**Міністерство освіти і науки України**

**Національний технічний університет України**

**«Київський політехнічний інститут імені 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 групи ІМ-22 доц. Павлов В. Г.

Тимофеєв Даниіл Костянтинович

номер в списку групи: 23

Київ 2024

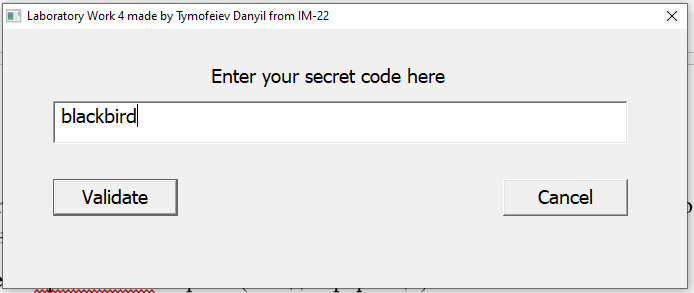
**Мета роботи:**

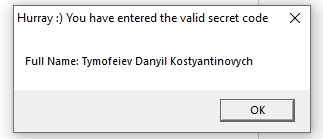
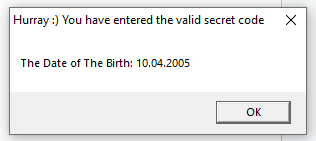
Вивчити технології створення і вживання макросів. Дослідження результатів роботи макрогенератору MASM.

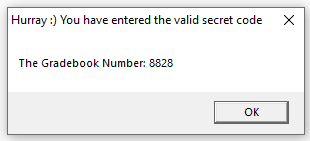
**Порядок виконання роботи:**

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.
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.
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 розташовуватись в окремому файлі.
9. Порівняти тексти розширених лістингів для обох варіантів розміщення макровизначень, знайти схожість і відмінності. Відобразити їх в звіті по лабораторній роботі.
10. Зробити висновки по лабораторній роботі.

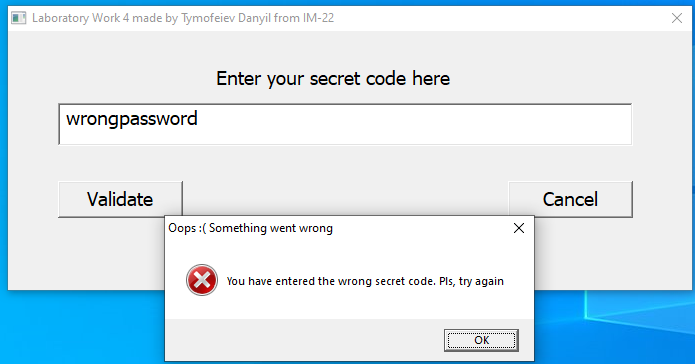
**Пункт 5.** Перевірити роботу програми шляхом введення як правильного, так і невірного паролів. Скріншоти програми

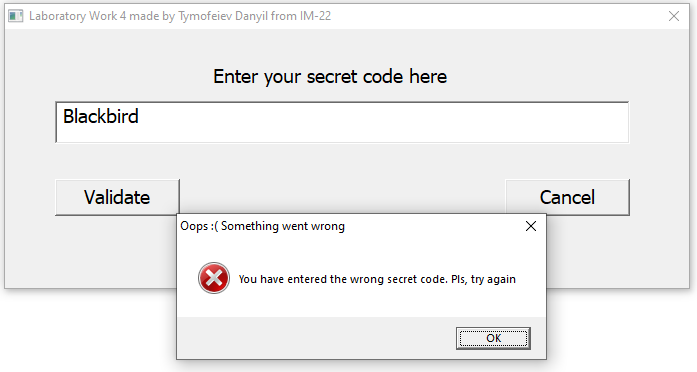
Введено правильний пароль (вивід інформації)

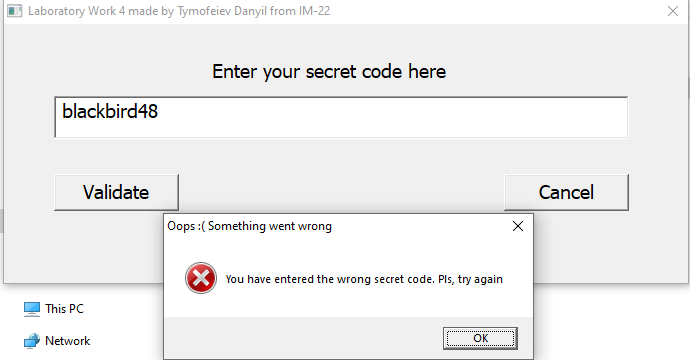




Що ж, як бачимо, тепер після введення валідного пароля інформація виводиться в трьох вікнах (спочатку повне ім’я, дата народження і номер залікової книжки). Такого “каскаду” інформації було досягнуто використанням макросу №1 ***TymofeievPrintMessageBox***

