**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 ІІ курсу ФІОТ доц. Павлов В. Г.

групи ІМ-22

Сачко Максим Євгенійович

номер у списку групи (варіант): 20

Київ 2024

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

**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.**

Схема шифрування:

У циклі шифруються усі символи пароля за допомогою базового метода шифрування XOR з ключем “gOLDeONRAIN”.

Ключ: gOLDeONRAIN

Пароль: ZeusOlympus

Переведення символів у двійкову систему координат використовуючи ASCII таблицю:

gOLDeONRAIN:

g = 10310 = 0110 01112

O = 7910 = 0100 11112

L = 7610 = 0100 11002

D = 6810 = 0100 01002

e = 10110 = 0110 01012

O = 7910 = 0100 11112

N = 7810 = 0100 11102

R = 8210 = 0101 00102

A = 6510 = 0100 00012

I = 7310 = 0100 10012

N = 7810 = 0100 11102

ZeusOlympus:

Z = 9010 = 0101 10102

e = 10110 = 0110 01012

u = 11710 = 0111 01012

s = 11510 = 0111 00112

O = 7910 = 0100 11112

l = 10810 = 0110 11002

y = 12110 = 0111 10012

m = 10910 = 0110 11012

p = 11210 = 0111 00002

u = 11710 = 0111 01012

s = 11510 = 0111 00112

Шифрування:

Z XOR g = 0101 10102 XOR 0110 01112 = 0011 11012 = 6110 = **=**

e XOR O = 0110 01012 XOR 0100 11112 = 0010 10102 = 4210 = **\***

u XOR L = 0111 01012 XOR 0100 11002 = 0011 10012 = 5710 = **9**

s XOR D = 0111 00112 XOR 0100 01002 = 0011 01112 = 5510 = **7**

O XOR e = 0100 11112 XOR 0110 01012 = 0010 10102 = 4210 = **\***

l XOR O = 0110 11002 XOR 0100 11112 = 0010 00112 = 3510 = **#**

y XOR N = 0111 10012 XOR 0100 11102 = 0011 01112 = 5510 = **7**

m XOR R = 0110 11012 XOR 0101 00102 = 0011 11112 = 6310 = **?**

p XOR A = 0111 00002 XOR 0100 00012 = 0011 00012 = 4910 = **1**

u XOR I = 0111 01012 XOR 0100 10012 = 0011 11002 = 6010 = **<**

s XOR N = 0111 00112 XOR 0100 11102 = 0011 11012 = 6110 = **=**

Отже, зашифрований пароль – **=\*97\*#7?1<=**

**4. Виконати компіляцію і компоновку файлу програми.**

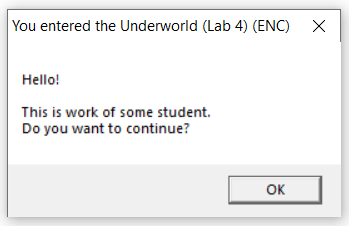
**5. Перевірити роботу програми шляхом введення як правильного, так і невірного паролів.**

Правильний пароль – ZeusOlympus

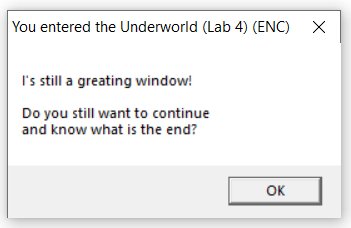
**Скріншоти виконання програми:**

Програма з макросами в коді:

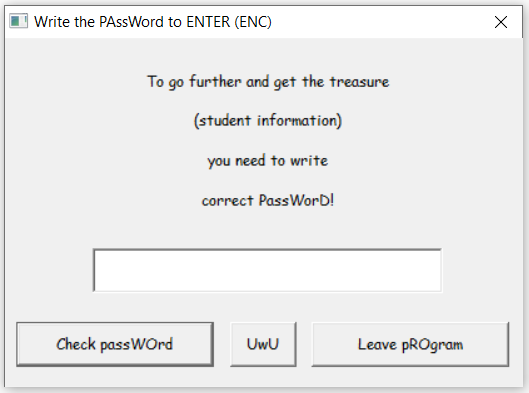
Вікно при запуску програми (прев’ю) – використано макрос 1 для відображення



Друге вікно після запуску програми (це теж прев’ю) – використано макрос 1 для відображення



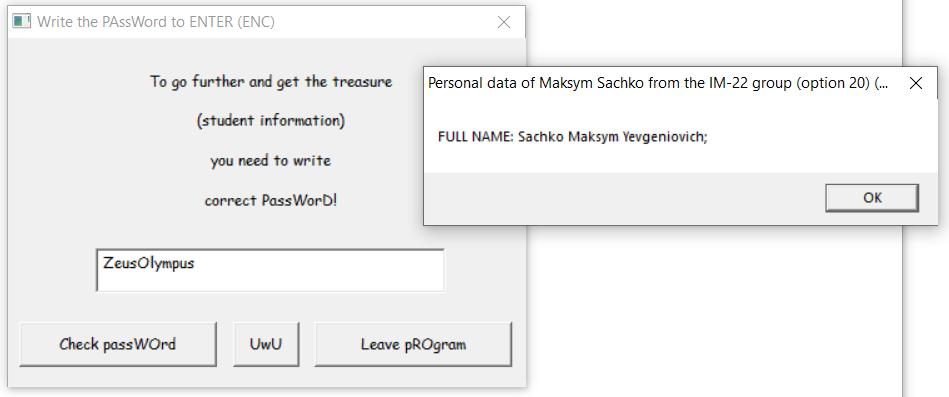
Вікно для введення паролю

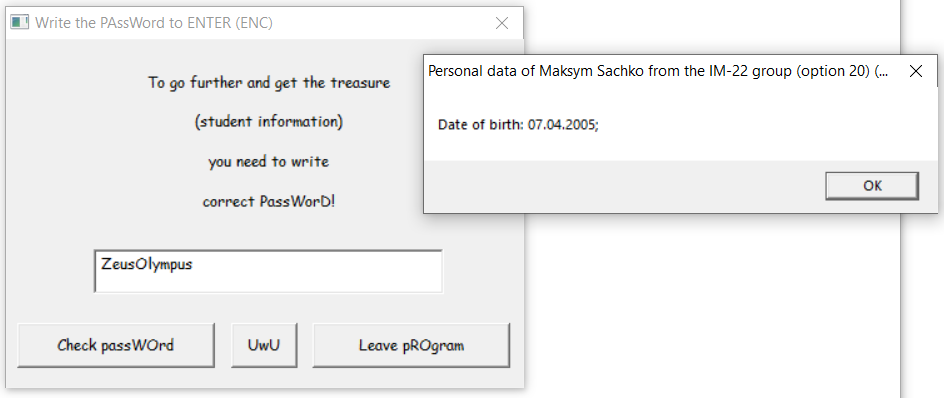


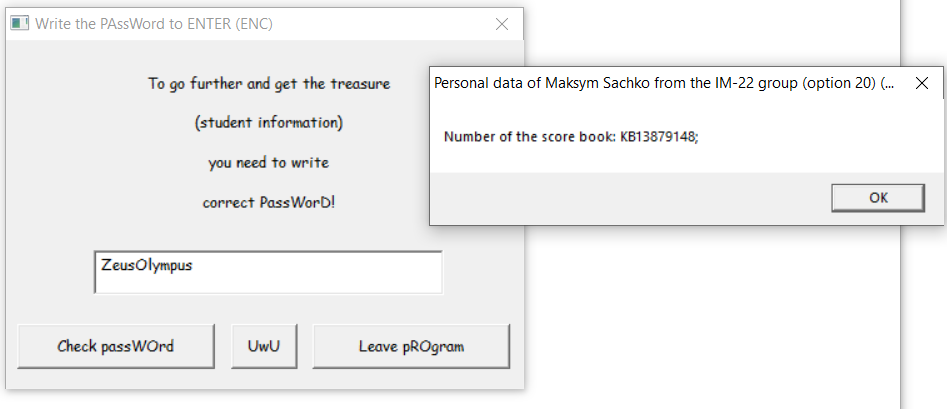
Введення паролю:

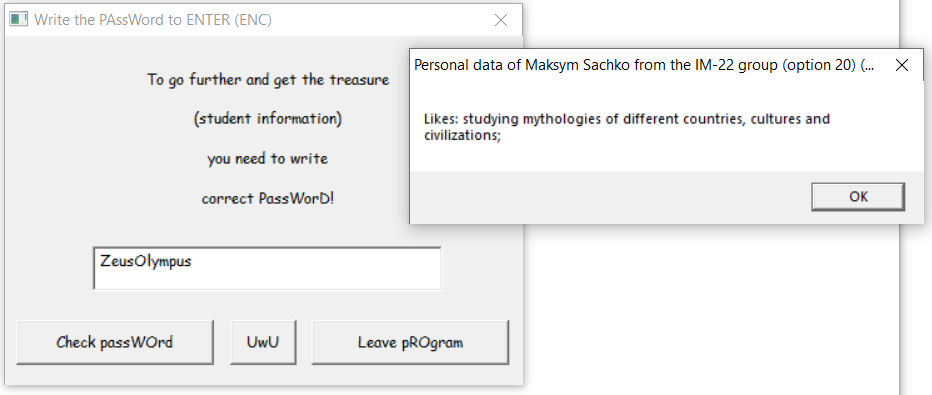
1. Введення правильного паролю

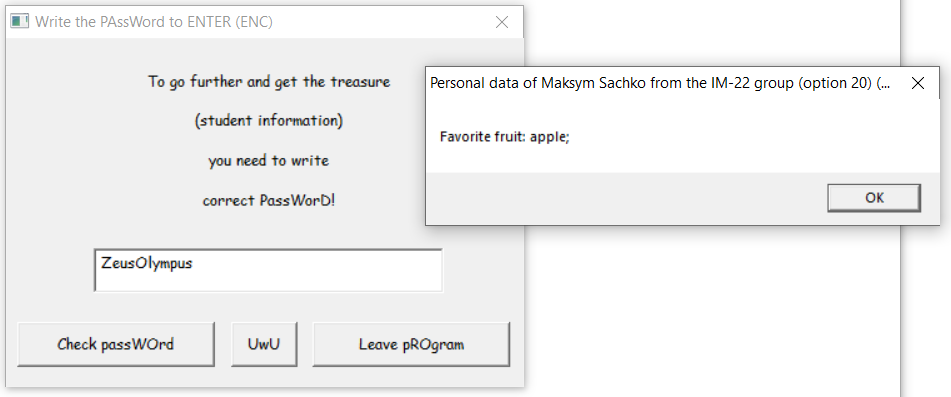
При введенні правильного паролю кожний фрагмент даних (ім’я, номер залікової книжки, дата народження і тд.) тепер виводиться в своєму вікні послідовно завдяки реалізації макроса 1, а не як у 3-й лабораторній, коли всі дані показувалися в одному вікні

