Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**ЛАБОРАТОРНА РОБОТА № 4**

з дисципліни «Системне програмування» на тему

«Макровизначення і макроси в MASM»

Виконав: Перевірив:

Студент 2 курсу ФІОТ-у доц. Павлов В.Г.

групи ІМ-33

Козарезов Кирил Олександрович

номер у списку групи - 10

Київ 2025

**Мета роботи:**

Вивчити технології створення і вживання макросів. Дослідження результатів роботи макрогенератору **MASM**.

**Порядок виконання роботи:**

1. Вивчити правила використання макросів в програмах [1].

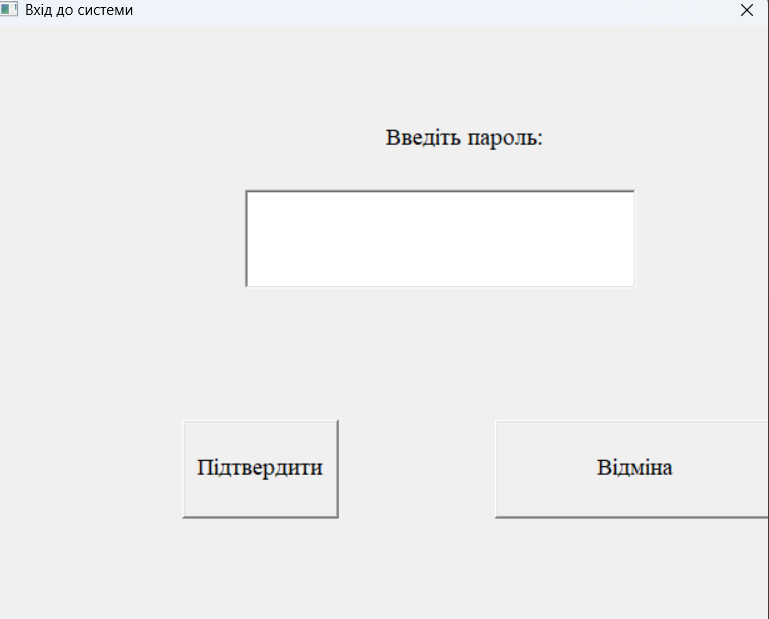
2. Переробити програму, підготовлену в лабораторній роботі 3 так, щоб виведення кожного виду інформації (дата народження, номер залікової книжки і т. п.) виконувалося у вигляді багатократного звернення до макросу (макрос 1), в якому у віконному інтерфейсі здійснювалося б виведення тексту, який передається в макрос як параметр.

3. Оформити також у вигляді окремих макросів фрагменти програми, де здійснюється шифрування введеного рядка символів (макрос 2), а також порівняння її з хеш-кодом оригіналу пароля, що зберігається у програмі (макрос 3). Макровизначення для всіх макросів розмістити в тому ж файлі, що і програма. У всіх макросах обов'язково використовувати звичайні і приховані коментарі, а в останньому макросі також використовувати механізм оголошення локальних міток.

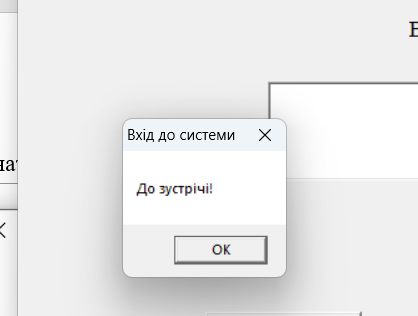
4. Виконати компіляцію і компоновку файлу програми.

5. Перевірити роботу програми шляхом введення як правильного, так і невірного паролів.

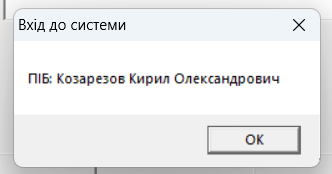
Початкове вікно програми:

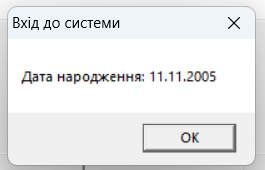


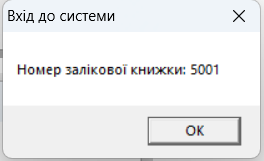
Вікно, яке з’являється при натисканні кнопки «Відміна»:



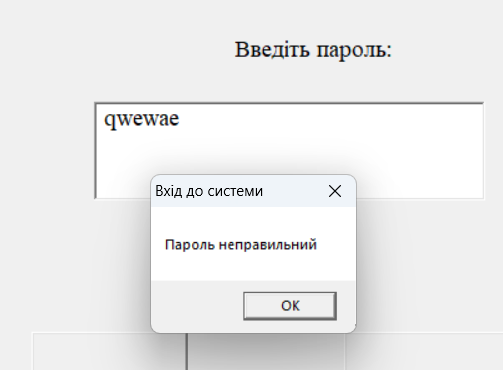
Вікна, які почергово з’являються після введення правильного паролю:







Вікно, яке з’являється після введення неправильного паролю:



**Використання макросів**

Під час виконання завдання було використано 3 макроси :

1. KozarezovKyryl\_ShowMsg — Макрос ,що виводить текст у вікні повідомлення. Параметри: заголовок вікна, текст повідомлення.