Введено неправильний пароль (повідомлення про помилку)

Введено неправильний пароль (перший символ у верхньому регістрі)

Введено неправильний пароль (додано зайві символи до кінця пароля)

Підсумовуючи, можу сказати, що згідно із завданням роботи в програму було додано 3 макроси:

1. ***TymofeievPrintMessageBox,*** який предназначений для “каскадного” виводу персональної інформації у різних вікнах (спочатку повне ім’я, дата народження і номер залікової книжки).
2. ***TymofeievEncryptPassword,*** який відповідає за шифрування введеного користувачем пароля.
3. ***TymofeievComparePasswords,*** який слугує для порівняння зашифрованого пароля, який ввів користувач, з хеш-кодом оригіналу пароля (**TymofeievPersonalSecredCode**).

**Демонстрація шифрування паролю за допомогою ключа**

Виконаємо шифрування пароля “blackbird” за допомогою **XOR** з використанням ключа “FIGHTEST1”**:**

1) Переведемо “blackbird” з ASCII в двійкову систему : 01100010 01101100 01100001 01100011 01101011 01100010 01101001 01110010 01100100

* b: 01100010
* l: 01101100
* a: 01100001
* c: 01100011
* k: 01101011
* b: 01100010
* i: 01101001
* r: 01110010
* d: 01100100

2) Переведемо “FIGHTEST1” з ASCII в двійкову систему : 01000110 01001001 01000111 01001000 01010100 01000101 01010011 01010100 00110001

* F: 01000110
* I: 01001001
* G: 01000111
* H: 01001000
* T: 01010100
* E: 01000101
* S: 01010011
* T: 01010100
* 1: 00110001

3) Зробимо XOR: 00100100 00100101 00100110 00101011 00111111 00100111 00111010 00100110 01010101

* 01100010 XOR 01000110 = 00100100 = $
* 01101100 XOR 01001001 = 00100101 = %
* 01100001 XOR 01000111 = 00100110 = &
* 01100011 XOR 01001000 = 00101011 = +
* 01101011 XOR 01010100 = 00111111 = ?
* 01100010 XOR 01000101 = 00100111 = '
* 01101001 XOR 01010011 = 00111010 = :
* 01110010 XOR 01010100 = 00100110 = &
* 01100100 XOR 00110001 = 01010101 = U

4) Переведемо в ASCII : “$%&+?':&U”

