**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 у MASM”

Виконав: Перевірив:

студент групи ІМ-21 доцент

Лесько Дмитро Миколайович Павлов В.Г.

Номер у списку групи: 14

**Київ 2024**

**Мета роботи:** Вивчити технології створення і вживання макросів. Дослі дження результатів роботи макрогенератору MASM.

**Виконання роботи:**

Отже, після вивчення правил використання макросів у MASM32 я переробив свою лабораторну роботу №3, виділивши три макроси:

1. Для виводу інформації студента або помилки вводу паролю
2. Для XORингу введеного користувачем паролю
3. Для порівняння паролю та виведення відповідних повідомлень

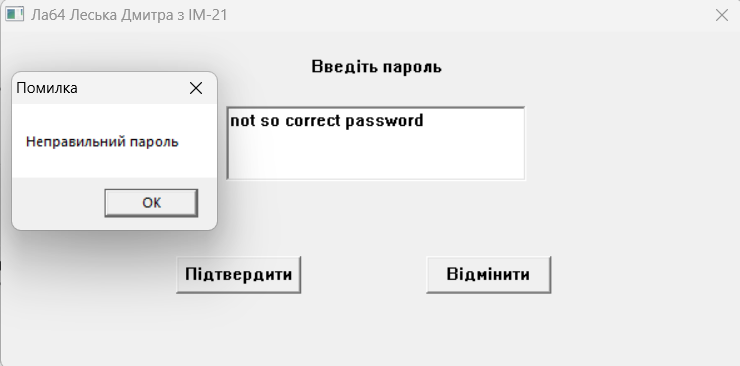
Можу сказати одразу, що використання макросів справило на мене вдале враження, адже вони нам дають дуже зручний інструмент для перевикористання нашого коду, тобто допомагає дотримуватися принципу DRY- don`t repeat yourself. Також я використав для цієї версії програми мітки та коментарі (приховані та звичайні). Отже перевіримо як програма працює з макросами:



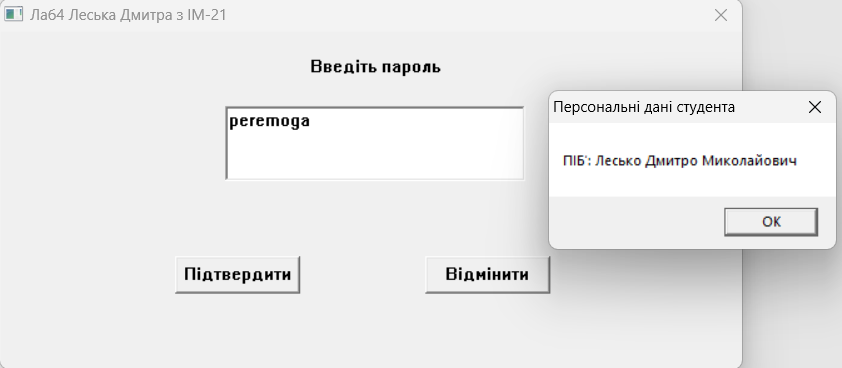
Вікно, яке пропонує нам ввести пароль (вікна взяті з моєї минулої роботи)

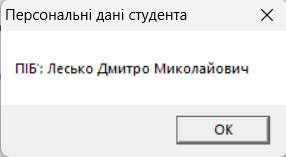


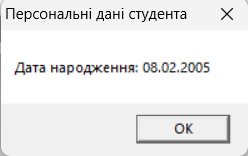
У першому випадку ми вводимо неправильний пароль і після натискання на кнопку «Підтвердити» ми побачимо повідомлення, що пароль неправильний, програма завершиться:

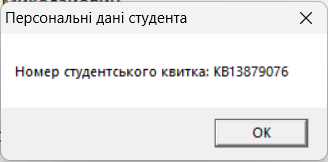


Коли ж ми вводимо правильний пароль, то ми можемо побачити, що програма буде виводити нам інформацію студента:









Після чого програма завершить своє виконання. Порівнюючи з минулою лабораторною роботою, функціонал сильно не змінився, проте тепер під капотом ми використовуємо макроси. Саме для демонстрації їхньої роботи я три рази поспіль виводжу інформацію студента частинами, щоб показати зручність їхнього використання. І це дійсно правда, адже у нашій програмі стало менше коду, бо нам тепер не потрібно для кожної події виводу інформації створювати окрему функцію і використовувати в ній окремі змінні, тепер у нас є універсальний макрос.

