



МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра Інформаційної Безпеки

Захист програмного забезпечення

Лабораторна робота 2

Захист додатків від несанкціонованого використання і копіювання

Мета роботи: оволодіти навиками захисту програмного забезпечення від несанкціонованого використання і копіювання.

Перевірив:

Виконав:

студент III курсу

групи ФБ-01

Сахній Н.Р.

Київ 2023

ФБ-01 Сахній Назар

Варіант №11:

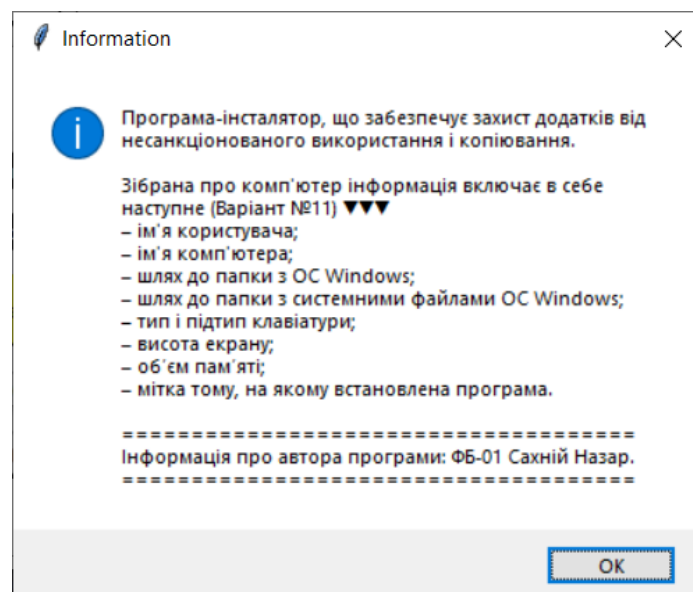
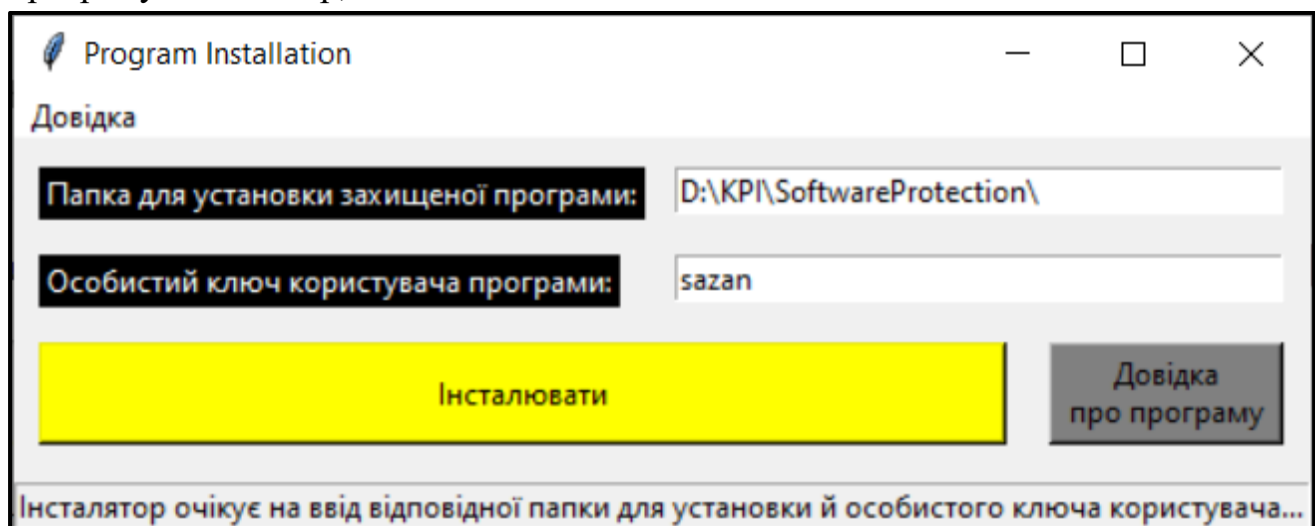
Індивідуальні варіанти завдань

Таблиця 1 - Інформація, що збирається про комп'ютер

№	Тип і підтип клавіатури	К-ть кнопок миши	Ширина екрану	Висота екрану	Набір дискових пристроїв	Об'єм пам'яті	Дані щодо диску, на якому встанов- лена програма
11	Да	Ні	Ні	Да	Ні	Да	Мітка тому

Завдання:

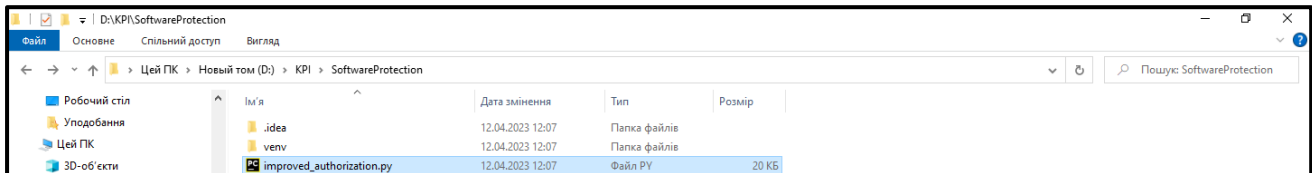
1. Для програми, розробленої при виконанні лабораторної роботи № 1, написати програму-інсталятор, яка



- запитує у користувача папку для установки захищеної програми,

```
# Create a label and entry widget for the installation directory
install_dir_label = tk.Label(root, text="Папка для установки захищеної програми:", background="black", foreground="white")
install_dir_label.place(x=10, y=10)
install_dir = tk.StringVar()
install_dir_entry = tk.Entry(root, width=40)
install_dir_entry.place(x=265, y=10)
```

- записує туди файл з дійсним кодом програми,



```
# Write 'improved_authorization.py' file to the specified directory
src_path = os.path.join(os.getcwd(), "improved_authorization.py")
dst_path = os.path.join(install_dir, "improved_authorization.py")
os.makedirs(install_dir, exist_ok=True)
os.system(f'copy "{src_path}" "{dst_path}"')
```

- збирає інформацію про комп'ютер, на якому встановлюється програма,

```
# Get user and system information
username = os.environ["USERNAME"] # `t-1000` (echo %USERNAME%)
computername = os.environ["COMPUTERNAME"] # `DESKTOP-DRI0PBB` (echo %COMPUTERNAME%)
windows_path = os.environ["WINDIR"] # `C:\Windows` (echo %WINDIR%)
system_path = os.environ["SYSTEMROOT"] # `C:\Windows` (echo %SYSTEMROOT%)

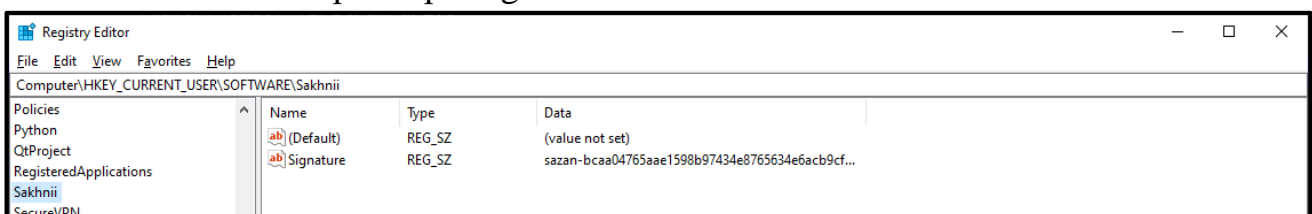
# Get information about peripheral equipment
user32 = ctypes.windll.user32
keyboard_type = user32.GetKeyboardType(0) # `7`: USB keyboard
keyboard_subtype = user32.GetKeyboardType(1) # `0`: NONE (Subtype of the keyboard is not defined)
screen_height = user32.GetSystemMetrics(1) # `864`: The height of the primary display monitor in pixels

# Get some information about the corresponding disk
install_dir = install_dir_entry.get()
memory_size = psutil.common.bytes2human(psutil.disk_usage(install_dir).total) # `!` (476.36 or 931.56)
program_drive = os.path.splitdrive(install_dir)[0] # `!` (C: or D:)
```

- хешує цю інформацію,

```
# Create a hash of user and system information
data = f"{username}-{computername}-{windows_path}-{system_path}" \
      f"{keyboard_type}-{keyboard_subtype}-{screen_height}" \
      f"{memory_size}-{program_drive}"
data_hash = hashlib.sha256(data.encode()).hexdigest()
```

- підписує її особистим ключем користувача програми і записує до реєстру Windows в розділ **HKEY_CURRENT_USER\Software\Прізвище_студента** як значення параметра Signature.



```
# Get user signature from input
signature = signature_entry.get()

# Write the hash and signature to the registry
key = winreg.OpenKey(winreg.HKEY_CURRENT_USER, "Software")
app_key = winreg.CreateKey(key, "Sakhnii")
winreg.SetValueEx(app_key, "Signature", 0, winreg.REG_SZ, f"{signature}-{data_hash}")
```

2. Зібрана про комп'ютер інформація включає в себе:

- ім'я користувача;

```
username = os.environ["USERNAME"] # `t-1000` (echo %USERNAME%)
```

- ім'я комп'ютера;

```
computername = os.environ["COMPUTERNAME"] # `DESKTOP-DRI0PBB` (echo %COMPUTERNAME%)
```

- шлях до папки з ОС Windows;

```
windows_path = os.environ["WINDIR"] # `C:\Windows` (echo %WINDIR%)
```