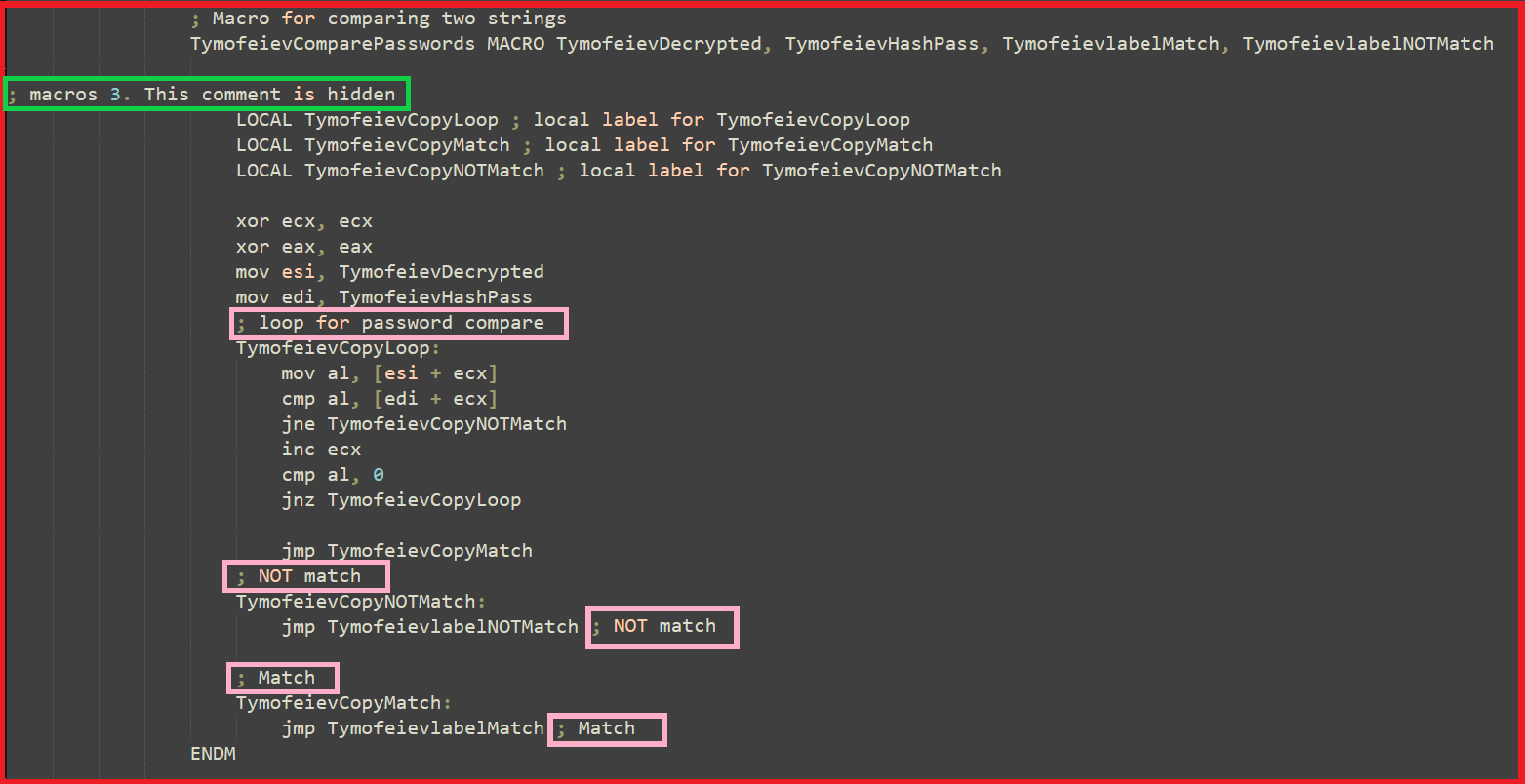
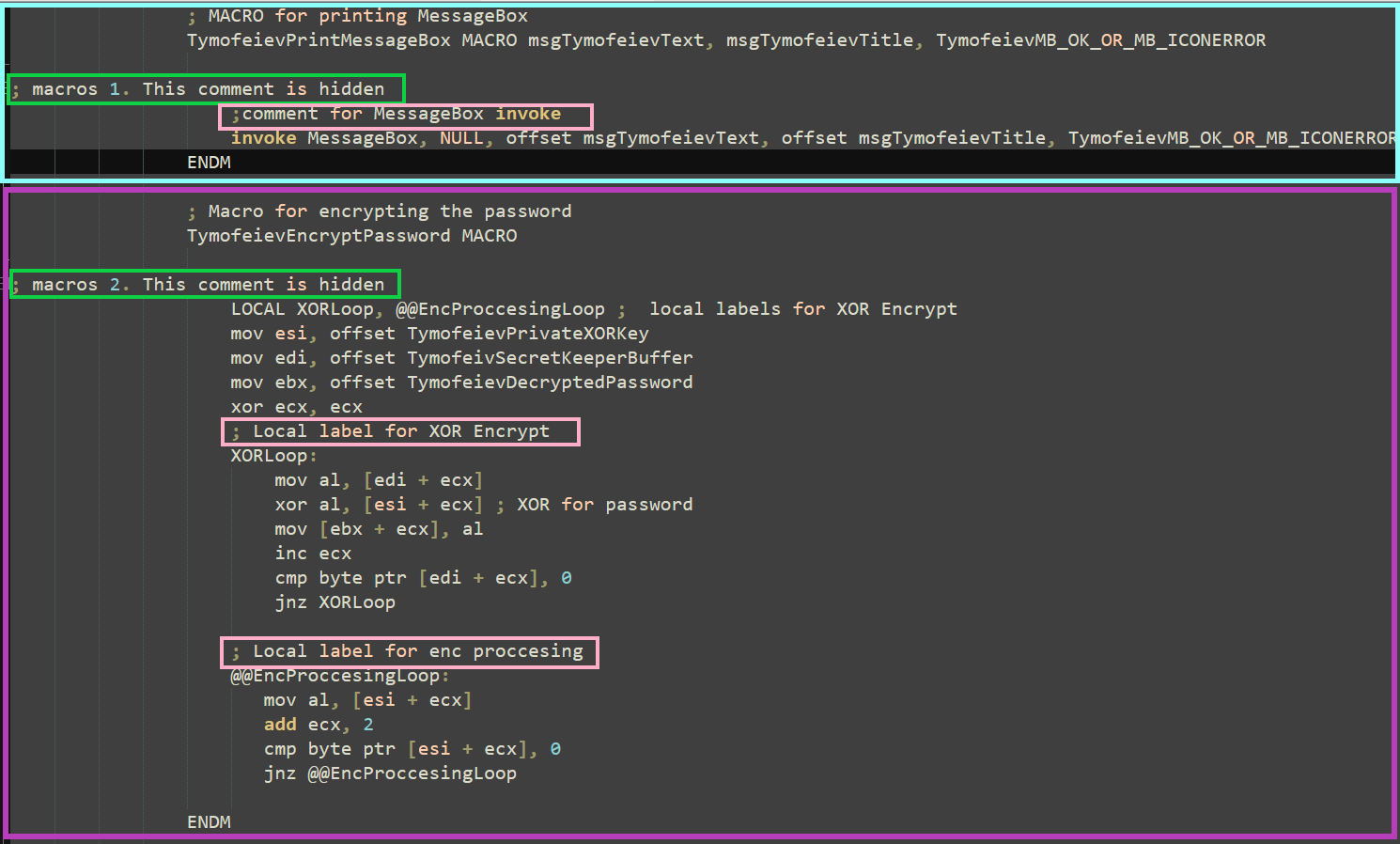
Тож, “$%&+?':&U” - зашифрований пароль.