2.KozarezovKyryl\_EncryptString— Макрос ,що шифрує рядок методом XOR. Параметри: вхідний рядок, ключ, буфер для результату, довжина.

3.KozarezovKyryl\_CompareEncrypted— Макрос, що порівнює два зашифрованих буфери. Параметри: перший буфер, другий буфер, довжина.

**Шифрування паролю функцією XOR**

Зашифровуємо пароль «**Bocchi**» за ключем «**!@!@!!**»:

B (01000010) XOR ! (00100001) = c (01100011)

o (01101111) XOR @ (01000000) = / (00101111)

c (01100011) XOR ! (00100001) = B (01000010)

c (01100011) XOR @ (01000000) = # (00100011)

h (01101000) XOR ! (00100001) = I (01001001)

i (01101001) XOR ! (00100001) = H (01001000)

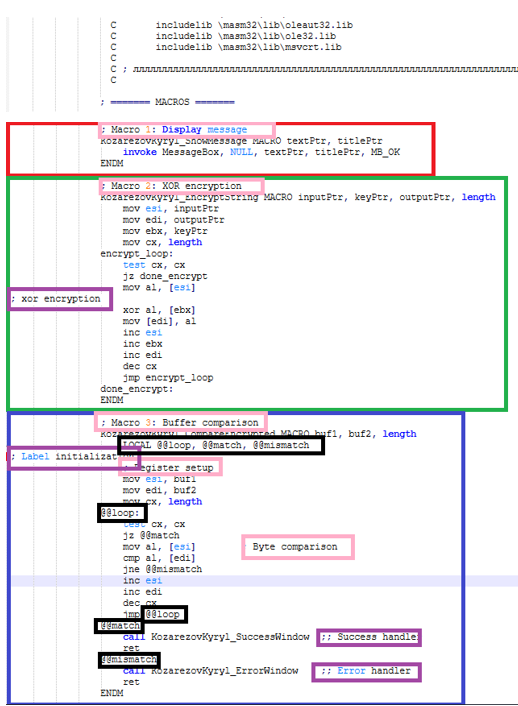
Після шифрування ми отримали зашифрований пароль c/B#IH

6. Отримати розширений лістинг програми за допомогою опції /Fl компілятора ML.

7. Провести дослідження отриманого лістингу: визначити, яким чином компілятор виконав трансляцію кожній з команд макросів і помітити їх в тексті розширеного лістингу кольоровими олівцями або кольоровими фломастерами.

