**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**



**Лабораторна робота № 2**

з дисципліни

“Захист інформації”

**Виконав:**

студент групи ШІ-32

Лещишин Маркіян

**Викладач:**

Данчак О. І.

Львів – 2024 р.

**Тема.** Створення програмного засобу для забезпечення цілісності інформації.

**Мета.** Ознайомитись з методами криптографічного забезпечення цілісності інформації, навчитись створювати програмні засоби для забезпечення цілісності інформації з використанням алгоритмів хешування.

**Завдання:**

Створити програмну реалізацію алгоритму хешування MD5. Тестування створеної програми провести з використання наступних тестових значень хешу (згідно RFC 1321):

H() = D41D8CD98F00B204E9800998ECF8427E

H(a) = 0CC175B9C0F1B6A831C399E269772661

H(abc) = 900150983CD24FB0D6963F7D28E17F72

H(message digest) = F96B697D7CB7938D525A2F31AAF161D0

H(abcdefghijklmnopqrstuvwxyz) = C3FCD3D76192E4007DFB496CCA67E13B

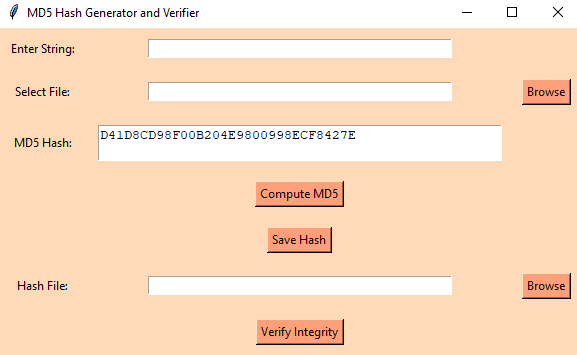
H(ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789) = D174AB98D277D9F5A5611C2C9F419D9F

H(12345678901234567890123456789012345678901234567890123456789012345678901234567890) = 57EDF4A22BE3C955AC49DA2E2107B67A

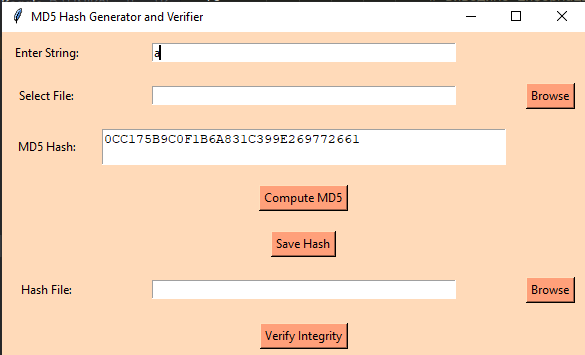
Програмна реалізація повинна виводити значення хешу як для рядка, заданого в полі вводу, так і для файлу (врахувати, що файли можуть бути великого розміру). Результат роботи програми повинен відображатись на екрані з можливістю наступного запису в файл. Крім того програма повинна мати можливість перевірити цілісність будь-якого файлу за наявним файлом з MD5 хешем, записаним у шістнадцятковому форматі. У звіті навести протокол тестування і роботи програми та зробити висновки.

**Результати виконання програми:**

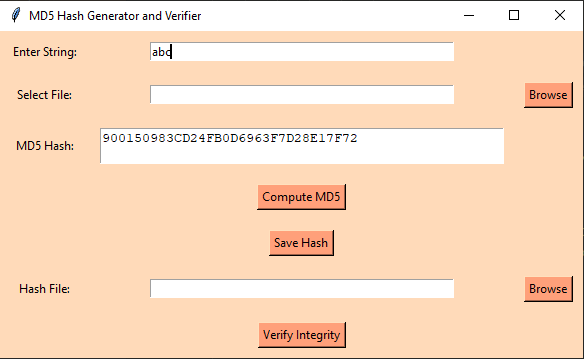
Результат для пустого рядка збігається з даними у методичці:

****

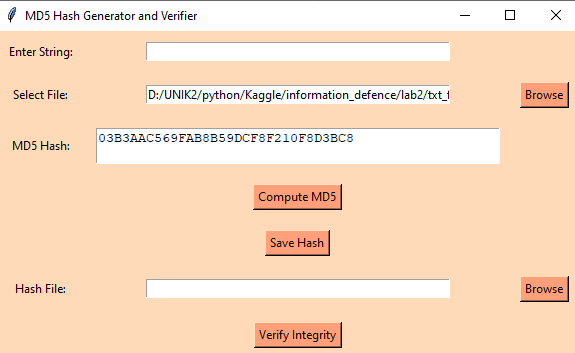
Результат для ‘a’ збігається з даними у методичці:

****

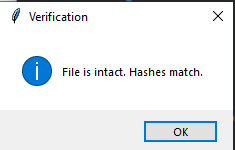
Результат для ‘abc’ збігається з даними у методичці:

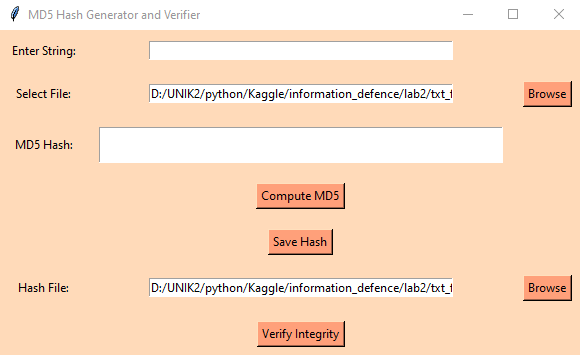
****

Результат хешування для файлу розміром 100MБ:

****

Результат перевірки цілісності файлу:





**Контрольні запитання**

1. **Що таке хеш функція?**

Хеш-функція — це математичний алгоритм, який перетворює вхідні дані (будь-якого розміру) у фіксовану довжину виходу, званого хеш-значенням або хешем. Вона забезпечує унікальне представлення для кожного набору даних, де навіть невеликі зміни у вхідних даних призводять до значних змін у вихідному хеші.