Також я повторно провів шифрування, щоб воно відповідало усім вимогам

Шифрування пароля за допомогою XOR з ключем, довжиною відповідно паролю:

Пароль: **peremoga**

Ключ: **GQLPPE54**

Шифрування паролю:

p = 11210 = 011100002

e = 10110 = 011001012

r = 11410 = 011100102

e = 10110 = 011001012

m = 10910 = 011011012

o = 11110 = 011011112

g = 10310 = 011001112

a = 9710 = 011000012

G = 7110 = 010001112

Q = 8110 = 010100012

L = 7610 = 010011002

P = 8010 = 010100002

P = 8010 = 010100002

E = 6910 = 010001012

5 = 5310 = 001101012

4 = 5210 = 001101002

p XOR G = 01110000 XOR 01000111 = 001101112 = 5510 = 7

e XOR Q = 01100101 XOR 01010001 = 001101002 = 5210 = 4

r XOR L = 01110010 XOR 01001100 = 001111102 = 6210 = >

e XOR P = 01100101 XOR 01010000 = 001101012 = 5310 = 5

m XOR P = 01101101 XOR 01010000 = 001111012 = 6110 = =

o XOR E = 01101111 XOR 01000101 = 001010102 = 4210 = \*

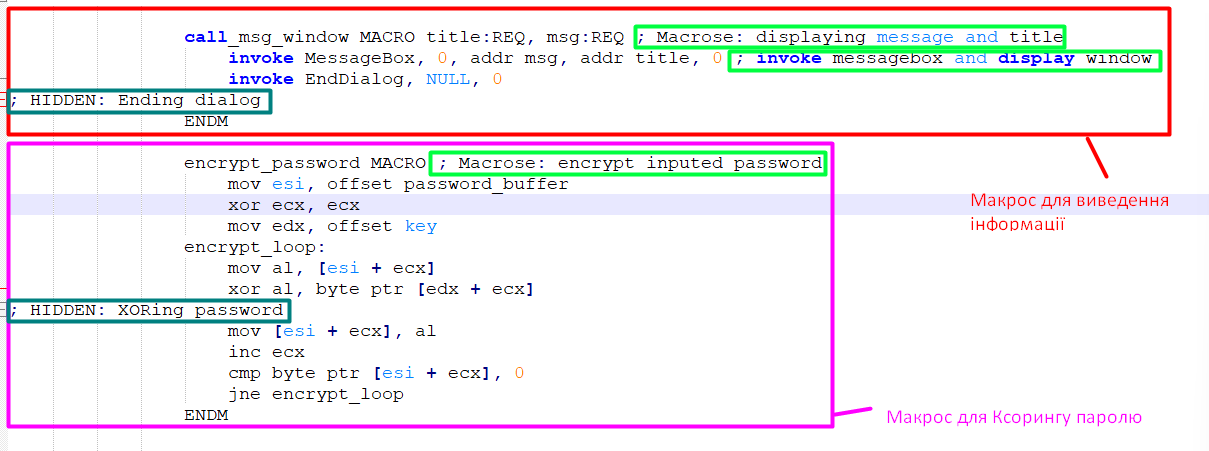
g XOR 5 = 01100111 XOR 00110101 = 010100102 = 8210 = R

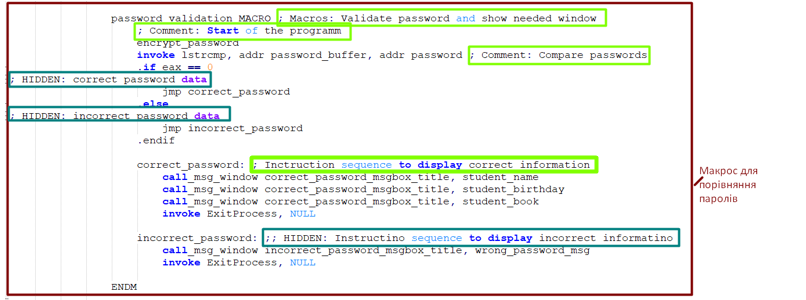
a XOR 4 = 01100001 XOR 00110100 = 010101012 = 8510 = U

