|  |  |
| --- | --- |
| **Program specification and checklist**  Your code will be written to the following specification | |
|  | Done |
| Include an options menu for the user allowing them to carry out the following actions   * Add stored credentials (username, password and URL/resource) * View stored credentials * Exit the program | **√** |
| Return to the menu after each action has completed | **√** |
| Create a text file for credential storage if a text file does not already exist | **√** |
| Append new records to the text file without overwriting previous entries | **√** |
| Display the text file contents in a visually presentable way including spacing and headings | **√** |
| Handle any input from the user and carry out actions, without errors | **√** |
| Include embedded explanatory comments (#) to clarify the meaning of the code | **√** |
| Provide simple rot3 encryption on all written data and, decryption on read data | **√** |

Include an options menu for the user allowing them to carry out the following actions

* Add stored credentials (username, password and URL/resource)
* View stored credentials
* Exit the program

def main():

    while True:

        print("Welcome to Password Manager!")

        print("1. Add Credentials")

        print("2. View Credentials")

        print("3. Exit")

        choice = input("Enter your choice: ")

        if choice == "1":

            website = input("Enter website: ")

            username = input("Enter username: ")

            password = input("Enter password: ")

            addCredentials(website, username, password)

        elif choice == "2":

            website = input("Enter website: ")

            viewCredentials(website)

        elif choice == "3":

            print("Goodbye!")

            break

        else:

            print("Invalid choice. Please choose again.")

Return to the menu after each action has completed

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Create a text file for credential storage if a text file does not already exist

"""Create a filename"""

filename = "credentials.txt"

with open(filename, "r"):

    print("File already exists.")

with open(filename, "w") as file:

    print("File created for credential storage.")

Append new records to the text file without overwriting previous entries

"""Append new records"""

filename = "credentials.txt"

new\_record = "New record to append\n"

with open(filename, "a") as file:

    file.write(new\_record)

|  |
| --- |
| Append new records to the text file without overwriting previous entries |

Display the text file contents in a visually presentable way including spacing and headings

"""Text file Content display"""

def display\_credentials(filename):

    with open(filename, "r") as file:

        lines = file.readlines()

    print("Credentials:")

    print("=" \* 20)

    for line in lines:

        line = line.strip()

        if line:

            heading, value = line.split(":")

            print(f"{heading.strip()}: {value.strip()}")

        else:

            print()

    print("=" \* 20)

filename = "credentials.txt"

display\_credentials(filename)

Handle any input from the user and carry out actions, without errors

def add\_record():

    website = input("Enter website: ")

    username = input("Enter username: ")

    password = input("Enter password: ")

def main():

    while True:

        print("1. Add Record")

        print("2. View Records")

        print("3. Exit")

        choice = input("Enter your choice: ")

        if choice == "1":

            add\_record()

        elif choice == "2":

            view\_records()

        elif choice == "3":

            print("Goodbye!")

            break

        else:

            print("Invalid choice. Please choose again.")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Include embedded explanatory comments (#) to clarify the meaning of the code

def add\_record():

    website = input("Enter website: ")

    username = input("Enter username: ")

    password = input("Enter password: ")

def view\_records():

    pass

def main():

    while True:

        print("1. Add Record")

        print("2. View Records")

        print("3. Exit")

        choice = input("Enter your choice: ")

        if choice == "1":

            add\_record()  # Call the function to add a new record

        elif choice == "2":

            view\_records()  # Call the function to view records

        elif choice == "3":

            print("Goodbye!")

            break

        else:

            print("Invalid choice. Please choose again.")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Provide simple rot3 encryption on all written data and, decryption on read data

def rot3\_encrypt(text):

    encrypted\_text = ""

    for char in text:

        if char.isalpha():

            offset = 3

            if char.islower():

                encrypted\_char = chr((ord(char) - ord('a' if char.islower() else 'A' if char.isupper() else 'a') + offset) % 26 + ord('a'))

            else:

                encrypted\_char = chr((ord(char) - ord('a' if char.islower() else 'A' if char.isupper() else 'a') + offset) % 26 + ord('A'))

        else:

            encrypted\_char = char

        encrypted\_text += encrypted\_char

    return encrypted\_text

plaintext = "Hello, World!"

encrypted\_text = rot3\_encrypt(plaintext)

print("Encrypted:", encrypted\_text)

|  |  |
| --- | --- |
| **Coding Checklist**  Your code must include following criteria. *The checklist is given to* ***ensure*** *all aspects of the assessment are covered. Your assessor will confirm that your code includes the criteria when marking your submission* | |
| Criteria | Tick when complete |
| At least one of each of the following:   * Global variable * Local variable | PasswordDatabase, website, username, password, choice, filename, etc |
| At least two library functions  *(internal or external)* | **Print, input, open, readline, write, strip, split, isalpha, islower, isupper, etc** |
| At least one self-created function | addCredentials(),view\_record(),viewCredentials(),displayCredentials(), add\_record(), etc |
| Clarifying comments | add\_record()  # Call the function to add a new record   view\_records()  # Call the function to view records |
| A data structure  (i.e list, dictionary, tuple or set) | passwordDatabase{} |
| Manipulation of strings |  |

*Website Password Manager : supervisor sign-off*

*Add Password:*

*The part of your code that does this is the addCredentials function. This function takes three parameters: website, username, and password. It checks if the website already exists in the password database, which is a global dictionary variable. If not, it adds a new key-value pair to the dictionary, where the key is the website and the value is another dictionary with the username and password as keys and values. It then prints a message to confirm that the credentials have been added. If the website already exists in the password database, it prints a message to inform the user that the website is already in the database.*

*Here is the code for the addCredentials function:*

*def addCredentials(website, username, password):*

*if website not in passwordDatabase:*

*passwordDatabase[website] = {"username": username, "password": password}*

*print("Credentials added for", website)*

*else:*

*print("Website already exists in the database.")*

*Get Password:*

*The part of your code that does this is the viewCredentials function. This function takes one parameter: website. It checks if the website exists in the password database, which is a global dictionary variable. If yes, it gets the credentials for that website from the dictionary and prints them to the user. It prints the website name, username, and password. If no, it prints a message to inform the user that the website is not found in the database.*

*Here is the code for the viewCredentials function:*

*def viewCredentials(website):*

*if website in passwordDatabase:*

*credentials = passwordDatabase[website]*

*print("Website:", website)*

*print("Username:", credentials["username"])*

*print("Password:", credentials["password"])*

*else:*

*print("Website not found in the database.")*

*Exit:*

elif choice == "3":

            print("Goodbye!")

            break