CNS LAB 1

SAHIL BONDRE: U18CO021

Implement a menu driven program for Caesar Cipher with following functions:

a. Encrypt given plain text.
b. Decrypt given ciphertext.
c. Find encryption key using brute force attack.
d. Find encryption key using frequency analysis attack. (Frequency distribution of characters is attached)

Note: Consider file as an input in program. The program should work for large and variable length input text.

Q1 and Q2:

```
def recursive_read(allowed_input, message=""):
    # Recursively reads user input until input is not in allowd_input
    while True:
        user_input = input(message)
        if user_input in allowed_input:
            return user_input



def recursive_read_int(message=""):
    # Recursively reads user input until input is not in allowd_input
    while True:
        user_input = input(message)
        try:
            value = int(user_input)
```

```
            return value
        except:
            pass



def file_to_str(filename):
    try:
        with open(filename, 'r') as file:
            return file.read()
    except:
      print("Error: File not found!")
      exit(1)



def perform_encryption():
    filename = input("Enter file to be encrypted: ")
    message = file_to_str(filename)
    key = recursive_read_int("Enter key value: ")
    result = ""


    for i in range(len(message)):
        char = message[i]
        if char.isupper():
            result += chr((ord(char) + key - 65) % 26 + 65)
        elif char.islower():
            result += chr((ord(char) + key - 97) % 26 + 97)
```

```
        else:
            result += char

    print(f"Final string:\n{result}")


def perform_decyption():
    filename = input("Enter file to be decrypted: ")
    message = file_to_str(filename)
    key = recursive_read_int("Enter key value: ")
    result = ""

    for i in range(len(message)):
        char = message[i]
        if char.isupper():
            result += chr((ord(char) - key - 65) % 26 + 65)
        elif char.islower():
            result += chr((ord(char) - key - 97) % 26 + 97)
        else:
            result += char

    print(f"Final string:\n{result}")


is_encrypt = recursive_read(
    ["e", "d"], "Enter 'e' for encryption or 'd' for decryption: ") == "e"
```

```
if is_encrypt:
    perform_encryption()
else:
    perform_decyption()
```

message.txt

```
Sahil Bondre
Hello World
123456
```

encrypted.txt

```
Govwz Pcbrfs
Vszzc Kcfzr
123456
```

```
PS F:\code\github.com\godcrampy\college-notes\cns\lab-01> python.exe .\caesar.py
Enter 'e' for encryption or 'd' for decryption: e
Enter file to be encrypted: message.txt
Enter key value: 14
Final string:
Govwz Pcbrfs
Vszzc Kcfzr
123456

PS F:\code\github.com\godcrampy\college-notes\cns\lab-01> python.exe .\caesar.py
Enter 'e' for encryption or 'd' for decryption: d
Enter file to be decrypted: encrypted.txt
Enter key value: 14
Final string:
Sahil Bondre
Hello World
123456

PS F:\code\github.com\godcrampy\college-notes\cns\lab-01>
```

Q3:

```
def file_to_str(filename):
    try:
        with open(filename, 'r') as file:
            return file.read()
    except:
        print("Error: File not found!")
        exit(1)



def recursive_read(allowed_input, message=""):
    # Recursively reads user input until input is not in allowd_input
    while True:
        user_input = input(message)
        if user_input in allowed_input:
            return user_input



def recursive_read_int(message=""):
    # Recursively reads user input until input is not in allowd_input
    while True:
        user_input = input(message)
        try:
            value = int(user_input)
            return value
        except:
```

```
            pass



def shift(message, key):
    result = ""

    for i in range(len(message)):
        char = message[i]
        if char.isupper():
            result += chr((ord(char) + key - 65) % 26 + 65)
        elif char.islower():
            result += chr((ord(char) + key - 97) % 26 + 97)
        else:
            result += char

    return result



def head(message):
    return "\n".join(message.split("\n")[:4])

filename = input("Enter file to be encrypted: ")
message = file_to_str(filename)
while True:
    key = recursive_read_int("Enter next key to try: ")
    result = shift(message, key)
```

```
    print("Text: ")

    print(head(result))

    is_ok = recursive_read(

        ["y", "n"], "Enter 'y' to quit or 'n' for new key: ") == "y"

    if is_ok:

        print("Final Message:")

        print(result)

        break
```