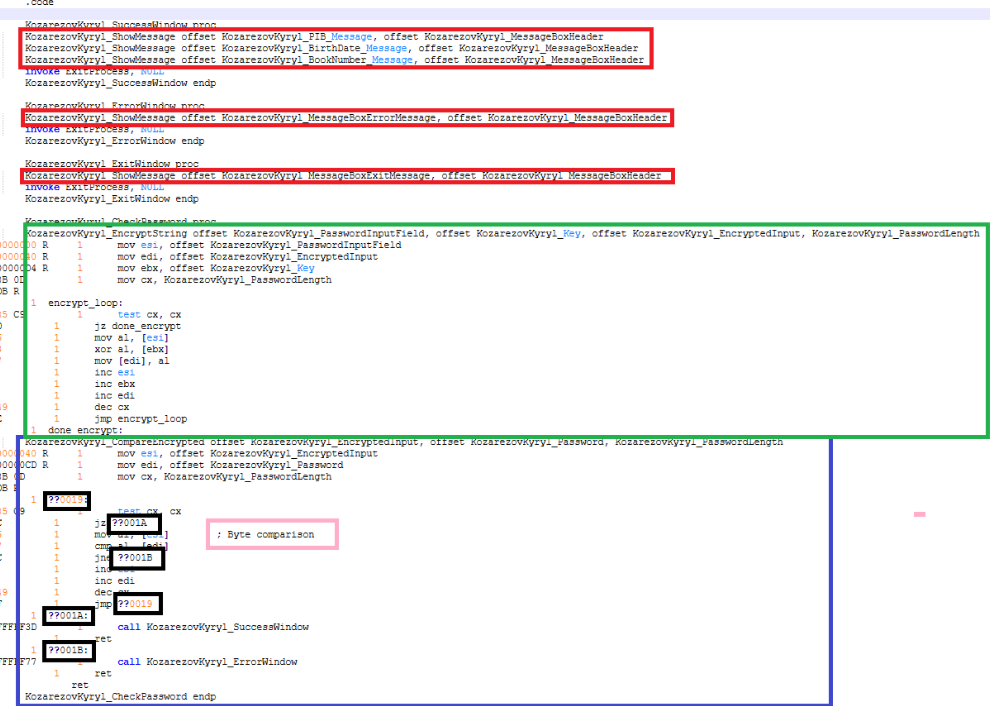
**Дослідження лістингу**

Макровизначення



З скріншотів можемо бачити, що макровизначення у варіанті, де макроси були оголошені разом з іншою логікою програми, знаходиться одразу після оголошення бібліотек **,** так само, як і в самій програмі. Перший макрос, відповідальний за вивід інформації, позначений червоним кольором, другий, відповідальний за шифрування паролю – позначений зеленим, а третій, відповідальний за порівняння паролів – синім. В усіх трьох програмах були використані звичайні та приховані коментарі , що позначенні рожевим та фіолетовим кольором відповідно. Також у третьому макросі був використаний механізм локальних міток , що позначенні на зображенні чорним кольором. З цього випливає , що при макровизначенні чітко видно звичайні та приховані коментарі, і назви локальних міток.

Трансляція

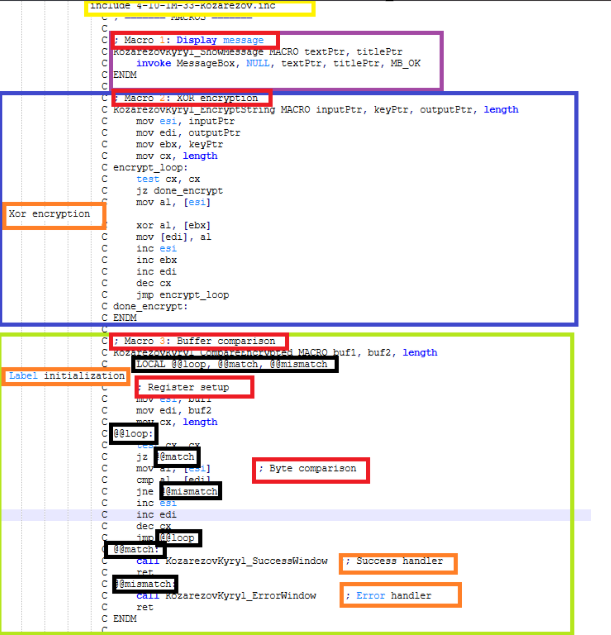


На зображенні вище представлено трансляцію трьох макросів: для порівняння введеного рядка з хеш-кодом оригінального пароля (позначено синім), для виведення вікна з повідомленням (червоним), а також для шифрування введеного рядка (зеленим). Зверніть увагу, що в трансляції макросу порівняння паролів відсутня оригінальна назва мітки — замість неї відображається її машинний код (виділено чорним). Крім того, після трансляції немає прихованих коментарів — лише звичайні (крім тих, що відповідальні за призначення макросу). Важливо зазначити, що виклики функцій invoke у макросах для порівняння паролів і виведення повідомлення є прихованими.

8. Повторити дослідження, які виконані в п. 4 – 7 для другої версії програми, де всі перераховані в п. 2 і 3 макровизначення для макросів повинні розташовуватися в окремому файлі.

