In [1]:

```
import numpy as np
import pandas as pd
import time
```

Encryption

function take input and outputs

In [2]:

```
def encryption():
    text = input("what is the plain text?  ")
    #upper case the text and split into list of char then ASCII code, then number code fo
    textlist = [*text.upper()]
    textnumlist = list(map(lambda x:ord(x)-65,textlist))

    #prepare the key into number list
    key = input("What is the key for encryption? ")
    keylist = [*key.upper()]
    keynumlist = list(map(lambda x:ord(x)-65,keylist))

    #encryption the number list of cipher text
    codenumlist = []
    for i in range(len(textnumlist)):
        keypos = i%len(key)
        codenumlist.append(textnumlist[i]+keynumlist[keypos])
        codenumlist = list(map(lambda x:x%26,codenumlist))

    #get the ascii list of cyphertext, and convert into textlist, then combine
    ciphertext = ''.join(list(map(lambda x: chr(x),list(map(lambda x:x+65,codenumlist)))))
    return print("The encrypted text is: "+ ciphertext)
```

In [3]:

```
encryption()
```

```
what is the plain text?  EthanlikeCS
What is the key for encryption? Sky
The encrypted text is: WDFSXJAUCUC
```

Decipher

In [ ]:

```
# take ciphertext, keylength, firstwordlength.
# key can be anything. check if the plaintext first word match the dictionary
```

Method

In [2]:

```
def decryption(ciphertext,keylength,firstwordlength):
    globalstart = time.time()
    #prepare the num list of the first word of the ciphertext
    cipherwordnumlist = list(map(lambda x:ord(x)-65,[*ciphertext.upper()][:firstwordlengt
    ciphernumlist = list(map(lambda x:ord(x)-65,[*ciphertext.upper()]))

    #prepare all the key combination
    keylistlist = []
    def printAllKLength(set, k):
        n = len(set)
        printAllKLengthRec(set, [], n, k)

    def printAllKLengthRec(set, prefix, n, k):
        if (k == 0) :
            keylistlist.append(prefix)
            return

        for i in range(n):

            # Next character of input added
            newlist = prefix.copy()
            newlist.append(set[i])

            printAllKLengthRec(set, newlist, n, k - 1)



    start = time.time()
    printAllKLength([0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25]
    end = time.time()
    print("preparing the key list take : " + str(end - start) + "s")


    #decipher
    start = time.time()
    deciphernumlistlist = []
    deciphernumlist = []
    for j in keylistlist:
        deciphernumlist = []
        for i in range(len(cipherwordnumlist)):
            keypos = i%keylength
            if(cipherwordnumlist[i] > j[keypos]):
                deciphernumlist.append((cipherwordnumlist[i] - j[keypos])%26)
            else:
                deciphernumlist.append((cipherwordnumlist[i] + 26 - j[keypos])%26)

        deciphernumlistlist.append(deciphernumlist)
    end = time.time()
    print("decipher the text using all the key take: " + str(end - start) + 's.')
    #transform numlist list into textlist list
    deciphertextlistlist= []
    for i in deciphernumlistlist:
        deciphertextlist = list(map(lambda x:chr(x+65),i))
        deciphertextlistlist.append(deciphertextlist)

    #create a list of decipher text
    decipherlist = []
    for i in deciphertextlistlist:
        decipher = ''.join(i)
```

```
        decipherlist.append(decipher)

    #create a list of key text
    keyasclist = []
    for i in keylistlist:
        ascnum = list(map(lambda x:x+65,i))
        keyasclist.append(ascnum)

    keytextlistlist = []
    for i in keyasclist:
        keytext = list(map(lambda x:chr(x),i))
        keytextlistlist.append(keytext)

    keytextlist = []
    for i in keytextlistlist:
        key = ''.join(i)
        keytextlist.append(key)

    #create a text, key pair list
    decipherpairlist = []
    for i in range(len(decipherlist)):
        pair = (decipherlist[i],keytextlist[i],keylistlist[i])
        decipherpairlist.append(pair)


    #load the dictionary in memeory
    #check with dictionary
    with open("MP1_dict.txt", "r") as f:
        all_lines = f.readlines()

    dictionary = list(map(lambda x:x.rstrip("\n"),all_lines))

    #check if the word contain memory
    start = time.time()
    print('the cipher text is: ' + ciphertext)
    print('============================================================')
    for i in decipherpairlist:
        for j in dictionary:
            if i[0] == j:
                wholeword = []
                for k in range(len(ciphernumlist)):
                    keypos = k%keylength
                    if(ciphernumlist[k] < i[2][keypos]):
                        wholeword.append((ciphernumlist[k] + 26 - i[2][keypos])%26 + 65)
                    else:
                        wholeword.append((ciphernumlist[k] - i[2][keypos])%26 + 65)
                wholetext = ''.join(list(map(lambda x:chr(x),wholeword)))
                print('A potential key is: ' + str(i[1]) + '  the first word will be: ' +
                print('The full sentence after decipher with this key is: ' + str(wholete
                print('============================================================')
    end = time.time()
    print('check with dictionary takes: ' + str(end-start) + 's.')

    globalend = time.time()
    print("the entire process take: " + str(globalend-globalstart) + 's')
```