encrypted.txt

```
Govwz Pcbrfs
Vszzc Kcfzr
123456
```

```
PS F:\code\github.com\godcrampy\college-notes\cns\lab-01> python .\brute.py
Enter file to be encrypted: encrypted.txt
Enter next key to try: 1
Text:
Hpwxa Qdcsgt
Wtaad Ldgas
123456

Enter 'y' to quit or 'n' for new key: n
Enter next key to try: 2
Text:
Iqxyb Redthu
Xubbe Mehbt
123456

Enter 'y' to quit or 'n' for new key: n
Enter next key to try: 12
Text:
Sahil Bondre
Hello World
123456

Enter 'y' to quit or 'n' for new key: y
Final Message:
Sahil Bondre
Hello World
123456

PS F:\code\github.com\godcrampy\college-notes\cns\lab-01>
```

Q4:

```
def file_to_str(filename):
    try:
        with open(filename, 'r') as file:
            return file.read()
    except:
        print("Error: File not found!")
        exit(1)



filename = input("Enter file to be encrypted: ")
message = file_to_str(filename)

actual_frequency = "ETAOINHSRLDWMUYGCFPBKVJXQZ"

message_frequency = []

for i in range(26):

    message_frequency.append([chr(ord("A") + i), 0])



for charecter in message:
    if charecter.isalpha():
        message_frequency[ord(charecter.upper()) - ord("A")][1] += 1
```

```
message_frequency.sort(key=lambda x: x[1], reverse=True)


result = ""


for charecter in message:
    if charecter.isalpha():
      # TODO: Change to binary search
      for i in range(26):
        if message_frequency[i][0] == charecter.upper():
          result += actual_frequency[i]
          break
    else:
      result += charecter


print("Decrypted Message:\n")
print(result.lower())
```

encrypted text:

```
Ftq baxuoqymz az ftq nqmf yahqp gb ftq mhqzgq uybdqeeuhqxk. Ftq
uybdqeeuhqzqee ime tmnufgmx mzp zaf rad etai, rad ebqofmfade iqdq rqi.
Ftq fuyq ime nmdqxk 10 a'oxaow mf zustf, ngf otuxxk sgefe ar iuzp iuft m
fmefq ar dmuz uz ftqy tmp iqxx zust pqbqabxqp ftq efdqqfe.
Fdkuzs paade me tq iqzf, fiudxuzs tue oxgn iuft ymzk uzfduomfq mzp
mdfrgx yahqyqzfe, fgdzuzs zai mzp ftqz fa omef tue imfotrgx qkq mpaiz
ftq bmouruo ftadagstrmdq, ftq arruoqd, iuft tue efmximdf rady mzp exustf
eimssqd, ympq m ruzq buofgdq ar m sgmdpumz ar ftq bqmoq. Ftq huouzufk
ime azq ftmf wqbf qmdxk tagde. Zai mzp ftqz kag yustf eqq ftq xustfe ar
m ousmd efadq ad ar mz mxx-zustf xgzot oagzfqd; ngf ftq ymvadufk ar ftq
paade nqxazsqp fa ngeuzqee bxmoqe ftmf tmp xazs euzoq nqqz oxaeqp.
Itqz mnagf yupimk ar m oqdfmuz nxaow ftq baxuoqymz egppqzxk exaiqp tue
imxw. Uz ftq paadimk ar m pmdwqzqp tmdpimdq efadq m ymz xqmzqp, iuft mz
gzxustfqp ousmd uz tue yagft. Me ftq baxuoqymz imxwqp gb fa tuy ftq ymz
ebawq gb cguowxk.
"Uf'e mxx dustf, arruoqd," tq emup, dqmeegduzsxk. "U'y vgef imufuzs rad
```