**Пункти 6-7**. 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. **Програма і макроси в одному файлі**:

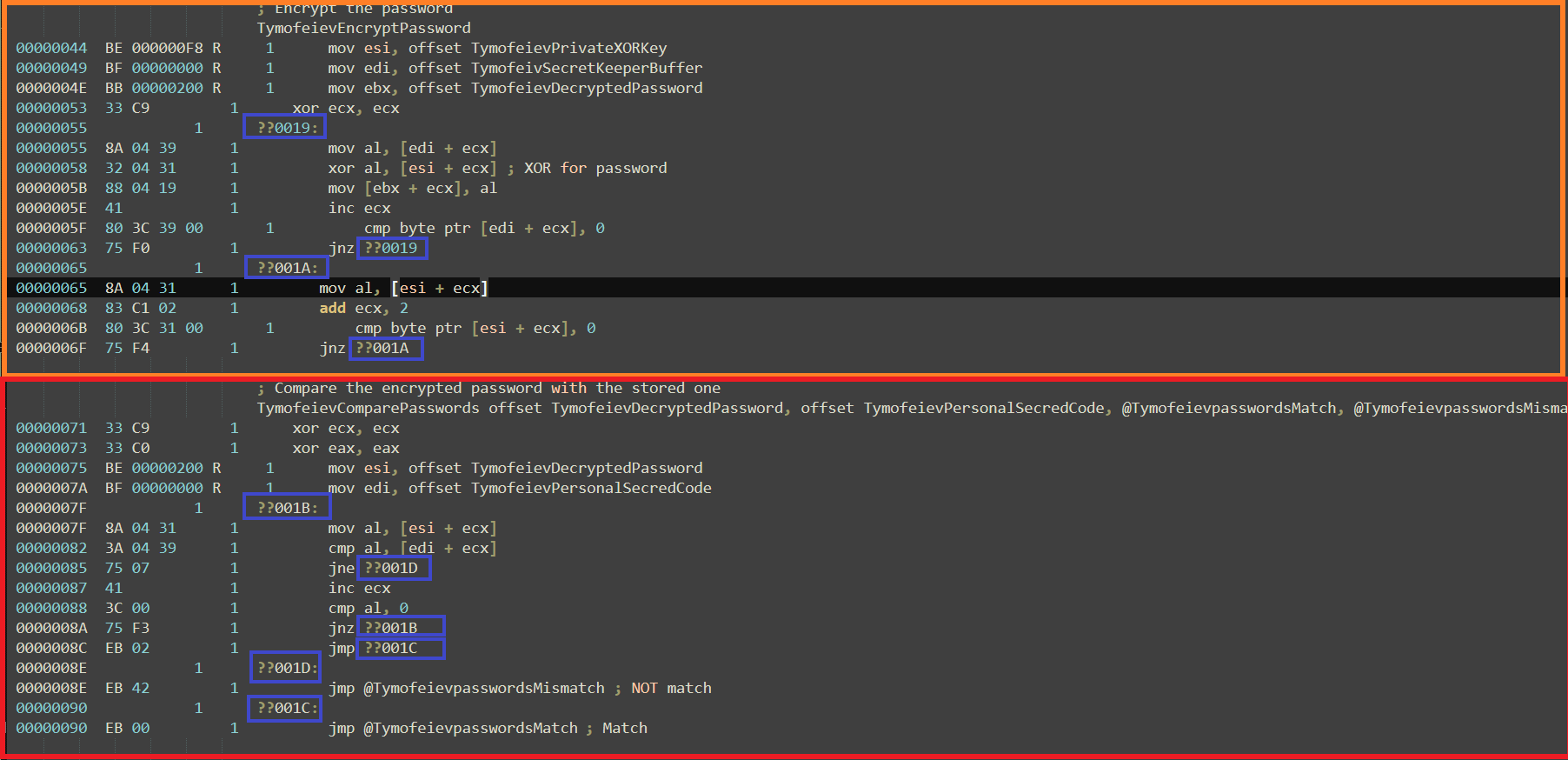
Три макроси згідно завданню було введено після секції **.data,** яка містить ініціалізовані дані. Спочатку введемо “легенду” для визначення макросів у лістингу програми, використовуючи кольоровий олівець.