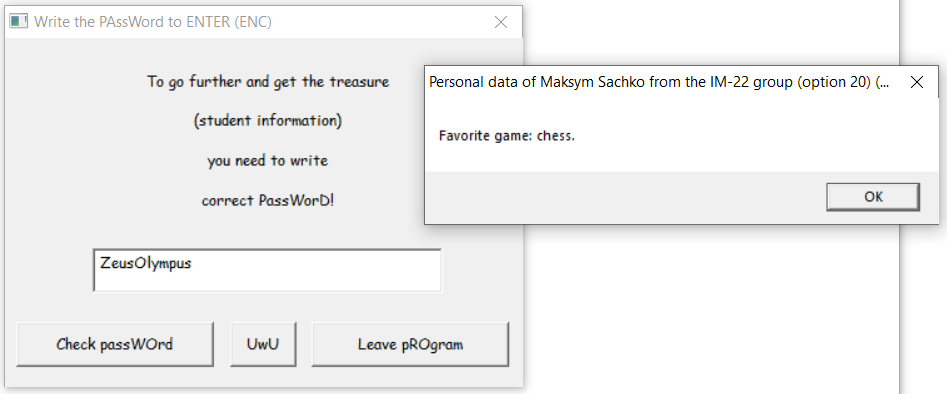




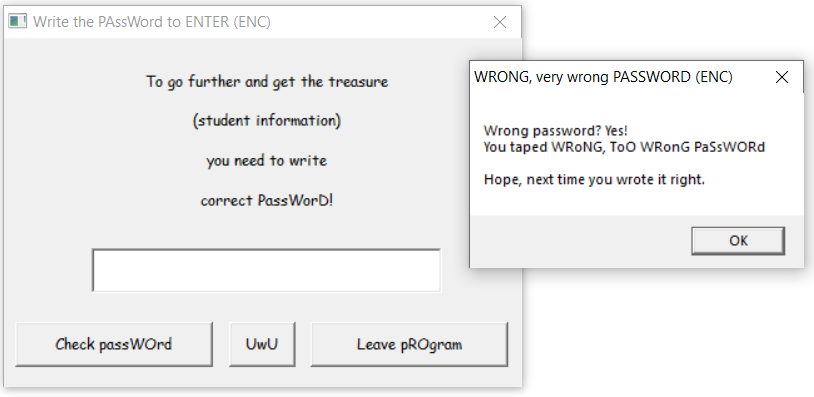




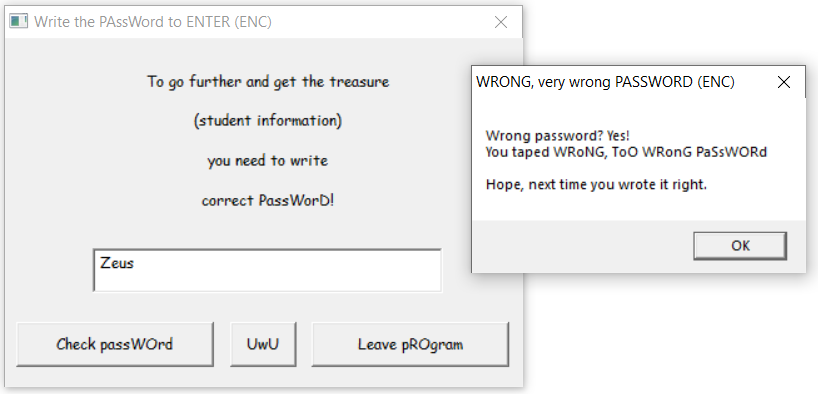




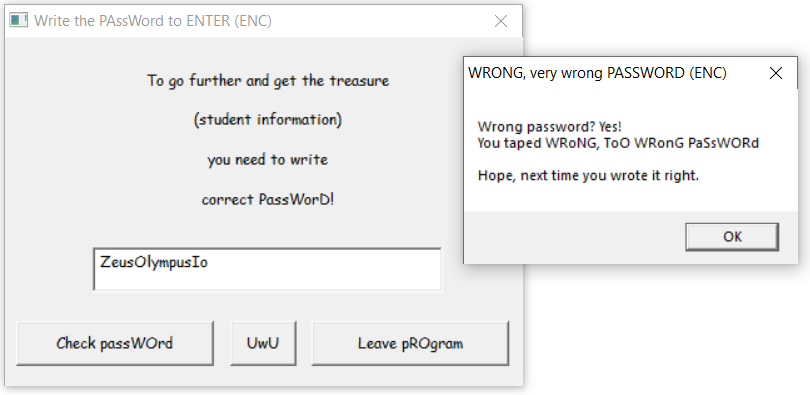
2. Введення неправильного паролю (пустий рядок) – використано макрос 1 для відображення вікна з повідомленням про неправильний пароль



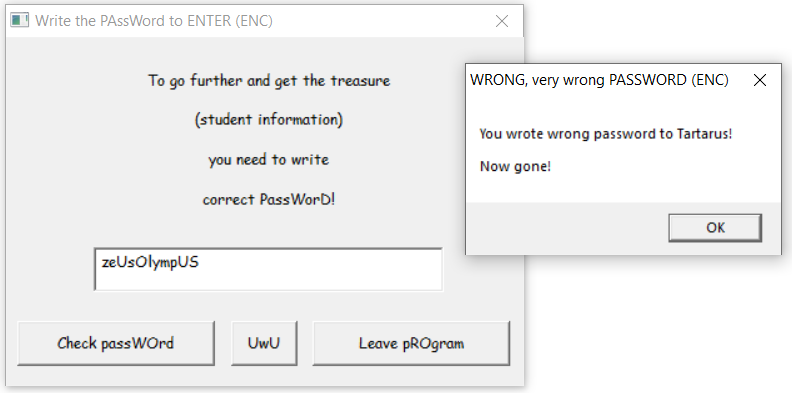
3. Введення неправильного паролю (неповний пароль) – використано макрос 1 для відображення вікна з повідомленням про неправильний пароль



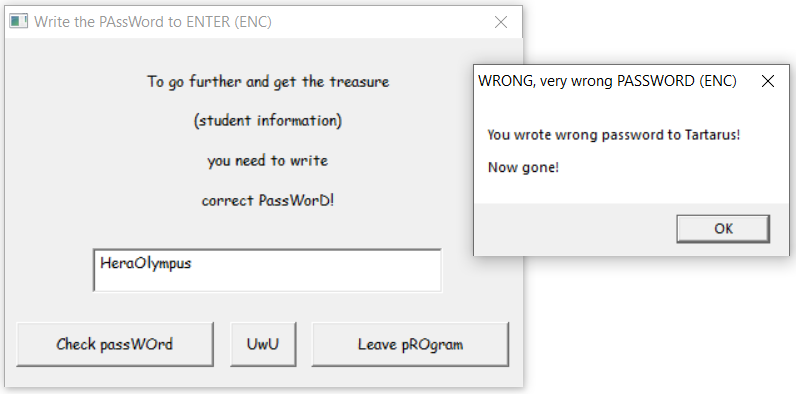
4. Введення неправильного паролю (пароль з надлишковою кількістю літер) – використано макрос 1 для відображення вікна з повідомленням про неправильний пароль



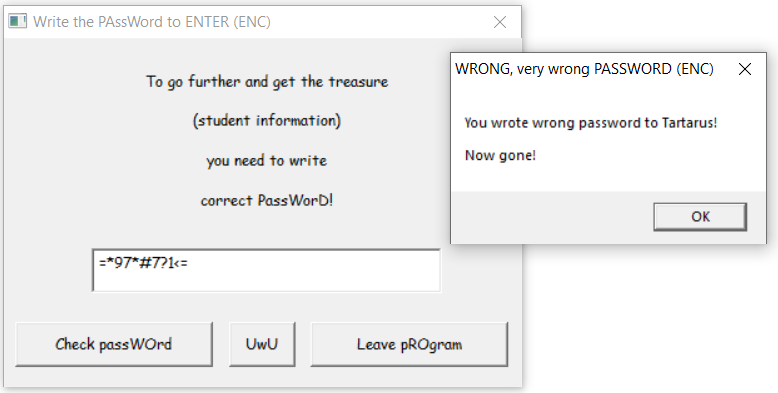
5. Введення неправильного паролю (пароль з іншим регістром деяких символів) – використано макрос 1 для відображення вікна з повідомленням про неправильний пароль



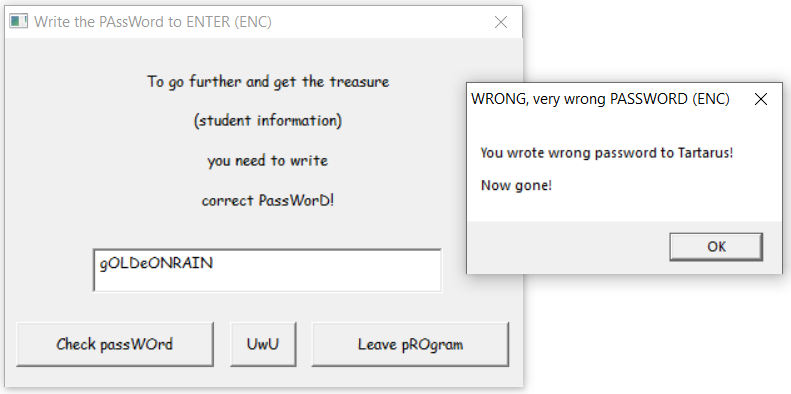
6. Введення неправильного паролю (зовсім інший пароль) – використано макрос 1 для відображення вікна з повідомленням про неправильний пароль