Пароль: **74>5=\*RU**

**Лістенінг**

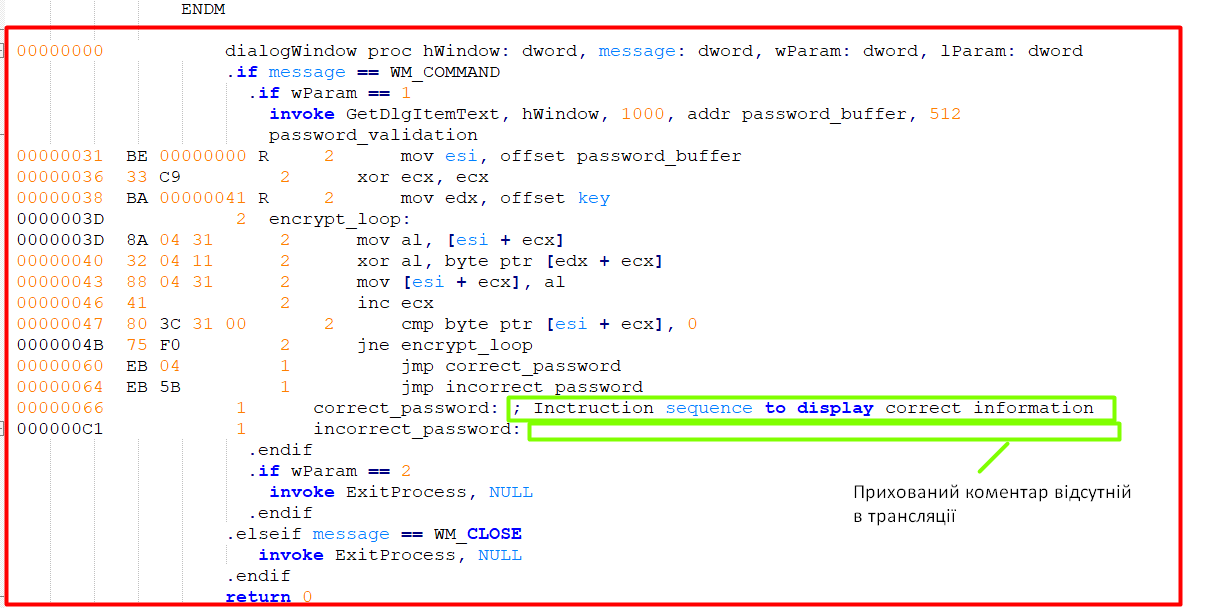
Далі, як і сказано у завданні, я провів лістенінг програми і на виході отримав файл з розширенням lst. Дослідимо його:



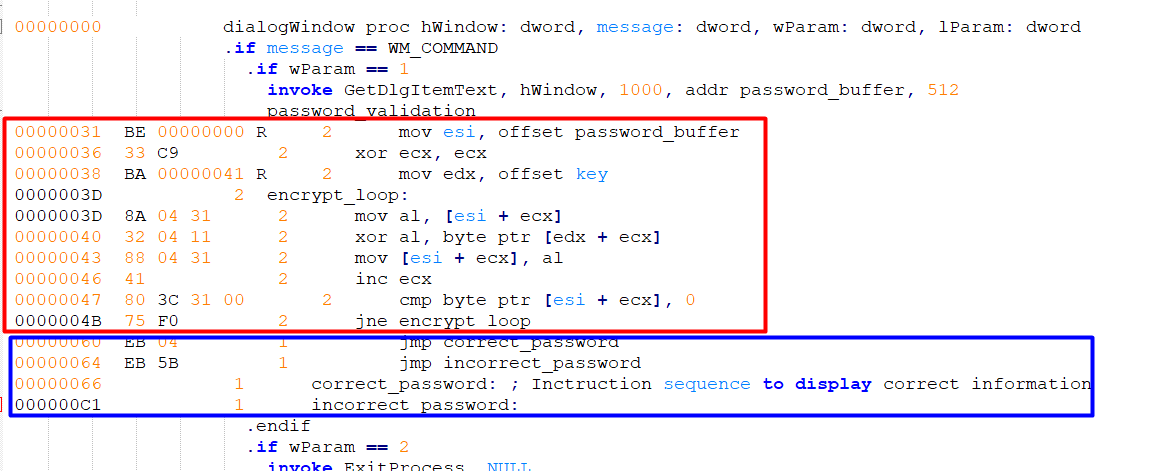


Тут на фотографіях я виділив у лістинінг файлі макроси, як вони виглядають, я їх виділив великими кольоровими квадратами. Також у їх тілі я виділив приховані та звичайні коментарі, приховані я виділив темнозеленим, звичайні- салатовим.

Також далі у програмі можем побачити **Трансляцію** макросів у наш код:



Таким чином ми можемо побачити трансляцію макросів у наш код, тобто яким чином вони застосовуються, також які команди використовуються для цього. Хотів би ще звернути уваги, що під час трансляції відображаються тільки звичайні коментарі, приховані не відображаються (я позначив місце, де у нашому коді є звичайний і прихований пароль, ми бачимо у трансляції тільки звичайний).

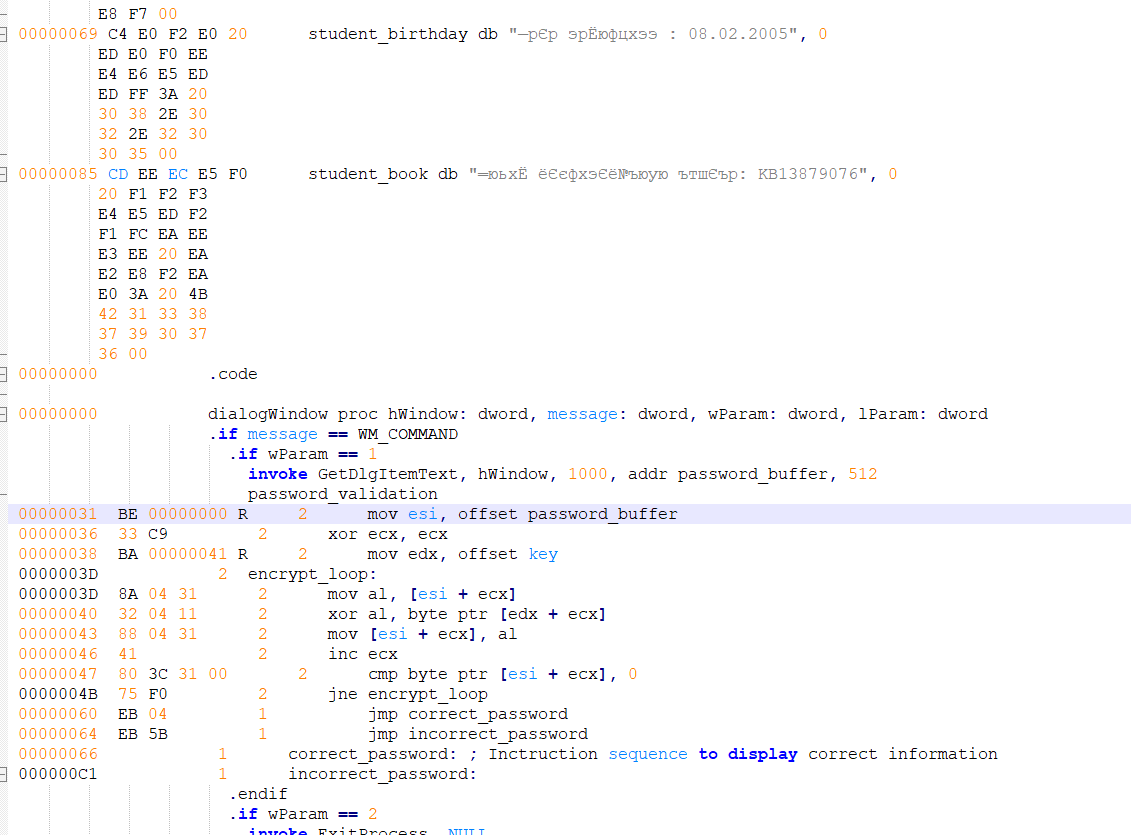


Також в цьому шматку видно трансляцію двох макросів. Червоний- шифрування паролю, синій- порівняння та вивід інформації.