* Макрос **TymofeievPrintMessageBox -** бірюзовий прямокутник
* Макрос **TymofeievEncryptPassword** - фіолетовий прямокутник
* Приховані коментарі - салатовий прямокутник
* Звичайні коментарі - рожевий прямокутник
* Макрос **TymofeievComparePasswords** - червоний прямокутник

**Макровизначення**:

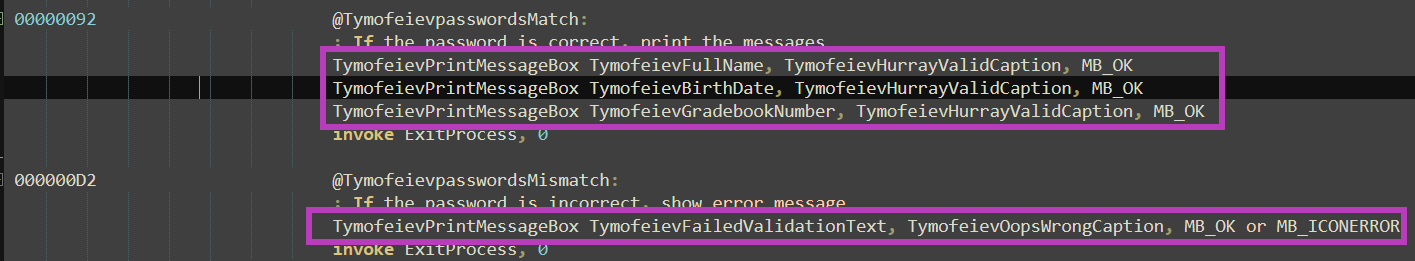
На цьому скріншоті можна побачити виділені макроси: TymofeievPrintMessageBox , який предназначений для виводу персональної інформації у різних вікнах. TymofeievEncryptPassword , який відповідає за шифрування введеного користувачем пароля. TymofeievComparePasswords , який слугує для порівняння зашифрованого пароля, який ввів користувач, з хеш-кодом оригіналу пароля. Ба більше, було побачено, що присутні, як приховані коментарі , так і звичайні коментарі. 

**Етап трансляції макросів:**

* Макрос **TymofeievEncryptPassword** - помаранчевий прямокутник
* Коди, замість локальних міток - синій прямокутник
* Макрос **TymofeievComparePasswords** - червоний прямокутник

На цьому малюнку виділені макроси на етапі трансляції. Перше, що можна побачити - це відсутність макросу **TymofeievPrintMessageBox**,це можна пояснити тим, що він містить єдиний рядок з **invoke**, а **invoke** при трансляції не відображається в макросах, тож ми не можемо побачити сам макрос, однак можемо побачити тільки використання макросу **TymofeievPrintMessageBox** (виділено фіолетовим прямокутником на малюнку нижче). Зате ми можемо побачити макроси **TymofeievEncryptPassword** та **TymofeievComparePasswords**.Ба більше, можна помітити, на прикладі третього макросу, що замість локальних міток з'явилися їхні коди. Ще однією особливістю трансляції макросів є відсутність прихованих коментарів, хоча звичайні коментарі присутні.