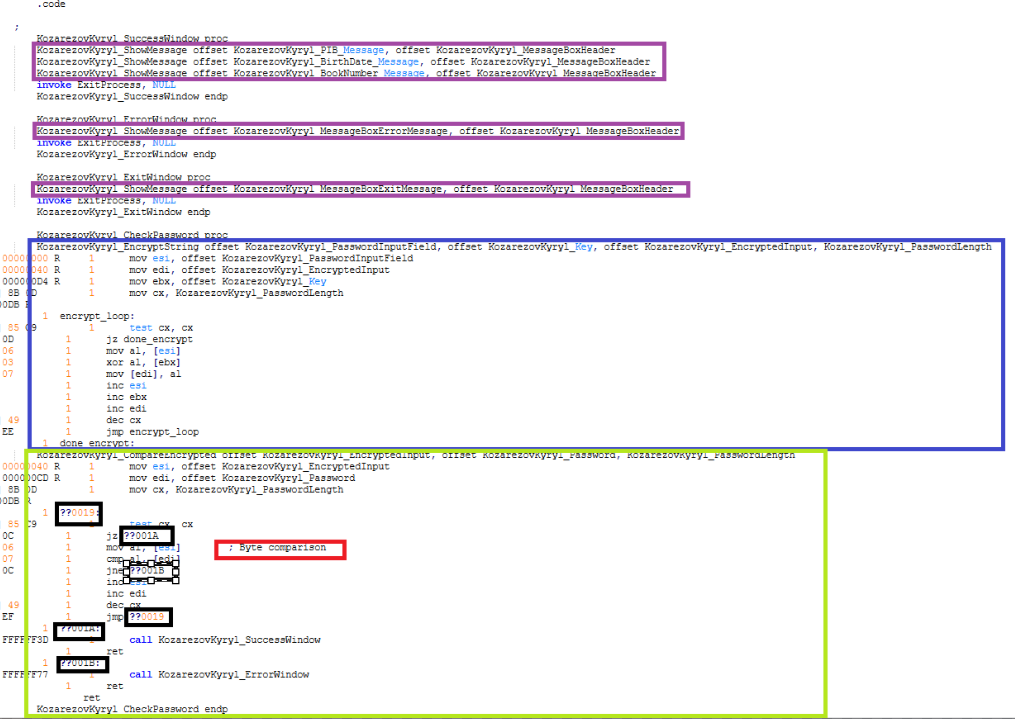
**Дослідження лістингу (макровизначення в окремому файлі)**

Макровизначення



У лістингу всі три макровизначення можна побачити одразу після їх імпорту, який виділений жовтим кольором. Це макроси для: виведення вікна з повідомленням (позначено фіолетовим), шифрування введеного рядка (синій колір) і порівняння цього рядка з хеш-кодом оригінального пароля (світло-зелений). Подібно до першої програми, у макровизначеннях присутні як приховані коментарі (виділені помаранчевим), так і звичайні (червоний колір). У третьому макросі додатково використано механізм локальних міток, які на зображенні позначено чорним

Трансляція



На зображенні вище представлено трансляцію трьох макросів: для виведення вікна з повідомленням (позначено фіолетовим), для шифрування введеного рядка символів (синім), а також для порівняння цього рядка з хеш-кодом оригінального пароля (світло-зеленим). Зверніть увагу, що в трансляції макросу для порівняння паролів відсутня оригінальна назва мітки — замість неї відображається її машинний код (виділено чорним). Крім того, після трансляції наявні як звичайні коментарі (позначені червоним), так і приховані (виділені помаранчевим). Важливо зазначити, що виклики функцій invoke у макросах для порівняння паролів та виведення повідомлення залишаються прихованими

9. Порівняти тексти розширених лістингів для обох варіантів розміщення макровизначень, знайти схожість і відмінності. Відобразити їх в звіті по лабораторній роботі.

Порівнявши розширені лістинги обох програм, вдалося чітко виокремити як спільні, так і відмінні елементи. Серед відмінностей — розташування макровизначень: у першому варіанті, де макроси були частиною основного коду, вони розміщувались після секції data?. Натомість у другому варіанті, де макроси винесені в окремий .inc файл, їх трансляція з’являється відразу після підключення бібліотек.

До спільних рис належать:

* у макровизначеннях наявні приховані коментарі та оголошення локальних міток;
* локальні мітки не були зашифровані;
* у трансляціях приховані коментарі не відображаються;
* у трансляціях також не видно локальних міток;
* виклики функцій через invoke не відображаються у трансляціях;
* присутні лише звичайні коментарі.

10. Зробити висновки по лабораторній роботі.

**Висновки**

Під час виконання лабораторної роботи було досліджено використання макросів у різних підходах — як при вбудовуванні в основний файл, так і при підключенні окремого .inc файлу. Виявлено відмінності у розташуванні макровизначень, а також спільні риси: приховані виклики invoke, відсутність прихованих коментарів і локальних міток у трансляції, але їх наявність у вихідному коді. Я зрозумів, що локальні мітки допомагають уникати конфліктів і спрощують структуру програми. Звичайні коментарі полегшують читання коду, а приховані — містять додаткову службову інформацію. Макроси роблять код коротшим, зрозумілішим і зручнішим для повторного використання. MASM32 при цьому забезпечує простий та ефективний інструментарій для роботи з макросами.