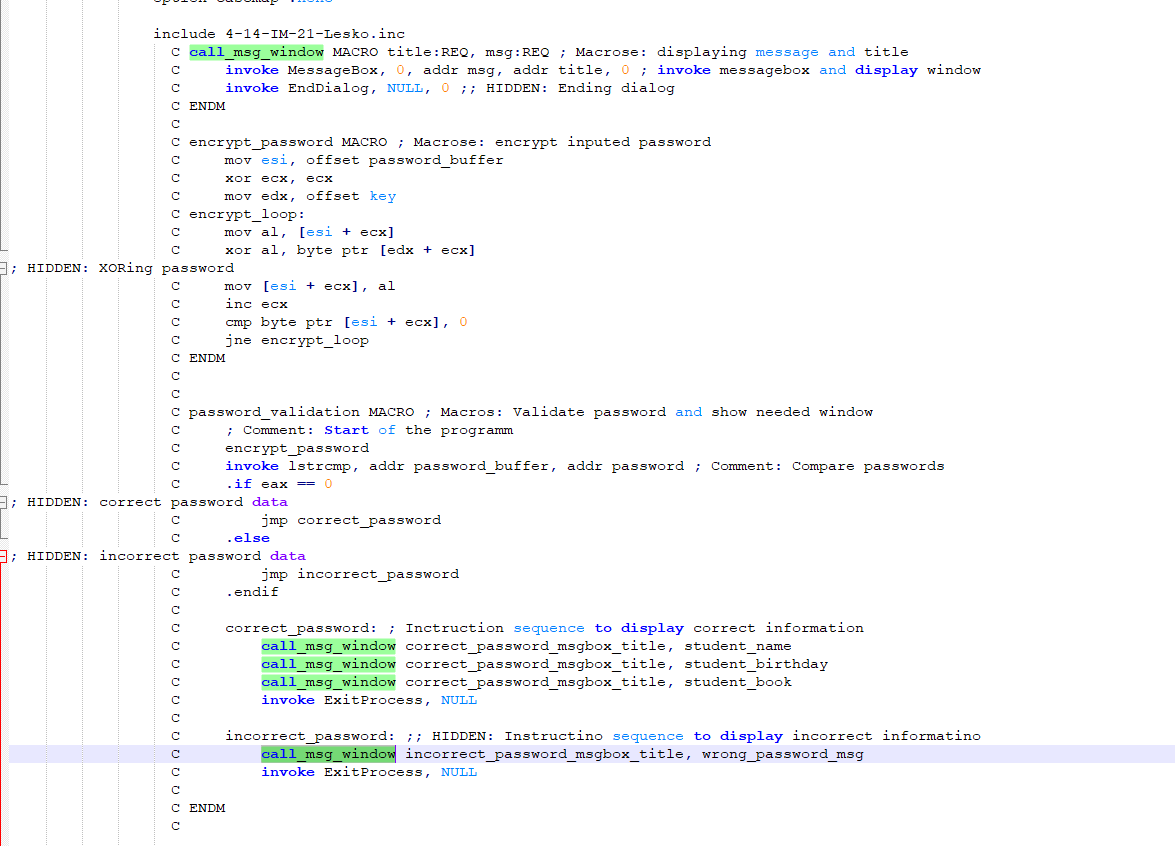
Також, оскільки макрос для виведення інформації викликається в макросі виділеним синім кольором (для порівняння паролю), то він і транслюється у нього останнім, тому ми його не бачимо.