Використання **TymofeievPrintMessageBox**

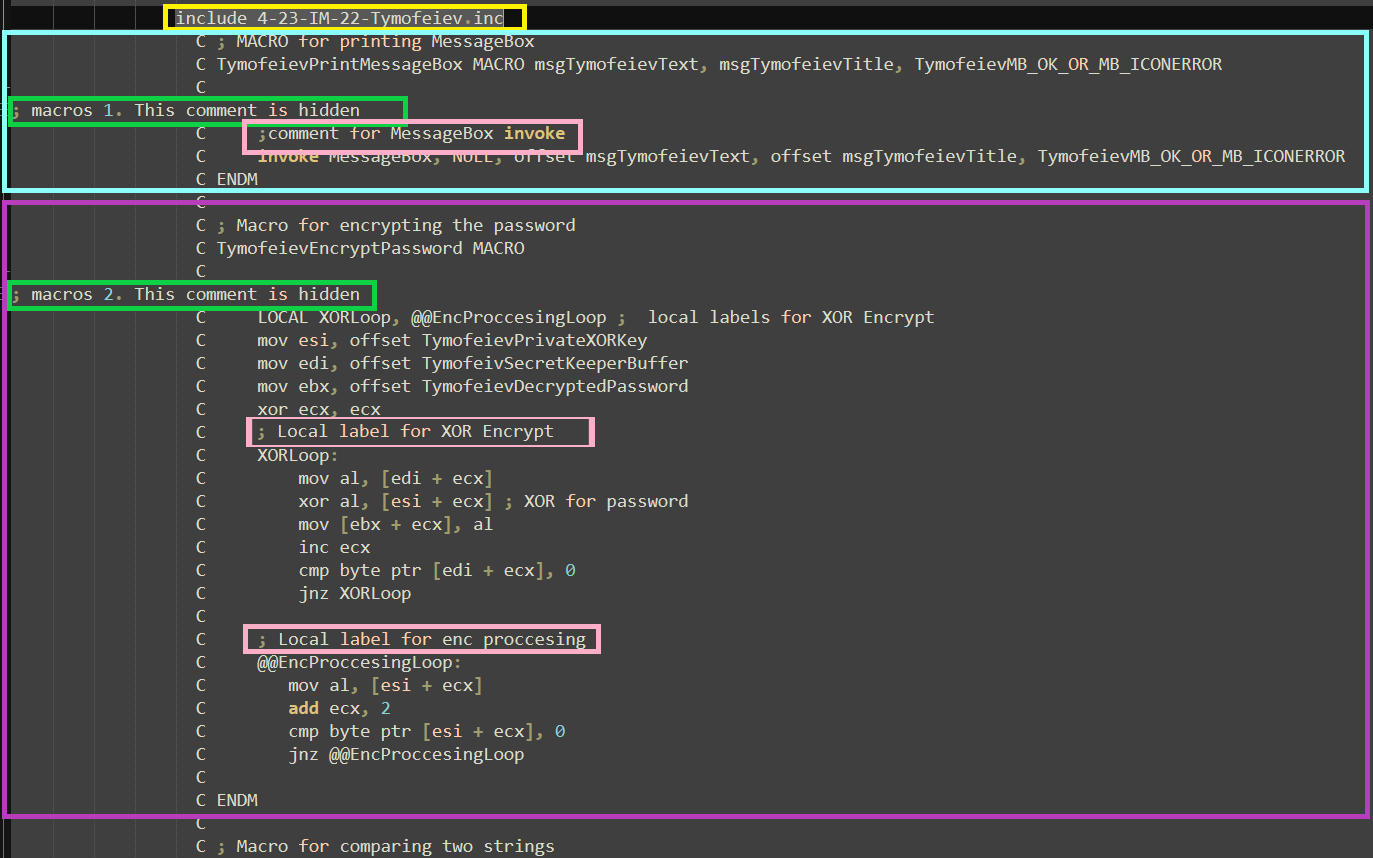


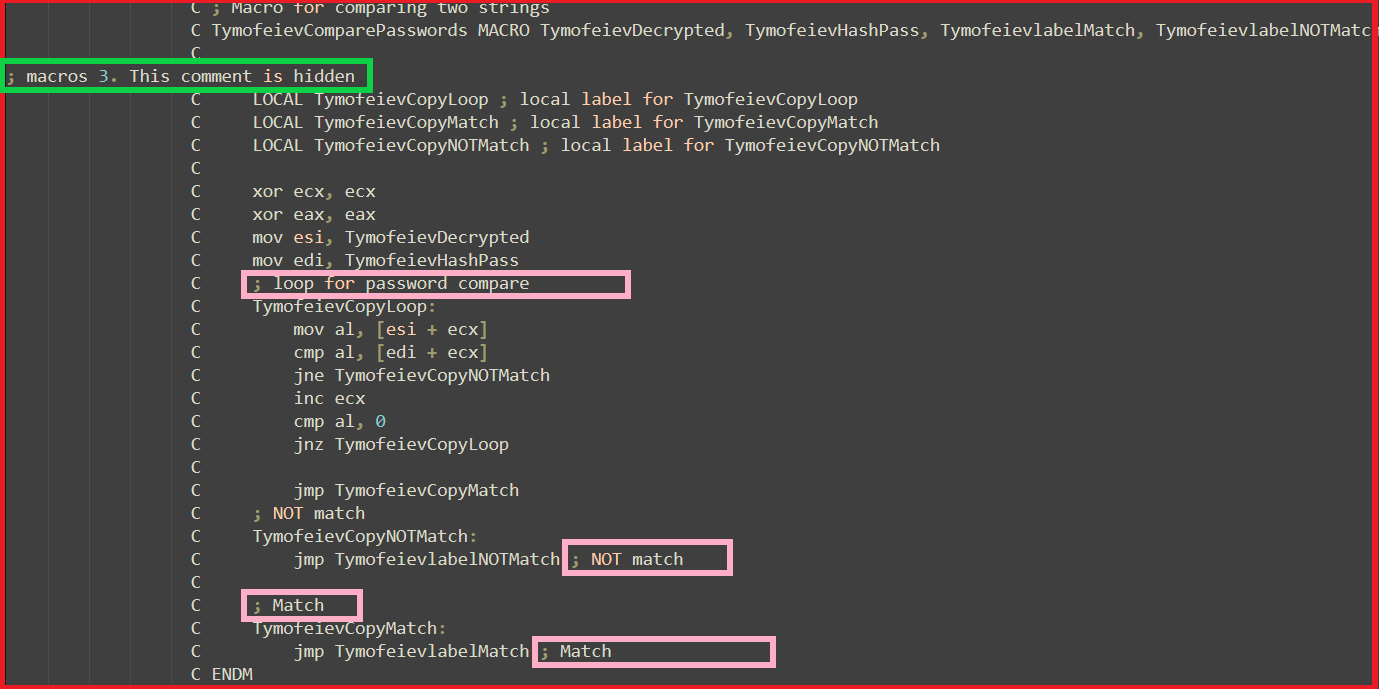
**Пункт 8**. Повторити дослідження, які виконані в п. 4 – 7 для другої версії програми, де всі перераховані в п. 2 і 3 макровизначення для макросів повинні розташовуватись в окремому файлі. **Макроси і сама програма в різних файлах** :

Три макроси з’явилися після **include 4-23-IM-22-Tymofeiev.inc**. Спочатку введемо “легенду” для визначення макросів у лістингу програми, використовуючи кольоровий олівець.

* Макрос **TymofeievPrintMessageBox -** бірюзовий прямокутник
* Макрос **TymofeievEncryptPassword** - фіолетовий прямокутник
* Приховані коментарі - салатовий прямокутник
* Звичайні коментарі - рожевий прямокутник
* імпорт макросів - жовтий прямокутник
* Макрос **TymofeievComparePasswords** - червоний прямокутник

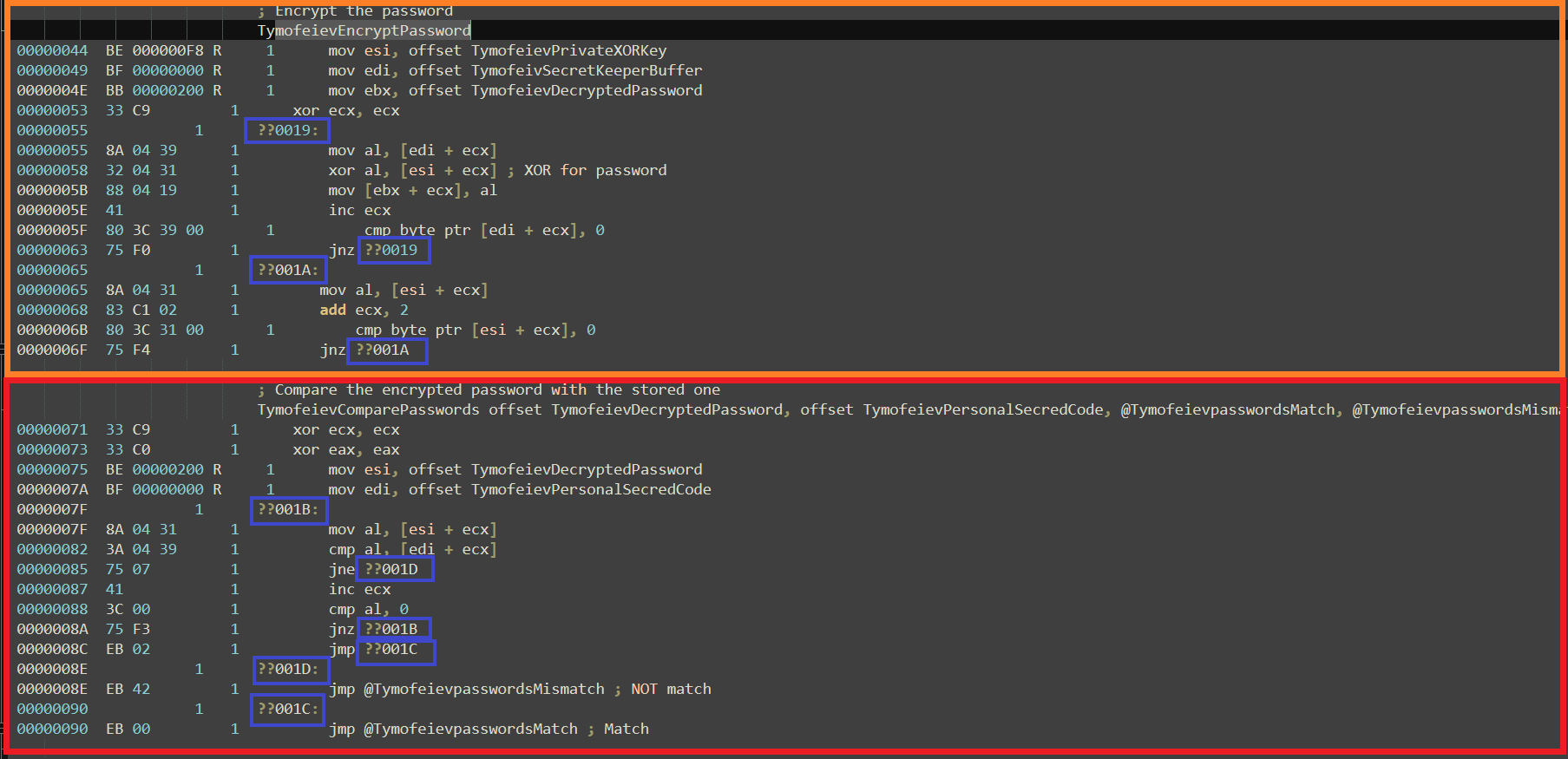
**Макровизначення**:



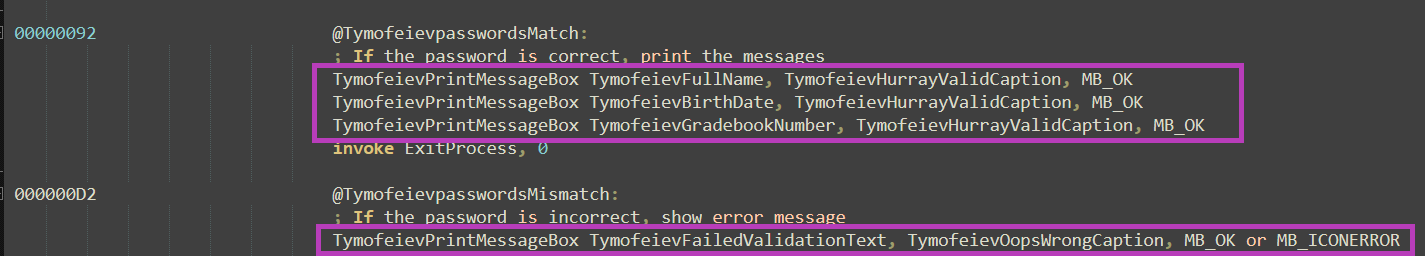


Бачимо, що в цьому варіанті лістингу програми можна побачити, що самі макроси йдуть після include-файлу. Також, як і минулого разу, виділено макроси : TymofeievPrintMessageBox, TymofeievEncryptPassword TymofeievComparePasswords. Ба більше, було побачено, що присутні, як приховані коментарі , так і звичайні коментарі.

**Етап трансляції макросів:**

* Макрос **TymofeievEncryptPassword** - помаранчевий прямокутник
* Коди, замість локальних міток - синій прямокутник
* Макрос **TymofeievComparePasswords** - червоний прямокутник

Що ж, у другому варіанті програми багато схожого. На цьому малюнку також виділено макроси на етапі трансляції. Так само відсутній макрос **TymofeievPrintMessageBox**. Це явище пояснюється тим, що він містить єдиний рядок з **invoke**, а **invoke** при трансляції не відображається в макросах, тож ми не можемо побачити сам макрос, однак можемо побачити тільки використання макросу **TymofeievPrintMessageBox** (виділено фіолетовим прямокутником на малюнку нижче). Зате ми можемо побачити макроси **TymofeievEncryptPassword** та **TymofeievComparePasswords**. Більш того було помічено, що замість локальних міток з'явилися їхні коди. Як і минулого разу присутня особливість трансляції макросів - відсутність прихованих коментарів, хоча звичайні коментарі присутні.

Використання **TymofeievPrintMessageBox**

**Висновок**:

Отже, під час виконання лабораторної роботи я вивчив і опанував технології створення і вживання макросів. Виявилося, що макроси дуже зручний інструмент: їх пишуть для повторного використання коду, що часто використовується. Ба більше, дослідив результати роботи макрогенератору MASM32. Також було написано дві програми: макроси і власне програма знаходилися в рамках одного файлу; макроси і власне програма були розмежовані в різних файлах. Більш того, було засвоєно використання локальних міток у макросах для організації внутрішньої логіки в межах блоку, локальні мітки не можуть бути використаними поза макросу.

При порівнянні двох застосунків було досліджено, що схожості переважають над відмінностями. Спільного в них багато, а саме: видно у макровизначенні звичайні і приховані коментарі; у лістингу під час трансляції відсутні приховані коментарі, однак наявні звичайні; під час трансляції відсутній макрос, що відповідає за виведення MessageBox, оскільки він містить єдиний рядок з **invoke**, а **invoke** при трансляції не відображається в макросах; назви локальних міток перетворюються на коди. Що стосується відмінностей, то основна відмінність - це розташування макровизначень у лістингу програм. За такою відміністю можна дізнатися, де знаходяться макроси : якщо макровизначення знаходяться біля **include-файлу,** то макроси і програма в різних файлах.

Що стосується, прихованих і звичайних коментарів, то вони використовуються для опису чи документації коду, однак приховані коментарі відсутні на етапі трансляції, хоча звичайні присутні. Зазвичай приховані коментарі використовуються в макросах. Локальні мітки - це блоки коду, які використовуються всередині макросів, і недоступні не в межах макросу. Їх створюють за допомогою ключового слова **LOCAL**. На етапі трансляції назви міток перетворюються на коди.