Caesar Cipher.

Source Code.

Name – Rusheel Raj Panakadan

1. Implement the Caesar cipher in C, Java, or Python, as a program that reads a Latin string from standard input and encrypts each low case a-z character encountered, returning the result as uppercase. All other characters are left unchanged.

***Source Code:***

#!/usr/bin/python3.10

import sys

text = sys.argv[1]

def encrypt(text,s):

result = ""

#Traverse the plain text

for i in range(len(text)):

char = text[i]

##Encrypt lowercase characters in plain text

#Checking for white spaces!

if ord(char) == 32:

result = result + " "

else:

result += chr((ord(char) + s - 97) % 26 + 97)

return result

#Check the above function

s = 3

print ("Plain text: " + text)

print ("Shift pattern: "+ str(s))

print ("Cipher: " + encrypt(text,s).upper())

2. Change your code at task 1 to obtain a shift cipher with key passed as command line argument.

***Source Code:***

#!/usr/bin/python3.10

import sys

text = sys.argv[1]

s = int (sys.argv[2])

def encrypt(text,s):

result = ""

#Traverse the plain text

for i in range(len(text)):

char = text[i]

#Encrypt lowercase characters in plain text

# Checking for white spaces

if ord(char)==32:

result = result + " "

else:

result += chr((ord(char) + s - 97) % 26 + 97)

return result

#Check the above function

print ("Plain text: " + text)

print ("Shift pattern: "+ str(s))

print ("Cipher: " + encrypt(text,s).upper())

3. Implementing a python program to decrypt the Caesar Cipher text.

***Source Code:***

#!/usr/bin/python3.10

import sys

text = sys.argv[1]

s = int (sys.argv[2])

if text.isupper():

text = text.lower()

def decrypt(text,s):

result = ""

#Traverse the plain text

for i in range(len(text)):

char = text[i]

#Encrypt lowercase characters in plain text

# Checking for white spaces

if ord(char)==32:

result = result + " "

else:

result += chr((ord(char) - s - 97) % 26 + 97)

return result

#Check the above function

print ("Cipher text: " + text)

print ("Shift pattern: "+ str(s))

print ("Plain text: " + decrypt(text,s).upper())

4. Break the ciphertexts by brute-forcing:

***Source Code:***

#!/usr/bin/python3.10

import sys

text = sys.argv[1]

#s = int (sys.argv[2])

if text.isupper():

text = text.lower()

def decrypt(text,s):

result = ""

#Traverse the plain text

for i in range(len(text)):

char = text[i]

#Encrypt lowercase characters in plain text

# Checking for white spaces

if ord(char)==32:

result = result + " "

else:

result += chr((ord(char) - s - 97) % 26 + 97)

print (result)

return result

#Check the above function

print ("Cipher text: " + text)

for s in range(1,27):

a=decrypt(text,s)

s=s+1