Далі для останніх трьох пунктів переводимо наші макроси у файл 4-14-IM-21-Lesko.inc та інклюдимо його у наш основний, щоб можна було використовувати макроси з іншого файлу. Одразу скажу, що це досить зручно, адже ми потім зможемо перевикористовувати ці макроси у інших програмах, які нам будуть потрібні. Компілюємо програму, вона компілюється вдало, працює так само, як і перший варіант з одним цільним файлом, проте тепер ми відокремили наші макроси в окремий файл.

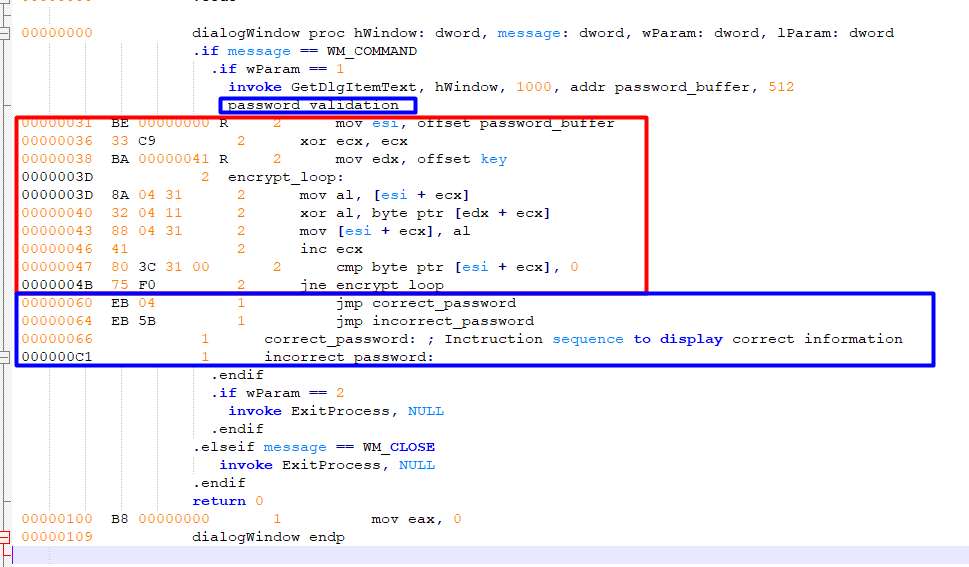
**Лістенінг**



Відкривши файл лістинінгу одразу бачимо, що макровизначення пропали з початку частини програми в секції .code.



Також бачимо, що на самому початку лістенінгу ми інклюдимо макроси і визначаємо їх, щоб надалі їх можна було перевикористовувати. Тобто це є однією з різниць макровизначень у двох варіантах програми.



Далі бачимо, що у нас у коді використовуються наші макроси і транслюються, аналогічно з першою програмою.

Так само бачимо переніс у трансляцію тільки звичайних коментарів, приховані не беруться до уваги.

Тобто можна сказати, що є різниця у місцях та способах оголошення макросів, проте різниці у їхній трансляції у код я не побачив.

**Висновок**

Зробивши цю лабораторну роботу, я вивчив новий спосіб перевикористання блоків коду- макроси, зміг використати їх з обов’язковими параметрами, покращив свій код, використав ще оди метод ксорингу паролю, дізнався про різні види коментарів та на що їх різновиди впливають. Також поглиблено дослідив програму за допомогою лістинингу: побачив як макроси «вбудовуються» в наш код, як вони оголошуються, як відображаються мітки і коментарі у лістинінгу. Також за допомогою виділення макросів в окремий файл з розширенням .inc, що дозволило зменшити розмір коду основного файлу та у майбутньому дає можливість повторно використовувати вже готові рішення, написані власноруч. Також я побачив різницю у макровизначеннях у двох варіантах програми, такі досліди дають якісніше дослідити принципи роботи програми, самі базиси програмування, що допомагає якісно відрізняти знання певних рівнів.

Дізнався про використання локальних міток, які можна зручно використовувати для виконання певних заготовлених блоків коду. Це можна також використовувати для повторення блоків коду, як у макросах, проте по іншому, все ж таки принципи та можливості їх роботи сильно відрізняються.