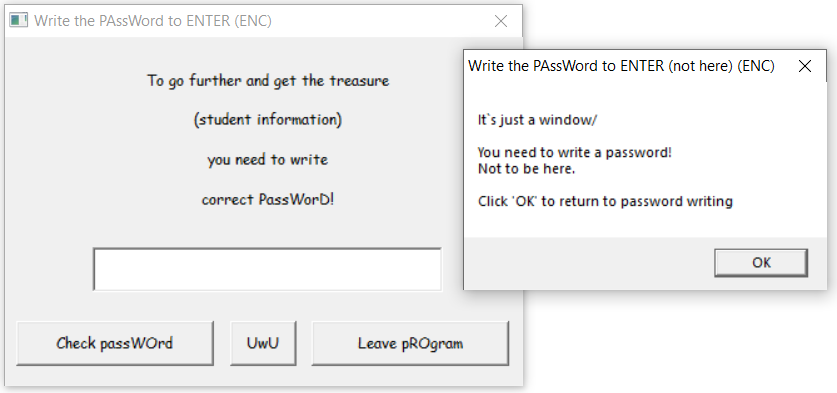
7. Введення неправильного паролю (зашифрований пароль) – використано макрос 1 для відображення вікна з повідомленням про неправильний пароль



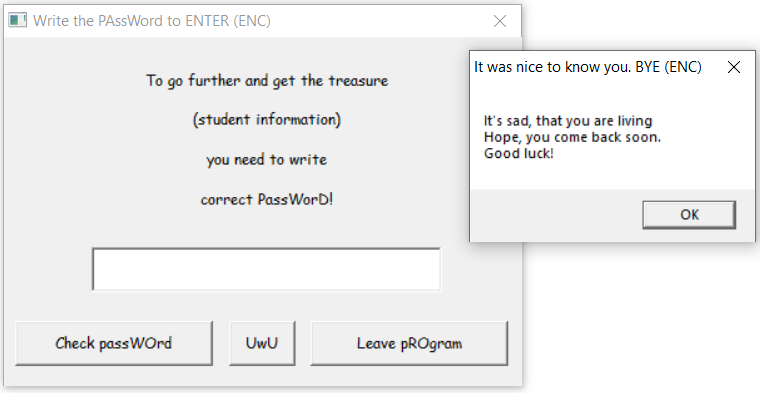
8. Введення неправильного паролю (ключ) – використано макрос 1 для відображення вікна з повідомленням про неправильний пароль



Вікно, яке з’являється після натискання кнопки “UwU” (**після закриття цього вікна вікно з введенням паролю залишиться відкритим**) ) – використано макрос 1 для відображення

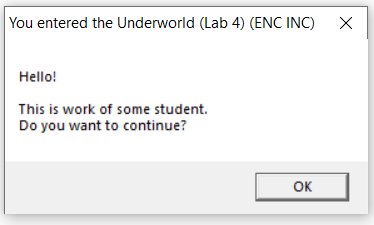


Вікно, яке з’являється після виходу з програми за допомогою кнопки “ Leave pROgram” ) – використано макрос 1 для відображення

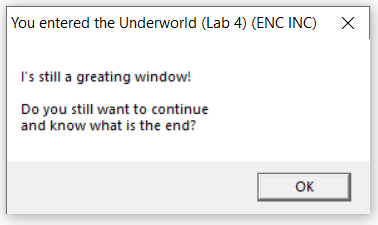


Програма, де всі перераховані в п. 2 і 3 макровизначення для макросів розташовані в окремому файлі (INC):

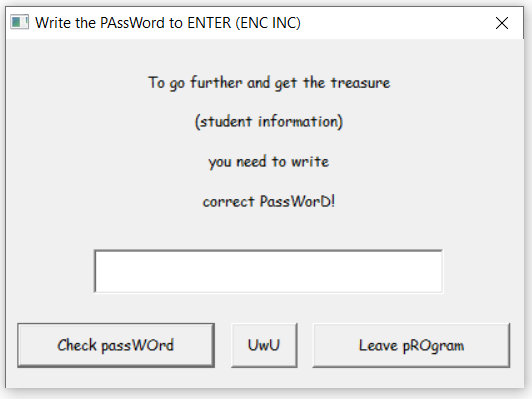
Вікно при запуску програми (прев’ю) – використано макрос 1 для відображення



Друге вікно після запуску програми (це теж прев’ю) – використано макрос 1 для відображення



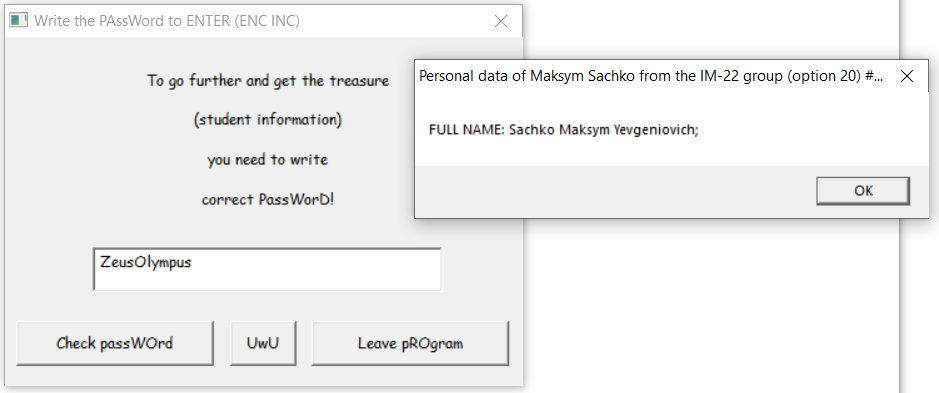
Вікно для введення паролю

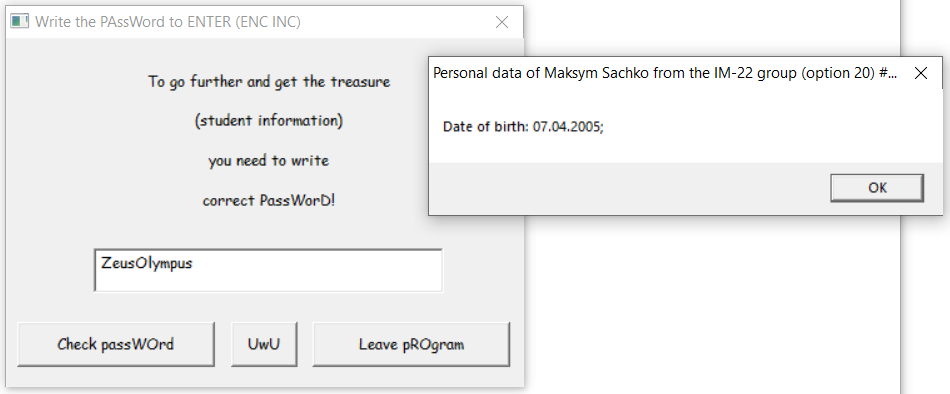


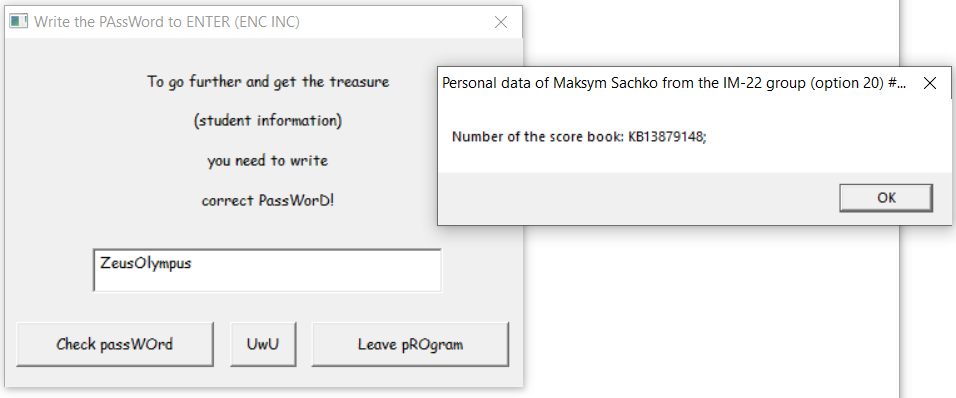
Введення паролю:

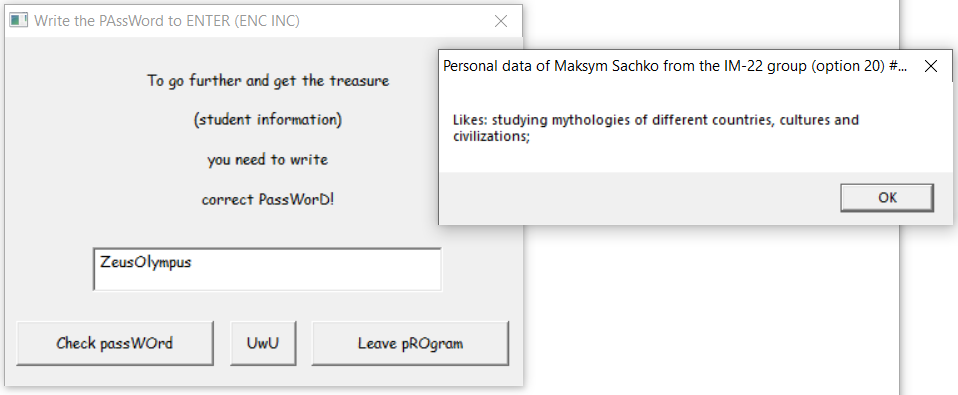
1. Введення правильного паролю

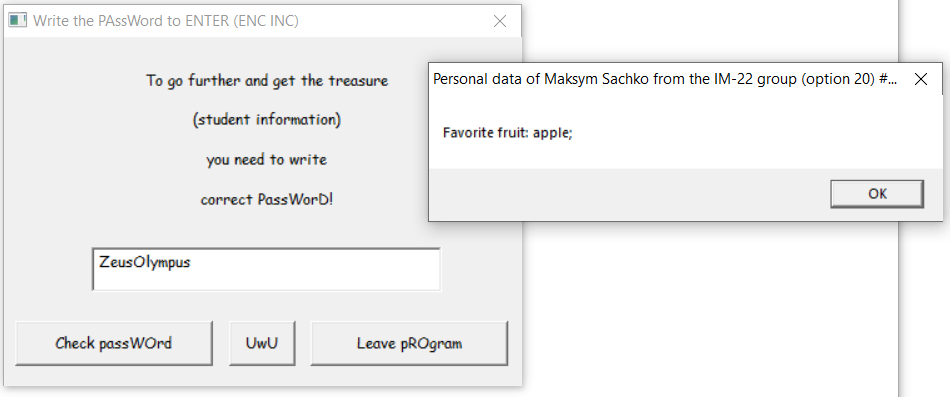
При введенні правильного паролю кожний фрагмент даних (ім’я, номер залікової книжки, дата народження і тд.) тепер виводиться в своєму вікні послідовно завдяки реалізації макроса 1, а не як у 3-й лабораторній, коли всі дані показувалися в одному вікні

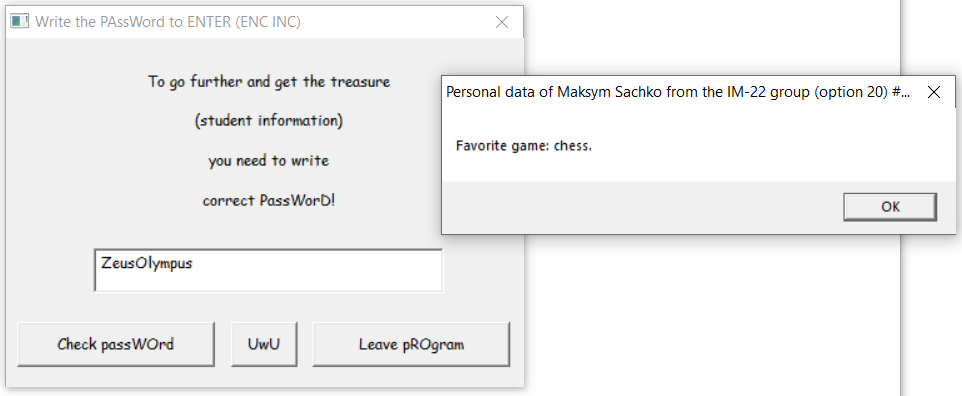




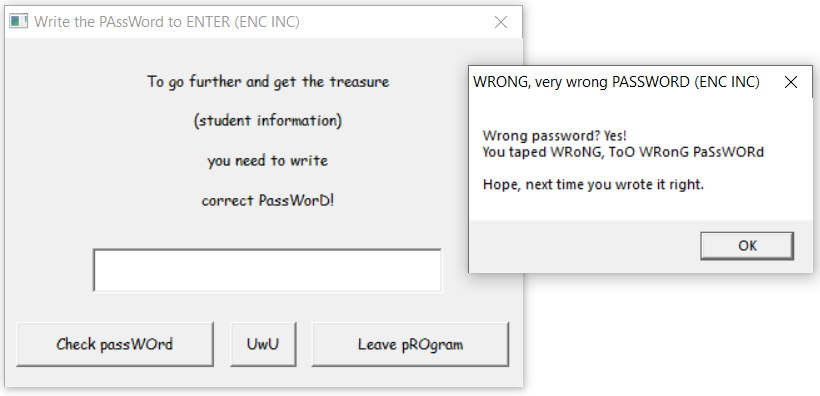




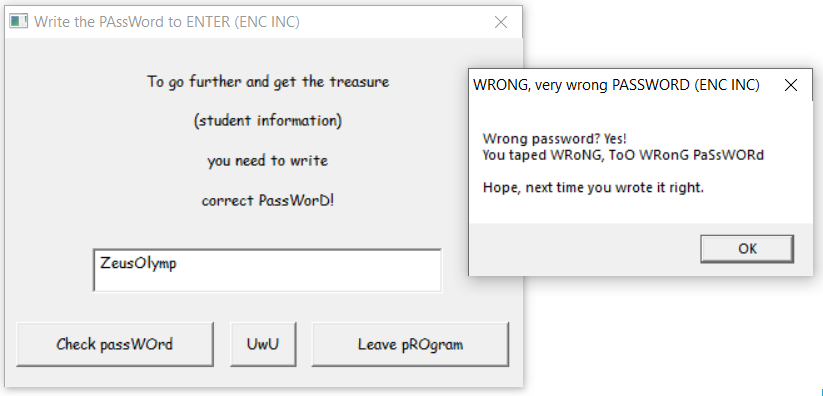




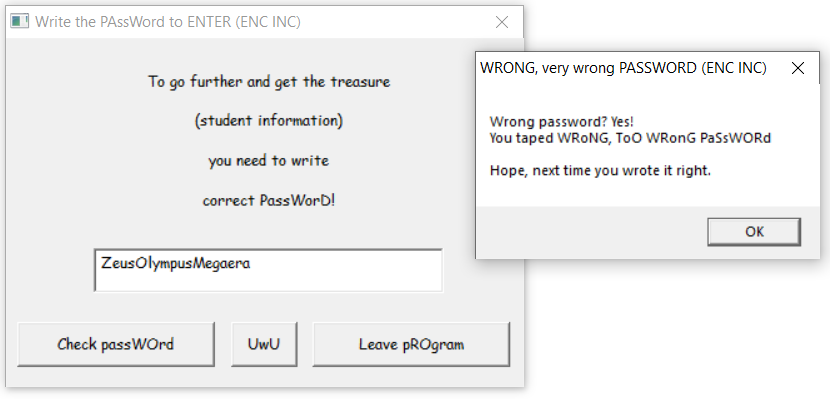
2. Введення неправильного паролю (пустий рядок) – використано макрос 1 для відображення вікна з повідомленням про неправильний пароль



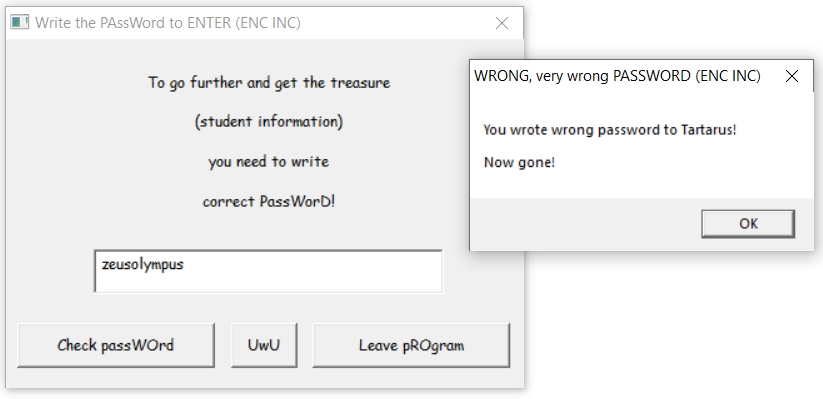
3. Введення неправильного паролю (неповний пароль) – використано макрос 1 для відображення вікна з повідомленням про неправильний пароль



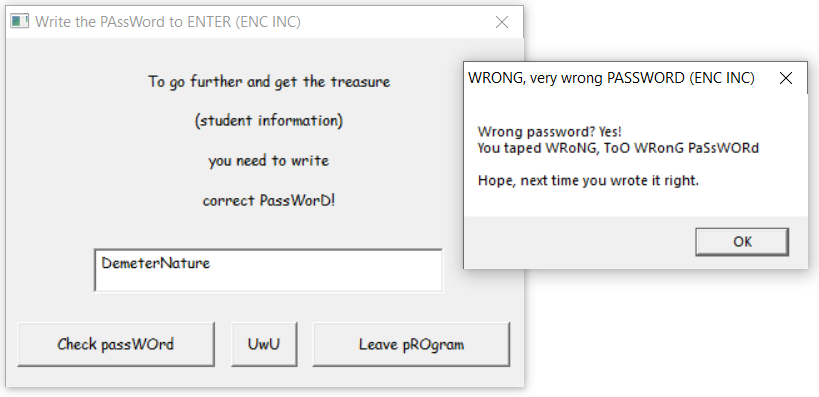
4. Введення неправильного паролю (пароль з надлишковою кількістю літер) – використано макрос 1 для відображення вікна з повідомленням про неправильний пароль



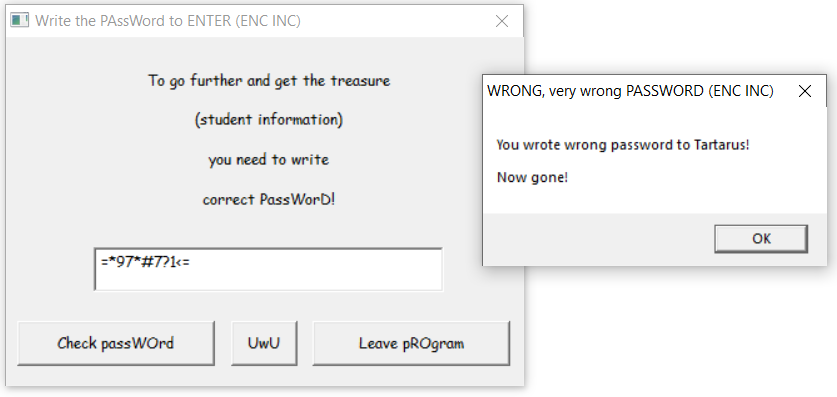
5. Введення неправильного паролю (пароль з іншим регістром деяких символів) – використано макрос 1 для відображення вікна з повідомленням про неправильний пароль