In [156]:

```
decryption("WDFSXGKQMGN",3,5)
```

...

In [3]:

```
decryption("MSOKKJCOSXOEEKDTOSLGFWCMCHSUSGX",2,6)
```

```
preparing the key list take : 0.0s
decipher the text using all the key take: 0.0009996891021728516s.
the cipher text is: MSOKKJCOSXOEEKDTOSLGFWCMCHSUSGX
============================================================
A potential key is: KS  the first word will be: CAESAR
The full sentence after decipher with this key is: CAESARSWIFEMUSTBEABOVES
USPICION
============================================================
check with dictionary takes: 3.695997953414917s.
the entire process take: 3.7249975204467773s
```

In [4]:

```
decryption("PSPDYLOAFSGFREQKKPOERNIYVSDZSUOVGXSRRIPWERDIPCFSDIQZIASEJVCGXAYBGYXFPSREKFMEX
```

```
preparing the key list take : 0.005000591278076172s
decipher the text using all the key take: 0.04099917411804199s.
the cipher text is: PSPDYLOAFSGFREQKKPOERNIYVSDZSUOVGXSRRIPWERDIPCFSDIQZIA
SEJVCGXAYBGYXFPSREKFMEXEBIYDGFKREOWGXEQSXSKXGYRRRVMEKFFIPIWJSKFDJMBGCC
============================================================
A potential key is: KEY  the first word will be: FORTUNE
The full sentence after decipher with this key is: FORTUNEWHICHHASAGREATDE
ALOFPOWERINOTHERMATTERSBUTESPECIALLYINWARCANBRINGABOUTGREATCHANGESINASITUA
TIONTHROUGHVERYSLIGHTFORCES
============================================================
check with dictionary takes: 96.55033421516418s.
the entire process take: 96.65533328056335s
```

In [5]:

```
decryption("MTZHZEOQKASVBDOWMWMKMNYIIHVWPEXJA",4,10)
```

```
preparing the key list take : 0.3019993305206299s
decipher the text using all the key take: 0.881000280380249s.
the cipher text is: MTZHZEOQKASVBDOWMWMKMNYIIHVWPEXJA
============================================================
A potential key is: IWKD  the first word will be: EXPERIENCE
The full sentence after decipher with this key is: EXPERIENCEISTHETEACHERO
FALLTHINGS
============================================================
check with dictionary takes: 2555.552938222885s.
the entire process take: 2558.383937597275s
```

In [6]:

```
decryption("SQLIMXEEKSXMDOSBITOTYVECRDXSCRURZYPOHRG",5,11)
```

```
preparing the key list take : 6.763997316360474s
decipher the text using all the key take: 24.40999984741211s.
the cipher text is: SQLIMXEEKSXMDOSBITOTYVECRDXSCRURZYPOHRG
============================================================

---------------------------------------------------------------------------
-
KeyboardInterrupt                         Traceback (most recent call las
t)
Cell In[6], line 1
----> 1 decryption("SQLIMXEEKSXMDOSBITOTYVECRDXSCRURZYPOHRG",5,11)

Cell In[2], line 98, in decryption(ciphertext, keylength, firstwordlength)
     96 for i in decipherpairlist:
     97     for j in dictionary:
---> 98         if i[0] == j:
     99             wholeword = []
    100             for k in range(len(ciphernumlist)):

KeyboardInterrupt:
```

In [ ]:

```
decryption("LDWMEKPOPSWNOAVBIDHIPCEWAETYRVOAUPSINOVDIEDHCDSELHCCPVHRPOHZUSERSFS",6,9)
```

In [ ]:

```
decryption("VVVLZWWPBWHZDKBTXLDCGOTGTGRWAQWZSDHEMXLBELUMO",7,13)
```

In [ ]:

```

```

In [ ]:

```

```