Для виконання даної лабораторної роботи були використані наступні списки з ТОП-ами паролів:

- ТОП-100: https://www.forbes.com/sites/daveywinder/2019/12/14/ranked-the-worlds-100-worst-passwords/?sh=3b07df6469b4
- ТОП-100000: https://github.com/danielmiessler/SecLists/blob/master/Passwords/Common-Credentials/10-million-password-list-top-100000.txt

Генерація паролів відбувається у класі `PasswordsGenerator.cs` . У конструктор класу передаються такі додаткові параметри:

- `top100Persentage` (10) – відсоток паролів, які треба генерувати зі списку ТОП-100;
- `top100000Persentage` (70) – відсоток паролів, які треба генерувати зі списку ТОП-100000;
- `randomPasswordsPersentage` (5) – відсоток паролів, які необхідно генерувати рандомно.

Паролі генеруються за допомогою методу `GeneratePasswords` наступними чином:

- `top100Persentage` відсоток паролів заповнюється зі списку ТОП-100;
- `top100000Persentage` відсоток паролів заповнюється зі списку ТОП-100000;
- `randomPasswordsPersentage` відсоток паролів генерується рандомно за допомогою методу
-
-
- `GenRandomPassword`, який створює пароль довжини `length` (переданої у вхідні параметри) із рядку допустимих символів = `allowable_chars =` `"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!:$_"`
- решта (100% - `top100Persentage` - `top100000Persentage` - `randomPasswordsPersentage`) відсоток паролів генерується на основі списку ТОП-100000 за допомогою правил, описаних у методі `GenHumanLikePassword`, `що використовує наступні правила`:
    1) Створює комбінацію з двох слів зі списку найпоширеніших англійських слів;
    2) Додає 5 цифр в рандомні місця паролю;
    3) `ReplaceNumbers` - замінює цифри на відповідну літеру з англійського алфавіту
    4) Додає від 1 до 10 цифр в кінець паролю;
    5) `ChangeCase` – замінює регістр літер у паролі таким чином, щоб верхній чергувався з нижнім (наприклад, “PaSsWoRd” або “pAsSwOrD”).

Кількість паролів `passwordsCount` , які потрібно згенерувати, передається у вхідні параметри методу `GeneratePasswords` .

Паролі хешуються за допомогою 3-ох алгоритмів:

1. MD5 (реалізація у класі `MD5H`);
2. Argon2id (`Argon2`).
3. Bcrypt(`Bcrypt`)

Класи MD5 та Argon2 містять у собі метод для створення «солі» `CreateSalt` , де «сіль» створюється за допомогою `System.Security.Cryptography.RNGCryptoServiceProvider.GetBytes` .

Дані класи містять метод для запису хешів паролі / паролів + солі в файл

Програма створює 3 набори паролів по 10000 записів у кожному, потім хешує ці набори наведеними алгоритмами хешування та записує результати у 3 csv файли:

1. https://github.com/ArturSavchuk/DataSecurity/blob/main/lab4/Records/MD5Hashes.csv – паролі захешовані алгоритмом MD5.
2. https://github.com/ArturSavchuk/DataSecurity/blob/main/lab4/Records/Argon2Hashes.csv - Argon2.
3. https://github.com/ArturSavchuk/DataSecurity/blob/main/lab4/Records/BCryptHashes.csv –BCrypt.