m rduqzp. Uf'e mz mbbauzfyqzf ympq fiqzfk kqmde msa. Eagzpe m xuffxq rgzzk fa kag, paqez'f uf? Iqxx, U'xx qjbxmuz ur kag'p xuwq fa ymwq oqdfmuz uf'e mxx efdmustf. Mnagf ftmf xazs msa ftqdq geqp fa nq m dqefmgdmzf itqdq ftue efadq efmzpe--'Nus Vaq' Ndmpk'e dqefmgdmzf."
"Gzfux ruhq kqmde msa," emup ftq baxuoqymz. "Uf ime fadz paiz ftqz."
Ftq ymz uz ftq paadimk efdgow m ymfot mzp xuf tue ousmd. Ftq xustf etaiqp m bmxq, ecgmdq-vmiqp rmoq iuft wqqz qkqe, mzp m xuffxq itufq eomd zqmd tue dustf qkqndai. Tue eomdrbuz ime m xmdsq pumyazp, appxk eqf.
"Fiqzfk kqmde msa fa-zustf," emup ftq ymz, "U puzqp tqdq mf 'Nus Vaq' Ndmpk'e iuft Vuyyk Iqxxe, yk nqef otgy, mzp ftq ruzqef otmb uz ftq iadxp. Tq mzp U iqdq dmueqp tqdq uz Zqi Kadw, vgef xuwq fia ndaftqde, fasqftqd. U ime qustfqqz mzp Vuyyk ime fiqzfk. Ftq zqjf yadzuzs U ime fa efmdf rad ftq Iqef fa ymwq yk radfgzq. Kag oagxpz'f tmhq pdmssqp Vuyyk agf ar Zqi Kadw; tq ftagstf uf ime ftq azxk bxmoq az qmdft. Iqxx, iq msdqqp ftmf zustf ftmf iq iagxp yqqf tqdq msmuz qjmofxk fiqzfk kqmde rday ftmf pmfq mzp fuyq, za ymffqd itmf agd oazpufuaze yustf nq ad rday itmf puefmzoq iq yustf tmhq fa oayq. Iq rusgdqp ftmf uz fiqzfk kqmde qmot ar ge agstf fa tmhq agd pqefuzk iadwqp agf mzp agd radfgzqe ympq, itmfqhqd ftqk iqdq sauzs fa nq."
"Uf eagzpe bdqffk uzfqdqefuzs," emup ftq baxuoqymz. "Dmftqd m xazs fuyq nqfiqqz yqqfe, ftagst, uf eqqye fa yq. Tmhqz'f kag tqmdp rday kagd rduqzp euzoq kag xqrf?"
"Iqxx, kqe, rad m fuyq iq oaddqebazpqp," emup ftq aftqd. "Ngf mrfqd m kqmd ad fia iq xaef fdmow ar qmot aftqd. Kag eqq, ftq Iqef ue m bdqffk nus bdabaeufuaz, mzp U wqbf tgefxuzs mdagzp ahqd uf bdqffk xuhqxk. Ngf U wzai Vuyyk iuxx yqqf yq tqdq ur tq'e mxuhq, rad tq mximke ime ftq fdgqef, efmzotqef axp otmb uz ftq iadxp. Tq'xx zqhqd radsqf. U omyq m ftagemzp yuxqe fa efmzp uz ftue paad fa-zustf, mzp uf'e iadft uf ur yk axp bmdfzqd fgdze gb."
Ftq imufuzs ymz bgxxqp agf m tmzpeayq imfot, ftq xupe ar uf eqf iuft eymxx pumyazpe.
"Ftdqq yuzgfqe fa fqz," tq mzzagzoqp. "Uf ime qjmofxk fqz a'oxaow itqz iq bmdfqp tqdq mf ftq dqefmgdmzf paad."
"Pup bdqffk iqxx agf Iqef, pupz'f kag?" mewqp ftq baxuoqymz.
"Kag nqf! U tabq Vuyyk tme pazq tmxr me iqxx. Tq ime m wuzp ar bxappqd, ftagst, saap rqxxai me tq ime. U'hq tmp fa oaybqfq iuft eayq ar ftq etmdbqef iufe sauzs fa sqf yk buxq. M ymz sqfe uz m sdaahq uz Zqi Kadw. Uf fmwqe ftq Iqef fa bgf m dmlad-qpsq az tuy."
Ftq baxuoqymz fiudxqp tue oxgn mzp faaw m efqb ad fia.
"U'xx nq az yk imk. Tabq kagd rduqzp oayqe mdagzp mxx dustf. Sauzs fa omxx fuyq az tuy etmdb?"
"U etagxp emk zaf!" emup ftq aftqd. "U'xx suhq tuy tmxr mz tagd mf xqmef. Ur Vuyyk ue mxuhq az qmdft tq'xx nq tqdq nk ftmf fuyq. Ea xazs, arruoqd."
"Saap-zustf, eud," emup ftq baxuoqymz, bmeeuzs az mxazs tue nqmf, fdkuzs paade me tq iqzf.

```
Ftqdq ime zai m ruzq, oaxp pdullxq rmxxuzs, mzp ftq iuzp tmp dueqz rday
ufe gzoqdfmuz bgrre uzfa m efqmpk nxai. Ftq rqi raaf bmeeqzsqde mefud uz
ftmf cgmdfqd tgdduqp pueymxxk mzp euxqzfxk mxazs iuft oamf oaxxmde
```