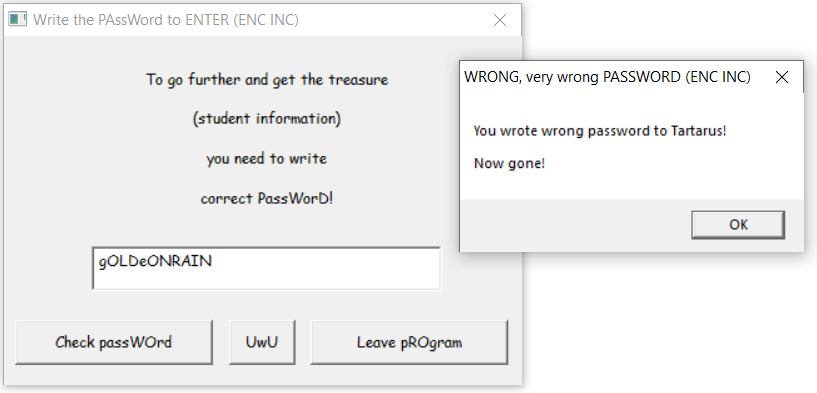
6. Введення неправильного паролю (зовсім інший пароль) – використано макрос 1 для відображення вікна з повідомленням про неправильний пароль



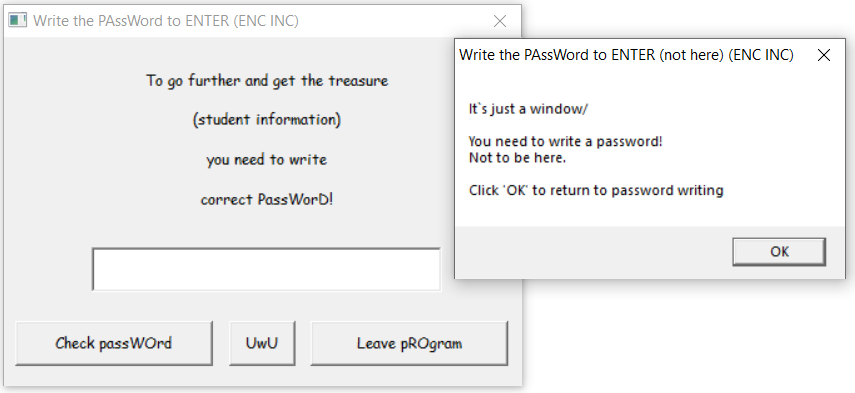
7. Введення неправильного паролю (зашифрований пароль) – використано макрос 1 для відображення вікна з повідомленням про неправильний пароль



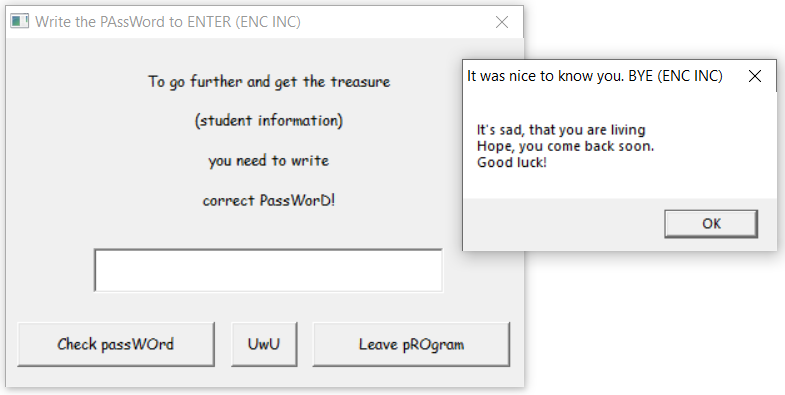
8. Введення неправильного паролю (ключ) – використано макрос 1 для відображення вікна з повідомленням про неправильний пароль



Вікно, яке з’являється після натискання кнопки “UwU” (**після закриття цього вікна вікно з введенням паролю залишиться відкритим**) ) – використано макрос 1 для відображення



Вікно, яке з’являється після виходу з програми за допомогою кнопки “ Leave pROgram” ) – використано макрос 1 для відображення



Відповідно до завдання лабораторної роботи, мною було використано три макроси:

* **SachkoHermes\_meSSeNGEfoRaLLwinDowsShOWTimE (макрос 1)** – виводить на екран повідомлення з даними у різних ситуаціях (наприклад, в разі введення правильного пароля, або ж повідомлення про те, що пароль неправильний), приймає два аргументи: текст вікна в аргументі SachkoAphrodite\_winDOwTeXtHaNdler та назва вікна в аргументі SachkoAres\_tiTLEofTHEwiNDowHanDler;
* **SachkoDionysus\_ENcryPtInGsoMePassWORds (макрос 2)** – здійснює шифрування введеного користувачем паролю;
* **SachkoArtemis\_comPaREjudgePasSwWoRdsCorrEcTInG (макрос 3)** – здійснює порівняння введеного користувачем паролю та зашифрованого правильного паролю. Також у цьому макросі викликані локальні мітки SachkoHephaestus\_coRReCtAnDcomPAREpaSsswoRdsLenTh, SachkoJudge\_MinosForCoRRecttHInG. Мітки оголошені за допомогою local.

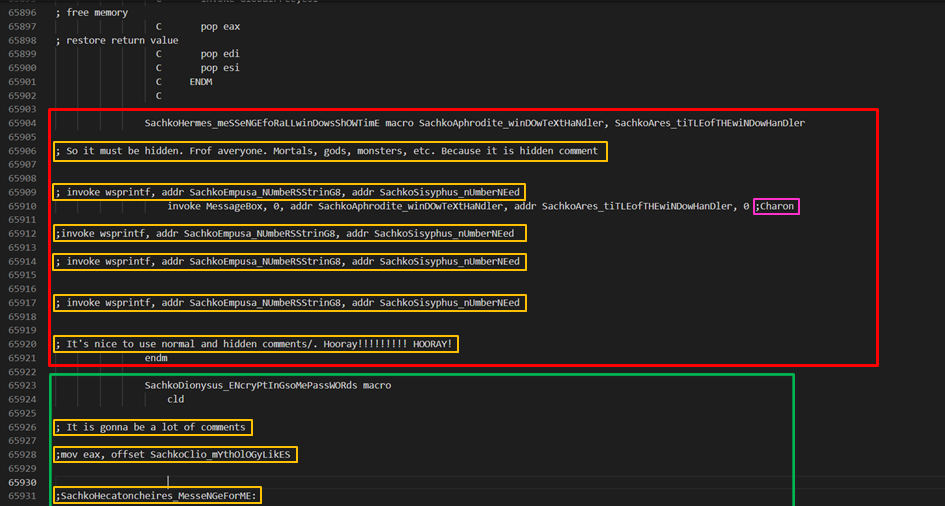
**6. Отримати розширений лістинг програми за допомогою опції /Fl компілятора ML.**

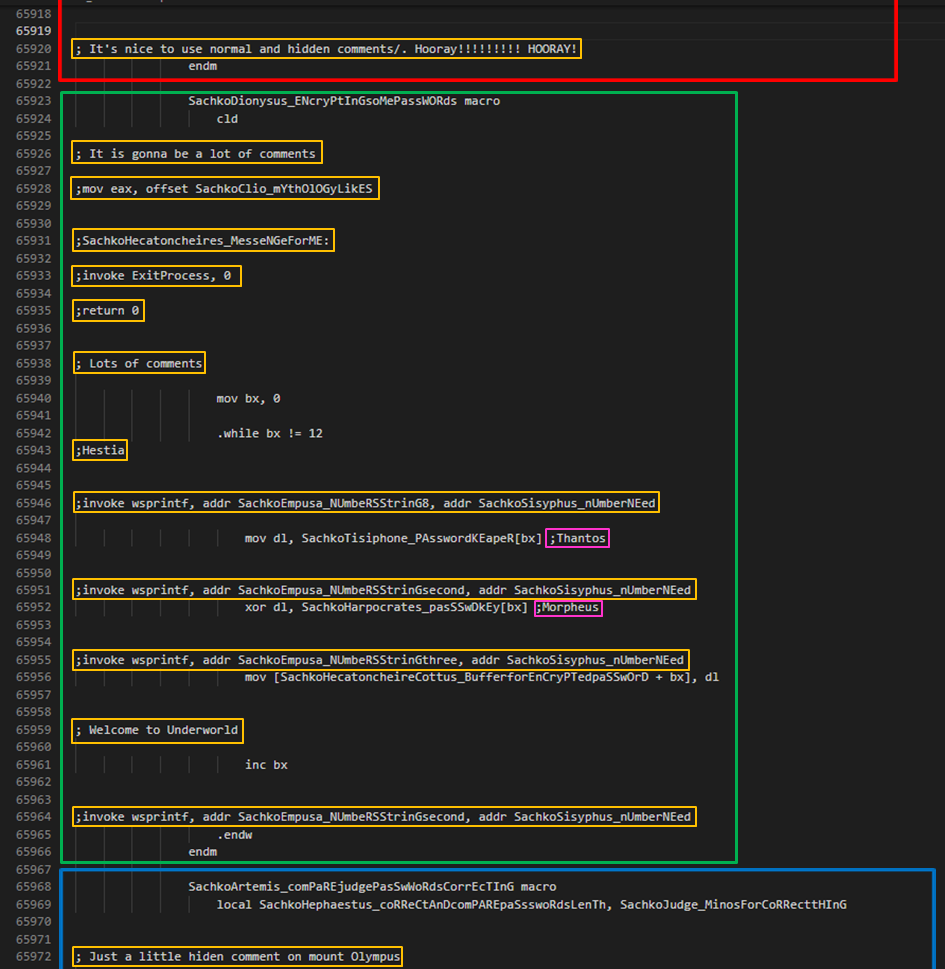
**7. Провести дослідження отриманого лістингу: визначити, яким чином компілятор виконав трансляцію кожній з команд макросів і помітити їх в тексті розширеного лістингу кольоровими олівцями або кольоровими фломастерами.**