1. **Області використання хеш функцій.**

Перевірка цілісності даних: Використовується для перевірки, чи були дані змінені. Наприклад, хеші файлів можуть бути порівняні, щоб переконатися, що файл не був пошкоджений або змінений.

Зберігання паролів: Замість зберігання паролів у відкритому вигляді, системи зберігають їх хеші, що забезпечує додатковий рівень безпеки.

Цифрові підписи: Хеш-функції використовуються для створення цифрових підписів, що забезпечують автентичність документа.

Віртуальні валюти: Хеш-функції використовуються в блокчейнах для підтвердження транзакцій.

1. **Який сервіс безпеки забезпечує хеш функція?**

Хеш-функції забезпечують цілісність даних, підтверджуючи, що дані не були змінені з моменту їх хешування. Вони також допомагають у забезпеченні автентичності даних і є важливими для криптографії.

1. **Чи використання хеш функцій без інших криптографічних алгоритмів забезпечує захист від дій третьої сторони?**

Ні, хеш-функції самі по собі не забезпечують захист від дій третьої сторони, оскільки вони не шифрують дані. Зловмисник може все ще виконати атаки, такі як підбор паролів, якщо доступне хеш-значення. Використання додаткових криптографічних алгоритмів, таких як шифрування, підвищує безпеку.

1. **Назвіть вимоги до криптографічних хеш функцій.**

Односторонність: Хеш-функція повинна бути такою, що з виходу неможливо відновити вихідні дані.

Стійкість до колізій: Не повинно бути можливості знайти два різних вхідних даних, які дають однакове хеш-значення.

Стійкість до попереднього обчислення: Неможливо знайти вхідне значення за заданим хешем.

Невизначуваність: Зміна навіть одного біта вхідних даних повинна радикально змінити вихідне хеш-значення.

1. **Чому хеш-функція є односторонньою?**

Хеш-функції проектуються так, що вихід (хеш) не може бути використаний для відновлення вхідних даних. Це досягається через складність обчислень, які ускладнюють зворотний процес, і властивості, що забезпечують великі простори вхідних даних для кожного можливого хешу.

1. **Що таке слабка опірність колізіям?**

Слабка опірність колізіям означає, що для певної хеш-функції можливо знайти два різних вхідних даних, які дають однаковий хеш. Це створює вразливість для атак, оскільки зловмисник може створити два різні документи з однаковими хеш-значеннями.

1. **Основні області використання алгоритму MD5.**

Перевірка цілісності даних: Часто використовується для генерації хешів файлів для перевірки їх цілісності.

Зберігання паролів: Хоча це не рекомендується через вразливість, MD5 використовувався для хешування паролів.

Ідентифікація даних: Використовується в системах, які потребують унікальної ідентифікації даних, таких як бази даних.

Цифрові підписи: Використовувався для створення цифрових підписів, хоча зараз вважається ненадійним.

1. **Яка довжина хеш-значення алгоритму MD5?**

Довжина хеш-значення MD5 становить 128 біт (16 байт), що зазвичай представляється у вигляді 32-значного шестигранного рядка.

**10. Яка довжина блоку, що обробляється алгоритмом MD5?**

Довжина блоку MD5 становить 512 біт (64 байти). Алгоритм обробляє дані в таких блоках під час обчислення хеш-значення.

**Висновки**:

MD5 (Message Digest Algorithm 5) — це популярний алгоритм хешування, який широко використовується для генерації 128-бітних хешів, що представляють собою фіксовану довжину незалежно від розміру вхідних даних. Мій код реалізує можливість обчислювати MD5-хеші як для рядків, так і для файлів, дозволяючи зручно перевіряти цілісність даних або ідентифікувати їх. Для цього використовується графічний інтерфейс, що дозволяє користувачу вводити дані або вибирати файл для обчислення його хешу, після чого результат можна зберегти або використати для перевірки цілісності.

Перевага MD5 полягає в його швидкості та простоті використання, що робить його ідеальним для задач, пов'язаних з перевіркою цілісності великих обсягів даних. Зокрема, для файлів, які передаються через інтернет або зберігаються на дисках, MD5 дає змогу швидко перевірити, чи не було вони змінені. У коді також реалізовано дерево-хешування для паралельної обробки файлів великих розмірів, що значно пришвидшує обчислення.

Однак MD5 має серйозні недоліки, коли мова йде про безпеку. Через виявлені вразливості до колізій (ситуації, коли два різні набори даних мають однаковий хеш), алгоритм більше не рекомендовано використовувати для криптографічно важливих задач, таких як хешування паролів або цифрових підписів. У таких випадках краще використовувати сучасніші та безпечніші алгоритми, як-от SHA-256 або bcrypt.

Таким чином, MD5 залишається корисним інструментом для перевірки цілісності файлів або ідентифікації дублікатів даних. Він підходить для ситуацій, де важливі швидкість і ефективність, але не для захисту конфіденційних або критичних даних.

Додаток А. Код реалізації програми: