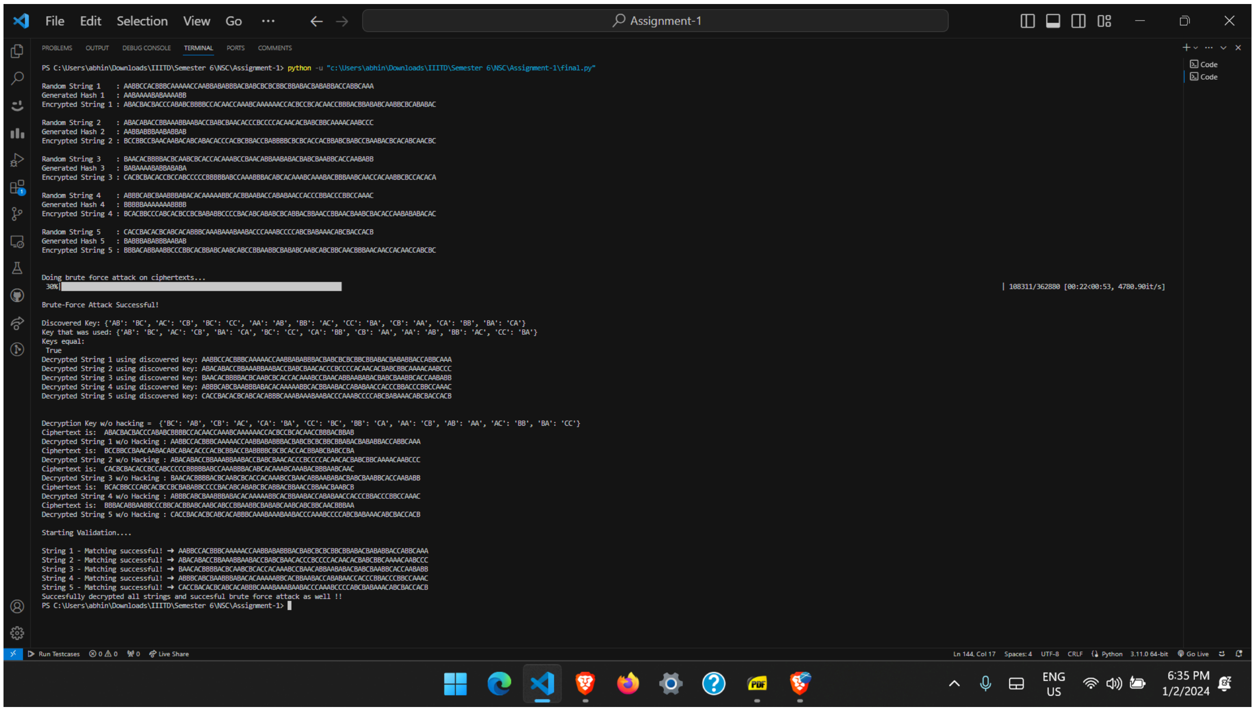
```
PS C:\Users\abhin\Downloads\IIITD\Semester 6\NSC\Assignment-1> python -u "c:\Users\abhin\Downloads\IIITD\Semester 6\NSC\Assignment-1\final.py"

Random String 1    : BCCCAAABCBCCCACB
Generated Hash 1   : AAABBBBBBABAAABA
Encrypted String 1 : CCBAABBCAABABBAAABBCACACCACAABCA

Random String 2    : CBAAAAABCCBABAAC
Generated Hash 2   : AAABBABBBABAAABB
Encrypted String 2 : AAABABBCBACACACBABBCCAACCACAABAC

Random String 3    : AACAACBCAACCACAB
Generated Hash 3   : BABBBAAAABBABBBB
Encrypted String 3 : ABBBCBCCABBACBBCCAACCAABBCCAACAC

Random String 4    : BAACBACBCACACCAC
Generated Hash 4   : BBABBBAABABBABBB
Encrypted String 4 : CACBCAAABBBBBACBACBCACABCAACBCAC

Random String 5    : BBCCAACBAABCBCBC
Generated Hash 5   : AABABAAAAABBBBBB
Encrypted String 5 : ACBAABAAABCCCCCCABCACAABABACACAC


Doing brute force attack on ciphertexts...
 30%|████████████████████                                              | 108311/362880 [00:03<00:09, 28158.11it/s]

Brute-Force Attack Successful!