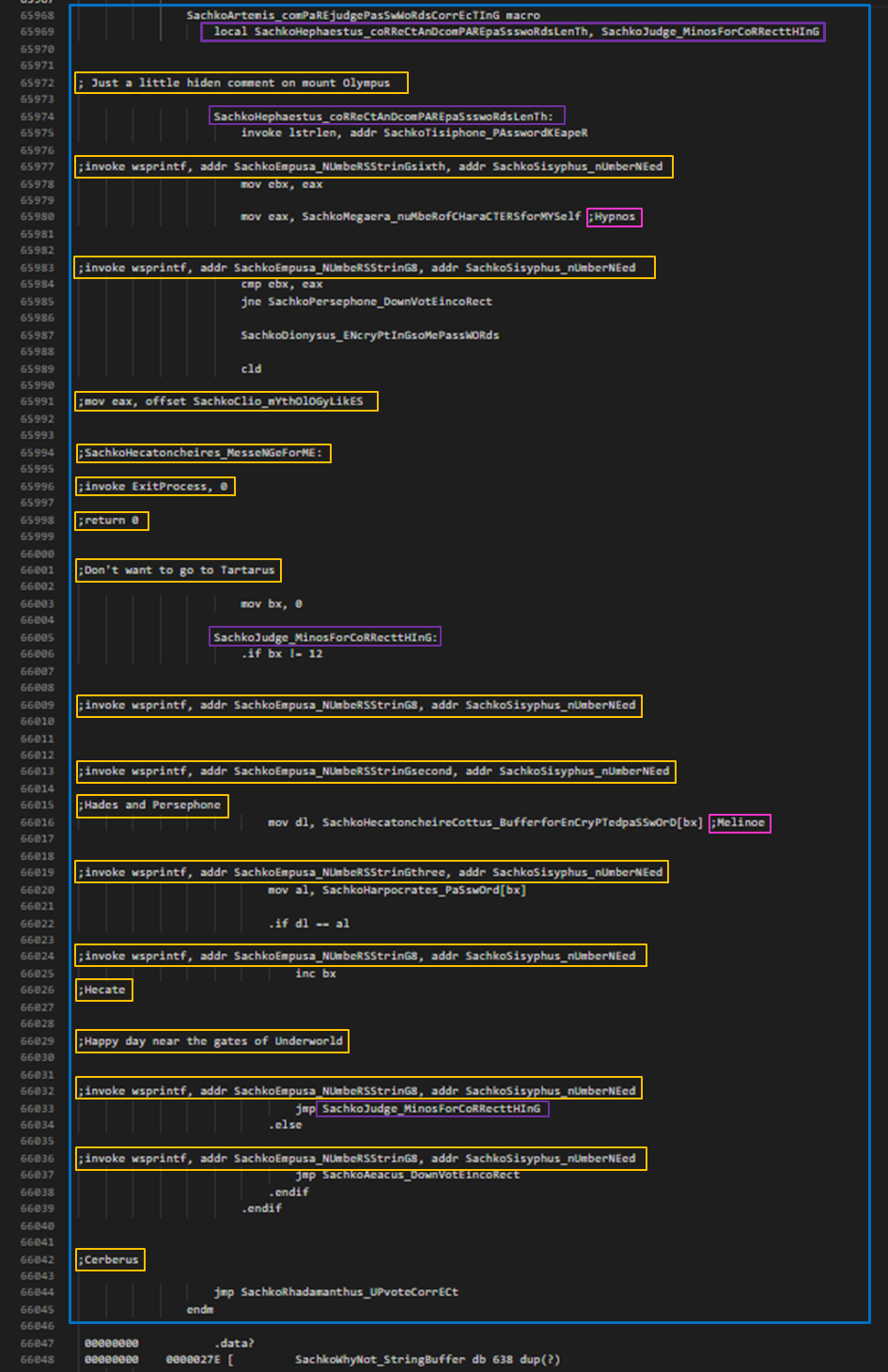
Дослідження лістингу програми з макросами в коді

**Макровизначення**

Макровизначення мною було визначено після підключення бібліотек та перед оголошенням сигнатури функції, яка обробляє поведінку вікна. Видно, що ми можемо побачити як звичайні, так і приховані коментарі.

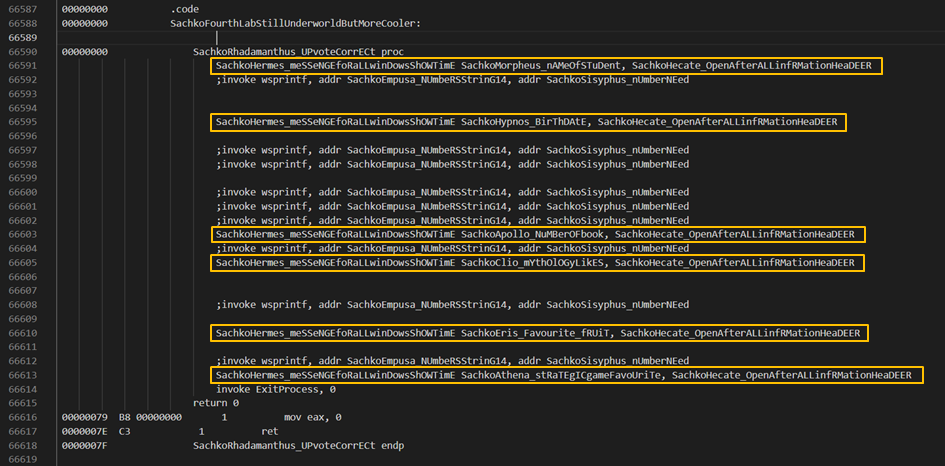


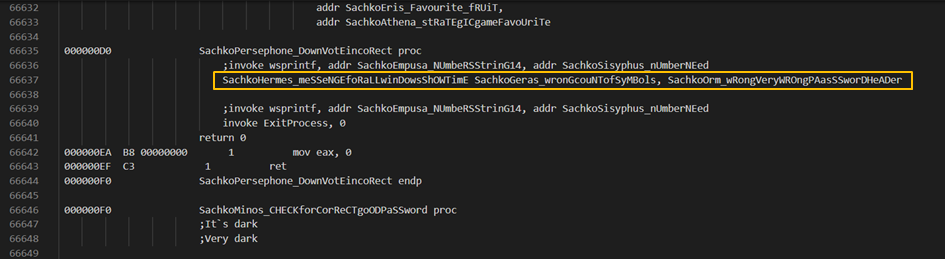


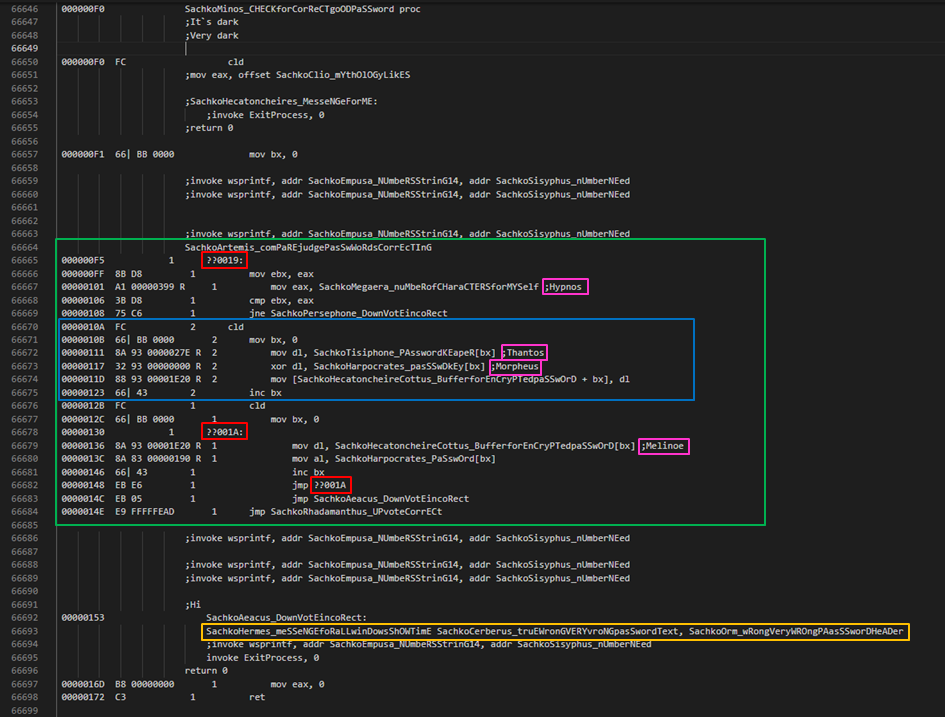


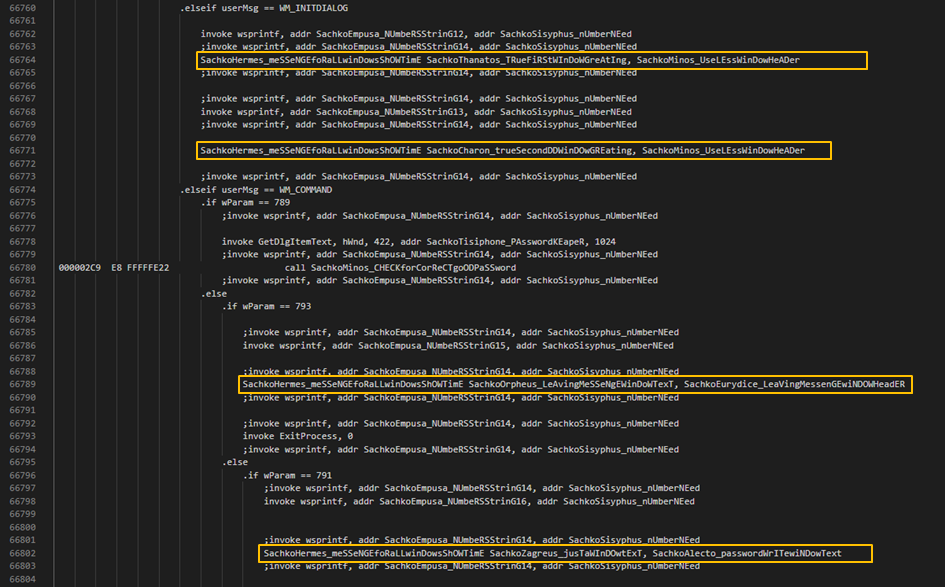
На скріншотах показано та обведено макроси: макрос 1 для виведення вікна з повідомленням (**червоним** кольором), макрос 2 для шифрування пароля (**зеленим** кольором) та макрос 3 для порівняння паролів (**синім** кольором). Варто зауважити на цьому етапі, що у макросі 3 для порівняння паролів використано локальні мітки (їх оголошення позначено **фіолетовим** кольором, як і назви міток), і пізніше буде розглянуто їхню поведінку при трансляції. Також **оранжевим** кольором обведені приховані коментарі. **Рожевим** позначені звичайні коментарі.

