

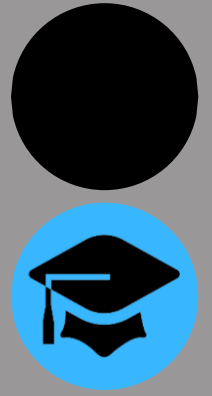


FORGE YOUR AMBITION

# HUNAR INTERN

WWW.HUNARINTERN.LIVE

LET'S GET STARTED



Name: Sriram Leburi

Date: 12/03/2024

## Task- 2(Medium)

### Task: Create a Encryption/Decryption Tool

#### Description:

This project is to develop a straightforward tool that performs encryption and decryption of messages or files. Using Python, I am creating a program that allows users to encode and decode information securely.

**Encryption(Encode):** Process of hiding file or normal plain text in cipher text.

Encryption is to maintain “Confidentiality”

**Decryption(Decode):** Process of converting cipher text into normal plain text.

#### Steps to Follow:

##### 1. Write Code:

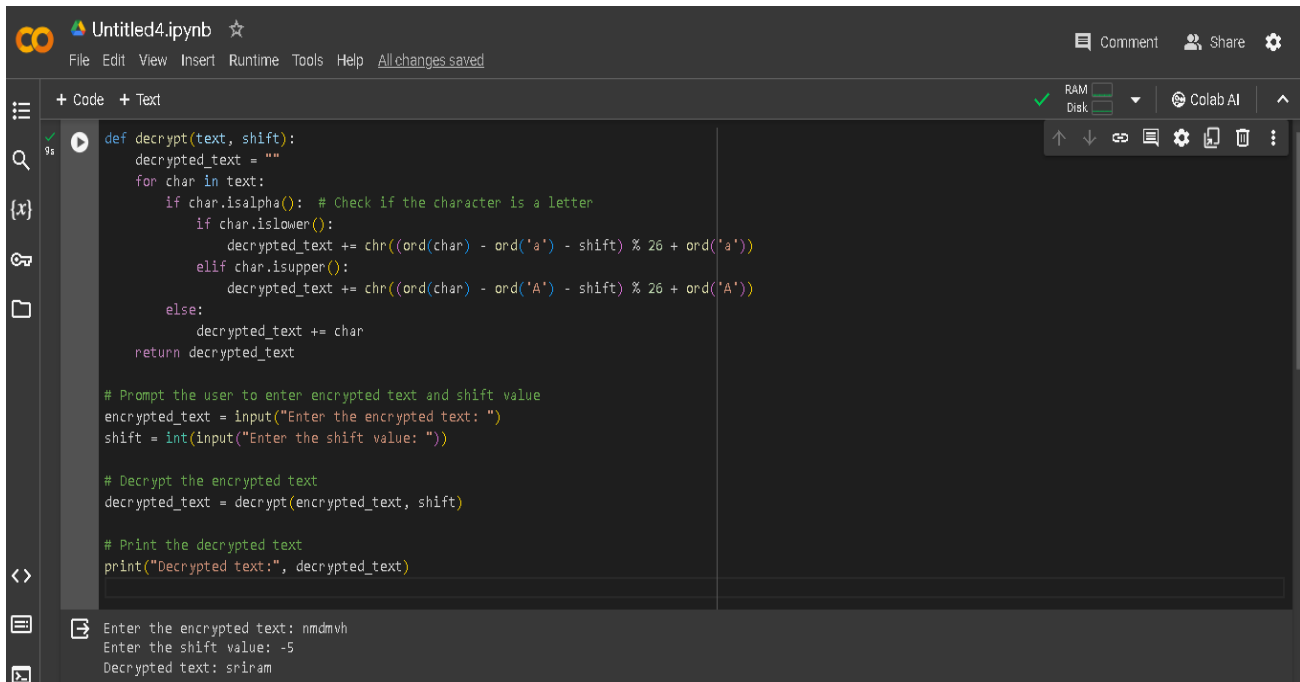
```
def decrypt(text, shift):
    decrypted_text = ""
    for char in text:
        if char.isalpha(): # Check if the character is a letter
            if char.islower():
                decrypted_text += chr((ord(char) - ord('a') - shift) % 26 + ord('a'))
            elif char.isupper():
                decrypted_text += chr((ord(char) - ord('A') - shift) % 26 + ord('A'))
        else:
            decrypted_text += char
    return decrypted_text
```

```
# Prompt the user to enter encrypted text and shift value
encrypted_text = input("Enter the encrypted text: ")
shift = int(input("Enter the shift value: "))

# Decrypt the encrypted text
decrypted_text = decrypt(encrypted_text, shift)

# Print the decrypted text
print("Decrypted text:", decrypted_text)
```

## 2. Display Results:



The screenshot shows a Jupyter Notebook titled 'Untitled4.ipynb'. The code cell contains a function `decrypt(text, shift)` that iterates through each character in the input text. If the character is a letter, it calculates its position in the alphabet (0-25), subtracts the shift value, and wraps around using modulo 26. For lowercase letters, it adds the result to the ASCII value of 'a'; for uppercase letters, it adds the result to the ASCII value of 'A'. Non-alphabetic characters are left unchanged. Below the function, the code prompts the user for encrypted text and shift value, then calls the `decrypt` function and prints the result. The output cell shows the user input 'nmnmvh' and shift value '-5', resulting in the decrypted text 'sriram'.

```
def decrypt(text, shift):
    decrypted_text = ""
    for char in text:
        if char.isalpha(): # Check if the character is a letter
            if char.islower():
                decrypted_text += chr((ord(char) - ord('a') - shift) % 26 + ord('a'))
            elif char.isupper():
                decrypted_text += chr((ord(char) - ord('A') - shift) % 26 + ord('A'))
            else:
                decrypted_text += char
    return decrypted_text

# Prompt the user to enter encrypted text and shift value
encrypted_text = input("Enter the encrypted text: ")
shift = int(input("Enter the shift value: "))

# Decrypt the encrypted text
decrypted_text = decrypt(encrypted_text, shift)

# Print the decrypted text
print("Decrypted text:", decrypted_text)
```

Enter the encrypted text: nmnmvh  
Enter the shift value: -5  
Decrypted text: sriram

## What I have Learnt:

- Gain practical experience in programming with Python.
- Understand the basics of encryption and decryption algorithms.
- Learn how to implement secure information handling.

## Conclusion:

This project is an opportunity to delve into the fundamentals of encryption and decryption while strengthening your Python coding skills. By the end of this task, I have created a functional tool that can securely encode and decode information.