Discovered Key: {'AB': 'BC', 'AC': 'CB', 'BC': 'CC', 'AA': 'AB', 'BB': 'AC', 'CC': 'BA', 'CB': 'AA', 'CA': 'BB', 'BA': 'CA'}
Key that was used: {'AB': 'BC', 'AC': 'CB', 'BA': 'CA', 'BC': 'CC', 'CA': 'BB', 'CB': 'AA', 'AA': 'AB', 'BB': 'AC', 'CC': 'BA'}
Decrypted String 1 using discovered key: BCCCAAABCBCCCACB
Decrypted String 2 using discovered key: CBAAAAABCCBABAAC
Decrypted String 3 using discovered key: AACAACBCAACCACAB
Decrypted String 4 using discovered key: BAACBACBCACACCAC
Decrypted String 5 using discovered key: BBCCAACBAABCBCBC


Decryption Key w/o hacking =  {'BC': 'AB', 'CB': 'AC', 'CA': 'BA', 'CC': 'BC', 'BB': 'CA', 'AA': 'CB', 'AB': 'AA', 'AC': 'BB', 'BA': 'CC'}
Ciphertext is:  CCBAABBCAABABBAA
Decrypted String 1 w/o Hacking : BCCCAAABCBCCCACB
Ciphertext is:  AAABABBCBACACACB
Decrypted String 2 w/o Hacking : CBAAAAABCCBABAAC
Ciphertext is:  ABBBCBCCABBACBBC
Decrypted String 3 w/o Hacking : AACAACBCAACCACAB
Ciphertext is:  CACBCAAABBBBBACB
Decrypted String 4 w/o Hacking : BAACBACBCACACCAC
Ciphertext is:  ACBAABAAABCCCCCC
Decrypted String 5 w/o Hacking : BBCCAACBAABCBCBC

Starting Validation....

String 1 - Matching successful! → BCCCAAABCBCCCACB
String 2 - Matching successful! → CBAAAAABCCBABAAC
String 3 - Matching successful! → AACAACBCAACCACAB
String 4 - Matching successful! → BAACBACBCACACCAC
String 5 - Matching successful! → BBCCAACBAABCBCBC
Succesfully decrypted all strings and succesful brute force attack as well !!
PS C:\Users\abhin\Downloads\IIITD\Semester 6\NSC\Assignment-1> 
```

2.



```
PS C:\Users\abhin\Downloads\IIITD\Semester 6\NSC\Assignment-1> python -u "c:\Users\abhin\Downloads\IIITD\Semester 6\NSC\Assignment-1\final.py"

Random String 1    : ABABACCCAABAACBBCCCCAABAAAACBAAB
Generated Hash 1   : BAABAAAABABAABAA
Encrypted String 1 : BCBCCBBAABCACBACBABAABCAABCBCABCCABCABABCACABCAB

Random String 2    : CCCCCCACBCAACABACCABBAABCBCBACBC
Generated Hash 2   : BAAABABBAABBBBAB
Encrypted String 2 : BABABACBCCABBBCABABCCABCAAAACBCCCAABCAACABACACBC

Random String 3    : CBBCACCAAAAAAAACCBACCCCACBBCACBB
Generated Hash 3   : AABBBBABBABBABBB
Encrypted String 3 : AACCCBBBABABABCBAACBBABBAACCCBACABACACBCCAACBCAC

Random String 4    : CCACBCBAAAAACBABBCACCABBBCCBAABC
Generated Hash 4   : AABAABBABBBBBABA
Encrypted String 4 : BACBCCCAABABAABCCCCBBBACCCAAABCCABCABCCAACACCACA

Random String 5    : AACBAACCCCCAAABABCBBCACCCACBBBAC
Generated Hash 5   : BBAABBBBABABAABA
Encrypted String 5 : ABAAABBABABBABCACCACBBBABBAAACCBACABACACBCBCABCA


Doing brute force attack on ciphertexts...
 30%|                                                  | 108311/362880 [00:09<00:21, 11637.12it/s]

Brute-Force Attack Successful!