**Трансляції макросів**







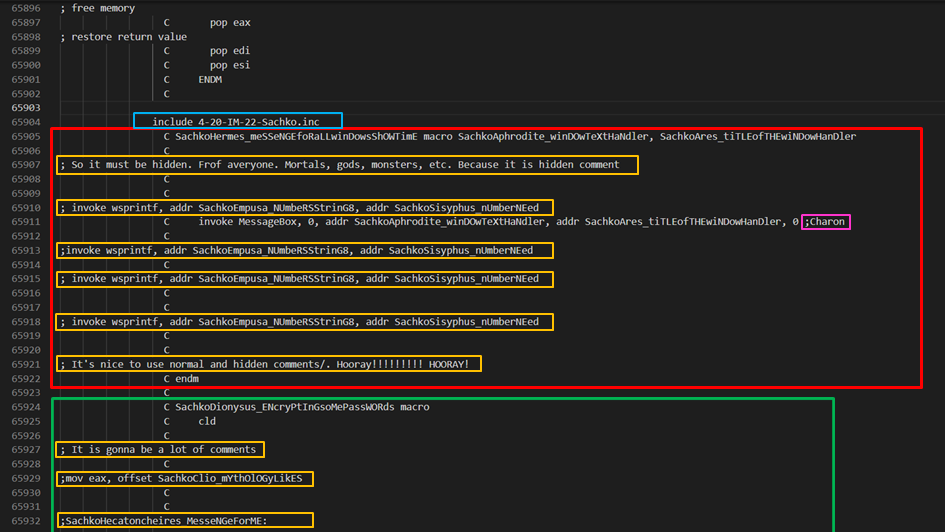


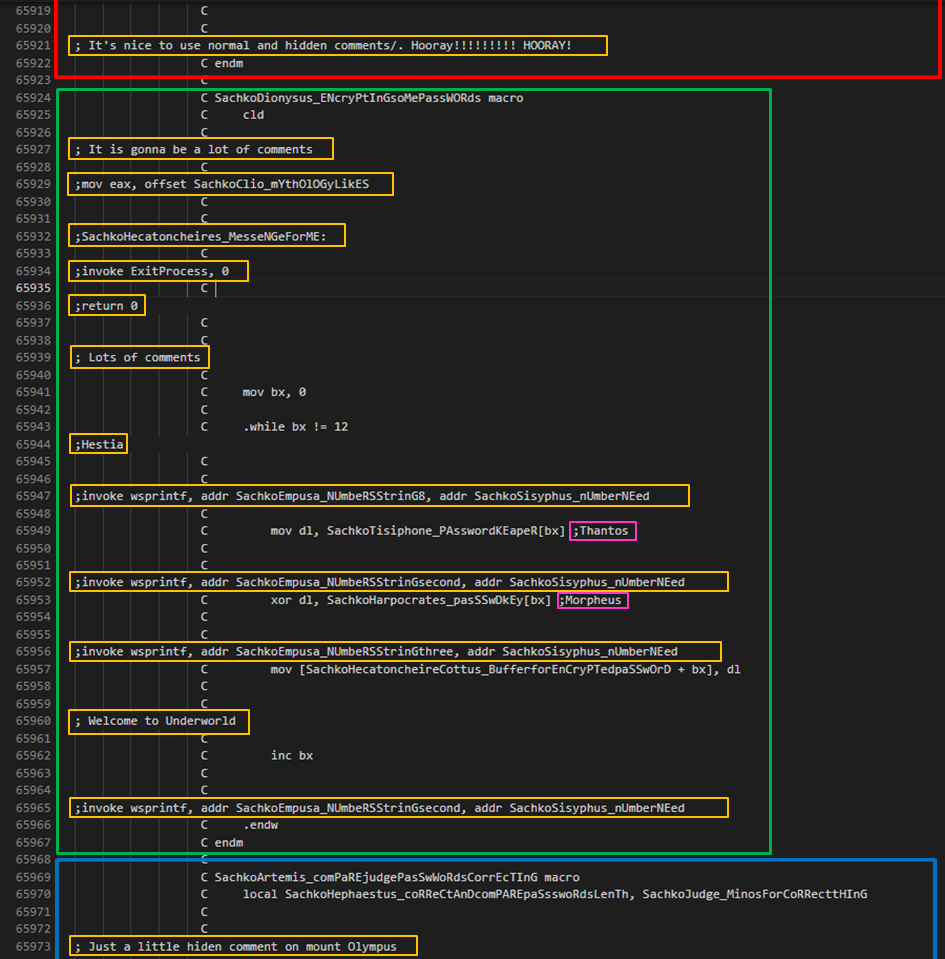
На скріншотах вище показано трансляції для всіх трьох макросів: для першого макроса (виведення повідомлення у вікно) обведено **оранжевим** кольором, для другого макроса (шифрування пароля) – **синім**, для третього макроса (порівняння паролів) – **зеленим**. Так як другий макрос викликається всередині третього, то він транслюється у цей третій макрос порівняння, а тому його назви SachkoDionysus\_ENcryPtInGsoMePassWORds не бачимо. Також примітно, що у третьому макросі замість назв локальних міток при трансляції з’явилися коди (їх обведено **червоним** кольором) і також приховано їх оголошення. Крім цього, варто зауважити, що після трансляції не видно приховані коментарі, натомість звичайні коментарі присутні (їх обведено **рожевим** кольором). Насамкінець, для першого макроса (виведення повідомлення у вікно), можна побачити, що команда invoke, що міститься у цьому макросі, у лістингу при трансляції приховано, тому при трансляції ми бачимо тільки виклик цього макроса з передачею йому аргументів.

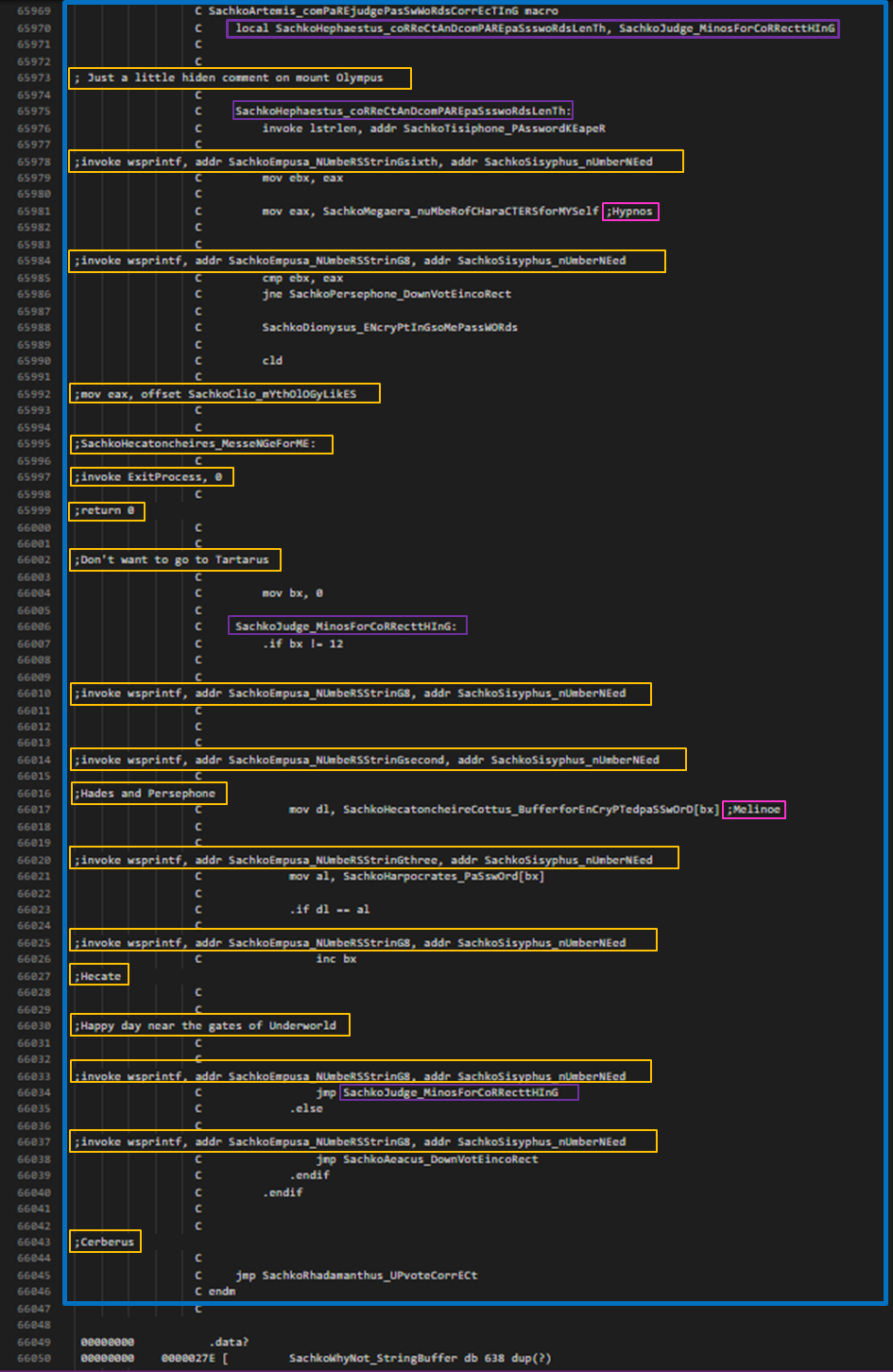
**8. Повторити дослідження, які виконані в п. 4-7 для другої версії програми, де всі перераховані в п. 2 і 3 макровизначення для макросів повинні розташовуватися в окремому файлі.**

