Vyshak S Bekal - 241059045 - MSISMPL

CRYPTOLOGY LAB 3 ASSIGNMENT

Vyshak S Bekal

241059045 M.E - Cyber Security MSIS, MAHE

1. **Write a python script to encrypt the string using Caesar cipher.**

* def caesar\_cipher\_encrypt(text, shift):

encrypted\_text = ""

for char in text:

if char.isalpha(): # Encrypt only alphabets

shift\_char = 'A' if char.isupper() else 'a'

encrypted\_text += chr((ord(char) - ord(shift\_char) + shift) % 26 + ord(shift\_char))

else:

encrypted\_text += char

return encrypted\_text

# Example

text = "hello"

shift = 3

print(caesar\_cipher\_encrypt(text, shift)) # khoor

1. **Write a Python script to Modify the above script to shift cipher based on user choice.**

* def caesar\_cipher\_encrypt\_user\_input():

text = input("Enter the text to encrypt: ")

while True:

try:

shift = int(input("Enter the shift value (integer): "))

break

except ValueError:

print("Please enter a valid integer for shift value.")

encrypted\_text = ""

for char in text:

if char.isalpha(): # Encrypt only alphabets

shift\_char = 'A' if char.isupper() else 'a'

encrypted\_text += chr((ord(char) - ord(shift\_char) + shift) % 26 + ord(shift\_char))

else:

encrypted\_text += char

return encrypted\_text

print(caesar\_cipher\_encrypt\_user\_input())

1. **Write a Python script to convert cipher text into uppercase characters and split the cipher into group of 5 of characters.**

* def format\_cipher\_text(cipher\_text):

upper\_text = cipher\_text.upper()

grouped\_text = " ".join([upper\_text[i:i+5] for i in range(0, len(upper\_text), 5)])

return grouped\_text

cipher\_text = "khoorworld"

print(format\_cipher\_text(cipher\_text))

1. **Write a Python program to Find the histogram for each character.**

* from collections import Counter

def character\_histogram(text):

histogram = Counter(text)

for char, freq in histogram.items():

print(f"Character: {char} | Frequency: {freq}")

text = "hello"

character\_histogram(text)

1. **Write a Python script to read the contents from the file.**

* def read\_file(file\_path):

with open(file\_path, 'r') as file:

content = file.read()

return content

file\_path = 'example.txt'

print(read\_file(file\_path))

1. **Write a Python script to encrypt the contents from the file.**

* def encrypt\_file\_content(file\_path, shift):

content = read\_file(file\_path)

encrypted\_content = caesar\_cipher\_encrypt(content, shift)

return encrypted\_content

file\_path = 'example.txt'

shift = 3

print(encrypt\_file\_content(file\_path, shift))

1. **Do validation to the python program (2)**

**- not to accept special characters**

**- not to accept numeric values**

**- not to accept empty value**

**- accept only string**

**- string should be lowercase if not convert the case**

* def caesar\_cipher\_encrypt\_user\_input():

while True:

text = input("Enter the text to encrypt (lowercase only): ")

if not text.isalpha():

print("Please enter only alphabetic characters, no numbers or special characters.")

elif not text.islower():

print("The input contains uppercase letters, converting to lowercase.")

text = text.lower()

else:

break

while True:

try:

shift = int(input("Enter the shift value (integer): "))

break

except ValueError:

print("Please enter a valid integer for shift value.")

encrypted\_text = ""

for char in text:

shift\_char = 'a'

encrypted\_text += chr((ord(char) - ord(shift\_char) + shift) % 26 + ord(shift\_char))

return encrypted\_text

print(caesar\_cipher\_encrypt\_user\_input())

1. **Write a Python program to checks if two given strings are anagrams of each other. example:**

**mug, gum**

**cork, rock**

**note, tone**

* def anagrams(str1, str2):

return sorted(str1) == sorted(str2)

print(anagrams("mug", "gum"))

print(anagrams("cork", "rock"))

1. **Write a Python program to check the given string is palindrome or not.**

**Do not use built in functions.**

**Example:**

**MADAM**

**RACECAR**

**LEVEL**

**CIVIC**

* def is\_palindrome(text):

length = len(text)

for i in range(length // 2):

if text[i] != text[length - i - 1]:

return False

return True

print(is\_palindrome("MADAM"))

print(is\_palindrome("RACECAR"))

1. **Write a Python program to check if a substring is present in a given string.**

**Example: Understand – stand**

* def is\_substring(main\_string, sub\_string):

return sub\_string in main\_string

print(is\_substring("Understand", "stand"))

1. **Explore string module**

**import the string module in your python script.**

**print all the lowercase characters**

**print all the uppercase characters**

**print all the lowercase and uppercase characters**

**print all the digits**

**print all the punctuation symbols**

**count the total number of punctuation symbols**

* import string

def explore\_string\_module():

print("Lowercase Letters: ", string.ascii\_lowercase)

print("Uppercase Letters: ", string.ascii\_uppercase)

print("All Letters: ", string.ascii\_letters)

print("Digits: ", string.digits)

print("Punctuation Symbols: ", string.punctuation)

print("Total Number of Punctuation Symbols: ", len(string.punctuation))

explore\_string\_module()