Discovered Key: {'AB': 'BC', 'AC': 'CB', 'BC': 'CC', 'AA': 'AB', 'BB': 'AC', 'CC': 'BA', 'CB': 'AA', 'CA': 'BB', 'BA': 'CA'}
Key that was used: {'AB': 'BC', 'AC': 'CB', 'BA': 'CA', 'BC': 'CC', 'CA': 'BB', 'CB': 'AA', 'AA': 'AB', 'BB': 'AC', 'CC': 'BA'}
Keys equal:
 True
Decrypted String 1 using discovered key: ABABACCCAABAACBBCCCCAABAAAACBAAB
Decrypted String 2 using discovered key: CCCCCCACBCAACABACCABBAABCBCBACBC
Decrypted String 3 using discovered key: CBBCACCAAAAAAAACCBACCCCACBBCACBB
Decrypted String 4 using discovered key: CCACBCBAAAAACBABBCACCABBBCCBAABC
Decrypted String 5 using discovered key: AACBAACCCCCAAABABCBBCACCCACBBBAC


Decryption Key w/o hacking =  {'BC': 'AB', 'CB': 'AC', 'CA': 'BA', 'CC': 'BC', 'BB': 'CA', 'AA': 'CB', 'AB': 'AA', 'AC': 'BB', 'BA': 'CC'}
Ciphertext is:  BCBCCBBAABCACBACBABAABCAABCBCABC
Decrypted String 1 w/o Hacking : ABABACCCAABAACBBCCCCAABAAAACBAAB
Ciphertext is:  BABABACBCCABBBCABABCCABCAAAACBCC
Decrypted String 2 w/o Hacking : CCCCCCACBCAACABACCABBAABCBCBACBC
Ciphertext is:  AACCCBBBABABABCBAACBBABBAACCCBAC
Decrypted String 3 w/o Hacking : CBBCACCAAAAAAAACCBACCCCACBBCACBB
Ciphertext is:  BACBCCCAABABAABCCCCBBBACCCAAABCC
Decrypted String 4 w/o Hacking : CCACBCBAAAAACBABBCACCABBBCCBAABC
Ciphertext is:  ABAAABBABABBABCACCACBBBABBAAACCB
Decrypted String 5 w/o Hacking : AACBAACCCCCAAABABCBBCACCCACBBBAC

Starting Validation....

String 1 - Matching successful! → ABABACCCAABAACBBCCCCAABAAAACBAAB
String 2 - Matching successful! → CCCCCCACBCAACABACCABBAABCBCBACBC
String 3 - Matching successful! → CBBCACCAAAAAAAACCBACCCCACBBCACBB
String 4 - Matching successful! → CCACBCBAAAAACBABBCACCABBBCCBAABC
String 5 - Matching successful! → AACBAACCCCCAAABABCBBCACCCACBBBAC
Succesfully decrypted all strings and succesful brute force attack as well !!
PS C:\Users\abhin\Downloads\IIITD\Semester 6\NSC\Assignment-1> 
```

3.



```
PS C:\Users\abhin\Downloads\IIITD\Semester 6\NSC\Assignment-1> python -u "c:\Users\abhin\Downloads\IIITD\Semester 6\NSC\Assignment-1\final.py"

Random String 1    : AABBCCACBBBCAAAAACCAABBABABBBACBABCBCBCBBCBBABACBABABBACCABBCAAA
Generated Hash 1   : AABAAAABABAAAABB
Encrypted String 1 : ABACBACBACCCABABCBBBBCCACAACCAAABCAAAAAACCACBCCBCACAACCBBBACBBABABCAABBCBCABABAC

Random String 2    : ABACABACCBBAAABBAABACCBABCBAACACCCBCCCCACAACACBABCBBCAAAACAABCCC
Generated Hash 2   : AABBABBBAABABBAB
Encrypted String 2 : BCCBBCCBAACAABACABCABACACCCACBCBBACCBABBBBCBCBCACCACBBABCBABCCBAABACBCACABCAACBC

Random String 3    : BAACACBBBBACBCAABCBCACCACAAABCCBAACABBAABABACBABCBAABBCACCAABABB
Generated Hash 3   : BABAAAABABBABABA
Encrypted String 3 : CACBCBACACCBCCABCCCCCBBBBBABCCAAABBBACABCACAAABCAAABACBBBAABCAACCACAABBCBCCACACA

Random String 4    : ABBBCABCBAABBBABACACAAAAABBCACBBAABACCABABAACCACCCBBACCCBBCCAAAC
Generated Hash 4   : BBBBBAAAAAAABBBB
Encrypted String 4 : BCACBBCCCABCACBCCBCBABABBCCCCBACABCABABCBCABBACBBAACCBBAACBAABCBACACCAABABABACAC

Random String 5    : CACCBACACBCABCACABBBCAAABAAABAABACCCAAABCCCCABCBABAAACABCBACCACB
Generated Hash 5   : BABBBABABBBAABAB
Encrypted String 5 : BBBACABBAABBCCCBBCACBBABCAABCABCCBBAABBCBABABCAABCABCBBCAACBBBAACAACCACAACCABCBC


Doing brute force attack on ciphertexts...