Дослідження лістингу програми, у якій макроси записано в окремий файл, і їх імпортовано за допомогою команди include

**Макровизначення**

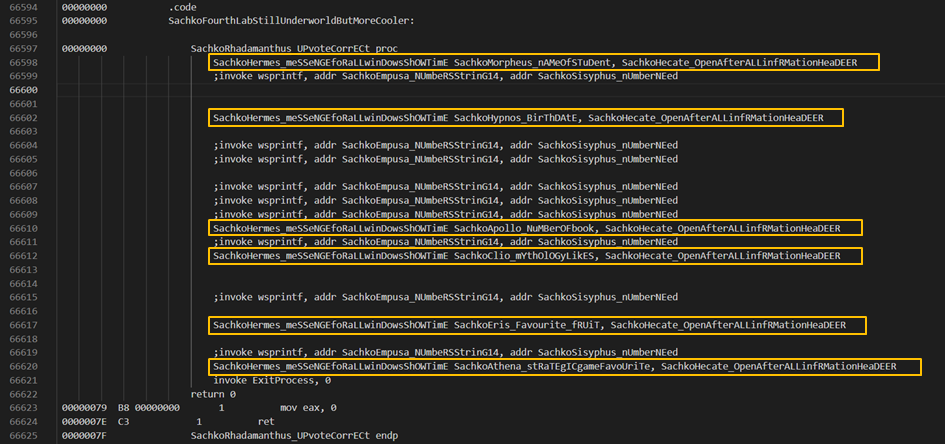


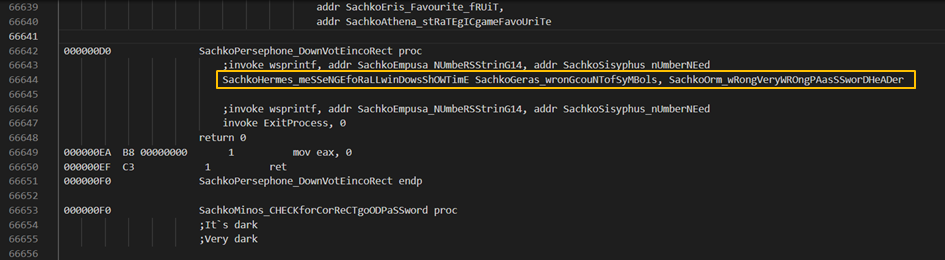


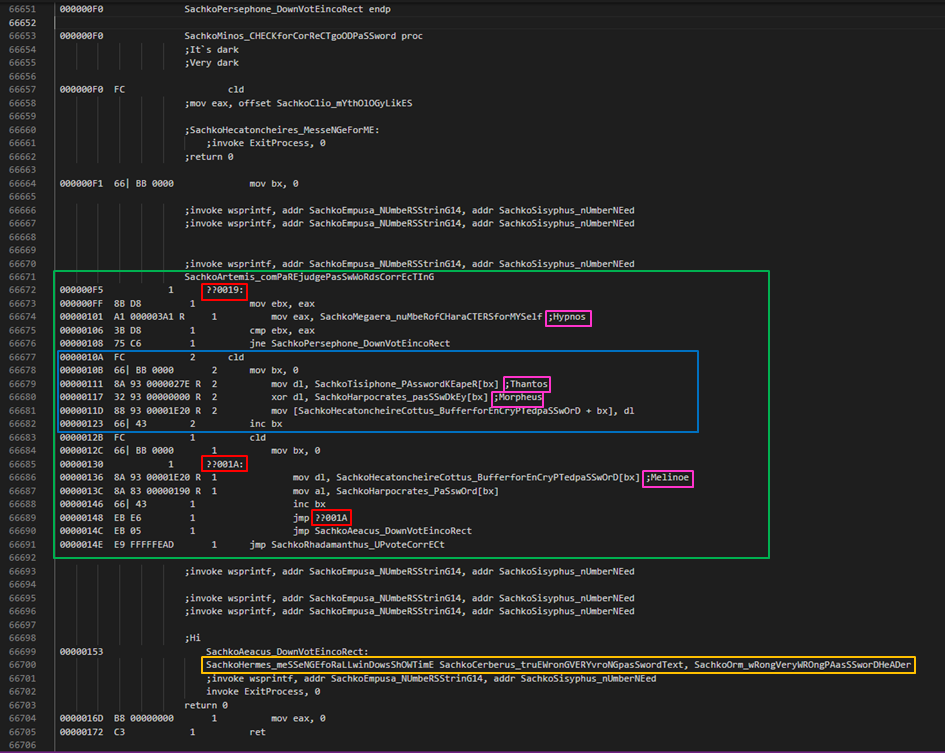


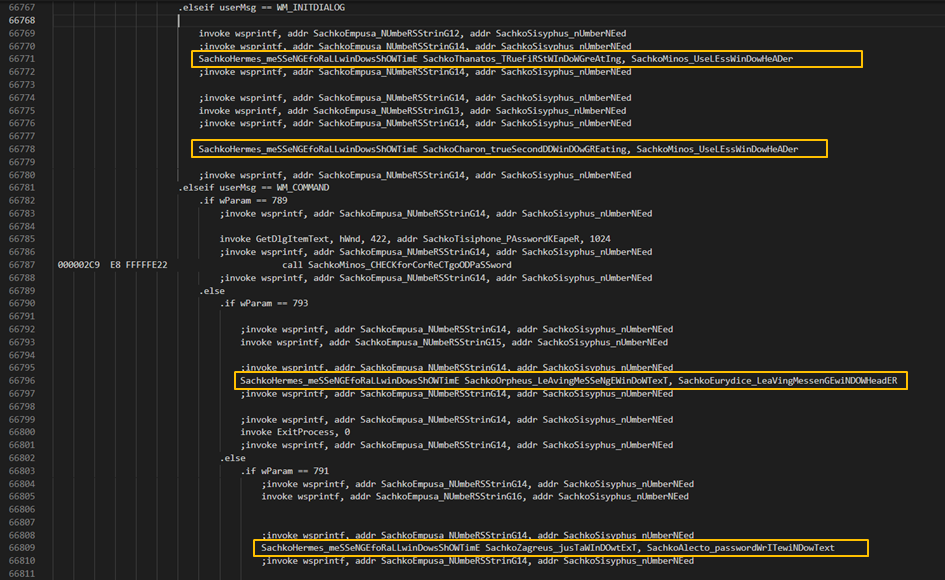
У лістингу другого варіанта програми макровизначення показуються одразу після команди include для їхнього імпорту (обведено **блакитним** кольором). На скріншотах показано та обведено макрос 1 для виведення вікна з повідомленням (**червоним** кольором), макрос 2 для шифрування пароля (**зеленим** кольором) та макрос 3 для порівняння паролів (**синім** кольором). У макросі для порівняння паролів також використано і видно локальні мітки (їх оголошення позначено **фіолетовим** кольором, як і назви міток). Також **оранжевим** кольором обведені приховані коментарі. **Рожевим** позначені звичайні коментарі. Також далі побачимо, що станеться з локальними мітками при трансляції.

**Трансляції макросів**









На скріншоті лістингу другого варіанту програми спостерігаємо загалом те саме, що і для першого варіанту. Знову ж таки, показано трансляції для всіх трьох макросів: для першого макроса (виведення повідомлення у вікно) обведено **оранжевим** кольором, для другого макроса (шифрування пароля) – **синім**, для третього макроса (порівняння паролів) – **зеленим**. Також, так як другий макрос викликається всередині третього, то він транслюється у цей третій макрос порівняння, а тому його назви SachkoDionysus\_ENcryPtInGsoMePassWORds не бачимо. Також примітно, що у третьому макросі замість назв локальних міток при трансляції з’явилися коди (їх обведено **червоним** кольором) і також приховано їх оголошення. Після трансляції, як і в першому варіанті програми, не видно приховані коментарі, натомість звичайні коментарі присутні (їх обведено **рожевим** кольором). Насамкінець, для першого макроса (виведення повідомлення у вікно), можна побачити, що команда invoke, що міститься у цьому макросі, у лістингу при трансляції приховано, тому при трансляції ми бачимо тільки виклик цього макроса з передачею йому аргументів.

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

Порівнюючи два варіанти програми (з визначенням макросів в одному файлі разом з рештою коду та визначенням макросів у окремому файлі з подальшим його імпортом у головний файл), можу визначити деякі схожості та відмінності між ними.

*Що в них схожого:*

* у макровизначеннях в обох варіантах видно як звичайні, так і приховані коментарі;
* при трансляції в обох варіантах приховані коментарі більше не видно;
* назви локальних міток замінюються при трансляції на унікальні коди в обох варіантах програми;
* рядки з викликом функцій за допомогою команди invoke при трансляції в обох варіантах приховано – це якраз добре видно на прикладі першого макроса, тому при трансляції залишається тільки рядок з викликом цього макроса із вказанням аргументів;
* у трансляціях макросів їхній код вставляється одразу під їхнім викликом.

*Що в них відмінного:*

* єдиною ключовою відмінністю є розташування макровизначень у файлах лістингу.

**10. Зробити висновки по лабораторній роботі.**

**Висновок:** У ході виконання даної лабораторної роботи я дізнався, що таке макроси в мові асемблера MASM32, і як їх використовувати, а також як їх можна різними шляхами додавати до своєї програми: або прописуючи у самому файлі програми разом з рештою коду, або ж прописати їх в окремому файлі, який включити до головного файлу за допомогою команди include. Також я познайомився з локальними мітками, назви яких при трансляції у лістингу замінюються на унікальні коди.

Після порівняння обох варіантів програм (у пункті 9), можна стверджувати, що в обох варіантах програми трансляція макросів однакова, єдина відмінність полягає в розташуванні макровизначень у розширеному лістингу.

Насамкінець можу сказати, що макроси в MASM32 – це невід'ємний інструмент для розробників, який значно спрощує створення блоків коду, які можна використовувати багаторазово. Вони дозволяють ефективно прискорити процес розробки, замінюючи довгі та складні фрагменти коду одним простим і зрозумілим викликом макросу.