Decrypted Text:

```
PS F:\code\github.com\godcrampy\college-notes\cns\lab-01> python .\frequency.py
Enter file to be encrypted: frequency.txt
Decrypted Message:

the policeman on the beat moved up the avenue impressively. the impressiveness was habitual and not
for show, for spectators were few. the time was barely 10 o'clock at night, but chilly gusts of wind
 with a taste of rain in them had well nigh depeopled the streets.
trying doors as he went, twirling his club with many intricate and artful movements, turning now and
 then to cast his watchful eye adown the pacific thoroughfare, the officer, with his stalwart form a
nd slight swagger, made a fine picture of a guardian of the peace. the vicinity was one that kept ea
rly hours. now and then you might see the lights of a cigar store or of an all-night lunch counter;
but the majority of the doors belonged to business places that had long since been closed.
when about midway of a certain block the policeman suddenly slowed his walk. in the doorway of a dar
kened hardware store a man leaned, with an unlighted cigar in his mouth. as the policeman walked up
to him the man spoke up quickly.
"it's all right, officer," he said, reassuringly. "i'm just waiting for a friend. it's an appointmen
t made twenty years ago. sounds a little funny to you, doesn't it? well, i'll explain if you'd like
to make certain it's all straight. about that long ago there used to be a restaurant where this stor
e stands--'big joe' brady's restaurant."
"until five years ago," said the policeman. "it was torn down then."
the man in the doorway struck a match and lit his cigar. the light showed a pale, square-jawed face
with keen eyes, and a little white scar near his right eyebrow. his scarfpin was a large diamond, od
dly set.
"twenty years ago to-night," said the man, "i dined here at 'big joe' brady's with jimmy wells, my b
est chum, and the finest chap in the world. he and i were raised here in new york, just like two bro
thers, together. i was eighteen and jimmy was twenty. the next morning i was to start for the west t
o make my fortune. you couldn't have dragged jimmy out of new york; he thought it was the only place
 on earth. well, we agreed that night that we would meet here again exactly twenty years from that d
ate and time, no matter what our conditions might be or from what distance we might have to come. we
 figured that in twenty years each of us ought to have our destiny worked out and our fortunes made,
 whatever they were going to be."
"it sounds pretty interesting," said the policeman. "rather a long time between meets, though, it se
ems to me. haven't you heard from your friend since you left?"
"well, yes, for a time we corresponded," said the other. "but after a year or two we lost track of e
ach other. you see, the west is a pretty big proposition, and i kept hustling around over it pretty
lively. but i know jimmy will meet me here if he's alive, for he always was the truest, stanchest ol
d chap in the world. he'll never forget. i came a thousand miles to stand in this door to-night, and
 it's worth it if my old partner turns up."
the waiting man pulled out a handsome watch, the lids of it set with small diamonds.
"three minutes to ten," he announced. "it was exactly ten o'clock when we parted here at the restaur
ant door."
"did pretty well out west, didn't you?" asked the policeman.
"you bet! i hope jimmy has done half as well. he was a kind of plodder, though, good fellow as he wa
s. i've had to compete with some of the sharpest wits going to get my pile. a man gets in a groove i
n new york. it takes the west to put a razor-edge on him."
the policeman twirled his club and took a step or two.
"i'll be on my way. hope your friend comes around all right. going to call time on him sharp?"
```