 30%|                                                  | 108311/362880 [00:22<00:53, 4780.90it/s]

Brute-Force Attack Successful!

Discovered Key: {'AB': 'BC', 'AC': 'CB', 'BC': 'CC', 'AA': 'AB', 'BB': 'AC', 'CC': 'BA', 'CB': 'AA', 'CA': 'BB', 'BA': 'CA'}
Key that was used: {'AB': 'BC', 'AC': 'CB', 'BA': 'CA', 'BC': 'CC', 'CA': 'BB', 'CB': 'AA', 'AA': 'AB', 'BB': 'AC', 'CC': 'BA'}
Keys equal:
 True
Decrypted String 1 using discovered key: AABBCCACBBBCAAAAACCAABBABABBBACBABCBCBCBBCBBABACBABABBACCABBCAAA
Decrypted String 2 using discovered key: ABACABACCBBAAABBAABACCBABCBAACACCCBCCCCACAACACBABCBBCAAAACAABCCC
Decrypted String 3 using discovered key: BAACACBBBBACBCAABCBCACCACAAABCCBAACABBAABABACBABCBAABBCACCAABABB
Decrypted String 4 using discovered key: ABBBCABCBAABBBABACACAAAAABBCACBBAABACCABABAACCACCCBBACCCBBCCAAAC
Decrypted String 5 using discovered key: CACCBACACBCABCACABBBCAAABAAABAABACCCAAABCCCCABCBABAAACABCBACCACB


Decryption Key w/o hacking =  {'BC': 'AB', 'CB': 'AC', 'CA': 'BA', 'CC': 'BC', 'BB': 'CA', 'AA': 'CB', 'AB': 'AA', 'AC': 'BB', 'BA': 'CC'}
Ciphertext is:  ABACBACBACCCABABCBBBBCCACAACCAAABCAAAAAACCACBCCBCACAACCBBBACBBAB
Decrypted String 1 w/o Hacking : AABBCCACBBBCAAAAACCAABBABABBBACBABCBCBCBBCBBABACBABABBACCABBCAAA
Ciphertext is:  BCCBBCCBAACAABACABCABACACCCACBCBBACCBABBBBCBCBCACCACBBABCBABCCBA
Decrypted String 2 w/o Hacking : ABACABACCBBAAABBAABACCBABCBAACACCCBCCCCACAACACBABCBBCAAAACAABCCC
Ciphertext is:  CACBCBACACCBCCABCCCCCBBBBBABCCAAABBBACABCACAAABCAAABACBBBAABCAAC
Decrypted String 3 w/o Hacking : BAACACBBBBACBCAABCBCACCACAAABCCBAACABBAABABACBABCBAABBCACCAABABB
Ciphertext is:  BCACBBCCCABCACBCCBCBABABBCCCCBACABCABABCBCABBACBBAACCBBAACBAABCB
Decrypted String 4 w/o Hacking : ABBBCABCBAABBBABACACAAAAABBCACBBAABACCABABAACCACCCBBACCCBBCCAAAC
Ciphertext is:  BBBACABBAABBCCCBBCACBBABCAABCABCCBBAABBCBABABCAABCABCBBCAACBBBAA
Decrypted String 5 w/o Hacking : CACCBACACBCABCACABBBCAAABAAABAABACCCAAABCCCCABCBABAAACABCBACCACB

Starting Validation....

String 1 - Matching successful! → AABBCCACBBBCAAAAACCAABBABABBBACBABCBCBCBBCBBABACBABABBACCABBCAAA
String 2 - Matching successful! → ABACABACCBBAAABBAABACCBABCBAACACCCBCCCCACAACACBABCBBCAAAACAABCCC
String 3 - Matching successful! → BAACACBBBBACBCAABCBCACCACAAABCCBAACABBAABABACBABCBAABBCACCAABABB
String 4 - Matching successful! → ABBBCABCBAABBBABACACAAAAABBCACBBAABACCABABAACCACCCBBACCCBBCCAAAC
String 5 - Matching successful! → CACCBACACBCABCACABBBCAAABAAABAABACCCAAABCCCCABCBABAAACABCBACCACB
Succesfully decrypted all strings and succesful brute force attack as well !!
PS C:\Users\abhin\Downloads\IIITD\Semester 6\NSC\Assignment-1> 
```