- шлях до папки з системними файлами ОС Windows;

```
system_path = os.environ["SYSTEMROOT"] # `C:\Windows` (echo %SYSTEMROOT%)
```

- тип і підтип клавіатури;

```
user32 = ctypes.windll.user32
keyboard_type = user32.GetKeyboardType(0) # `7`: USB keyboard
keyboard_subtype = user32.GetKeyboardType(1) # `0`: NONE (Subtype of the keyboard is not defined)
```

- висота екрану;

```
screen_height = user32.GetSystemMetrics(1) # `864`: The height of the primary display monitor in pixels
```

- об'єм пам'яті;

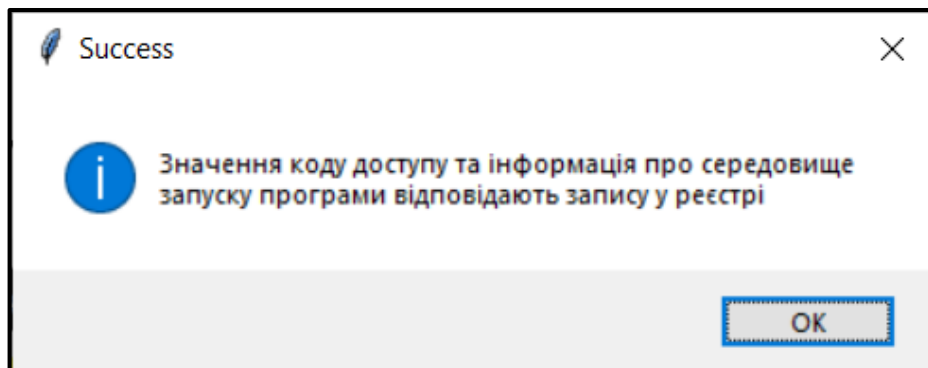
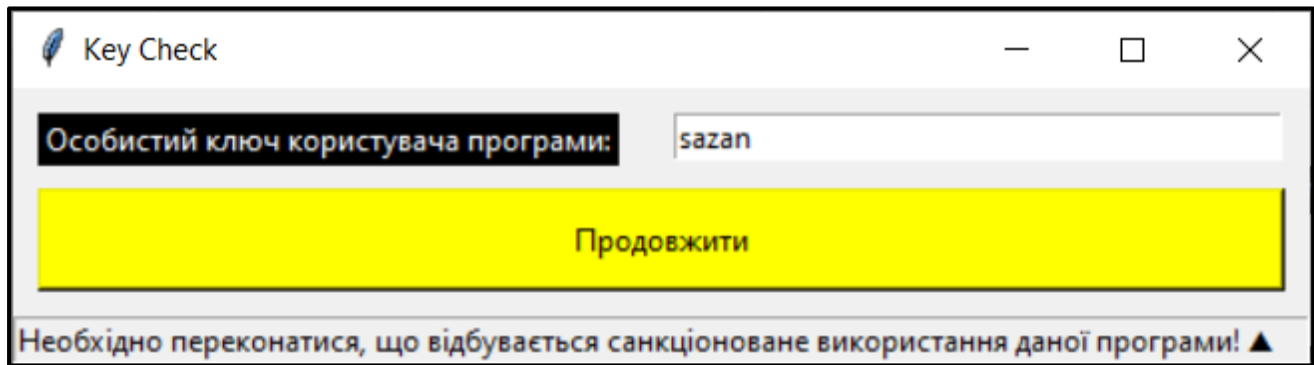
```
memory_size = psutil._common.bytes2human(psutil.disk_usage(install_dir).total) # `!` (476.36 or 931.56)
```

- мітка тому, на якому встановлена програма.

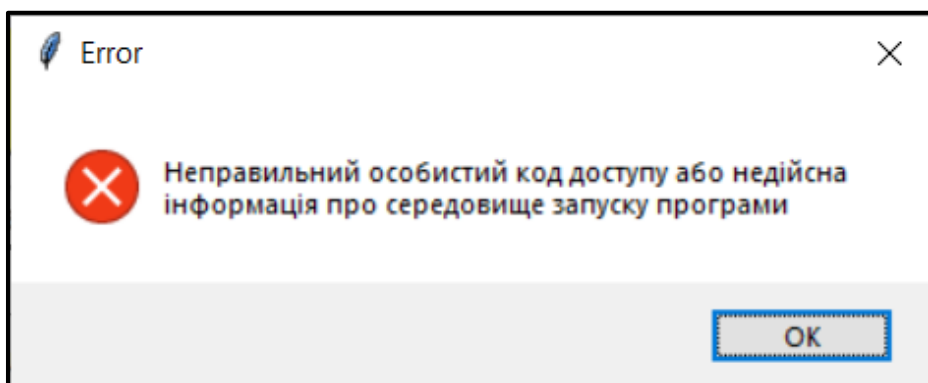
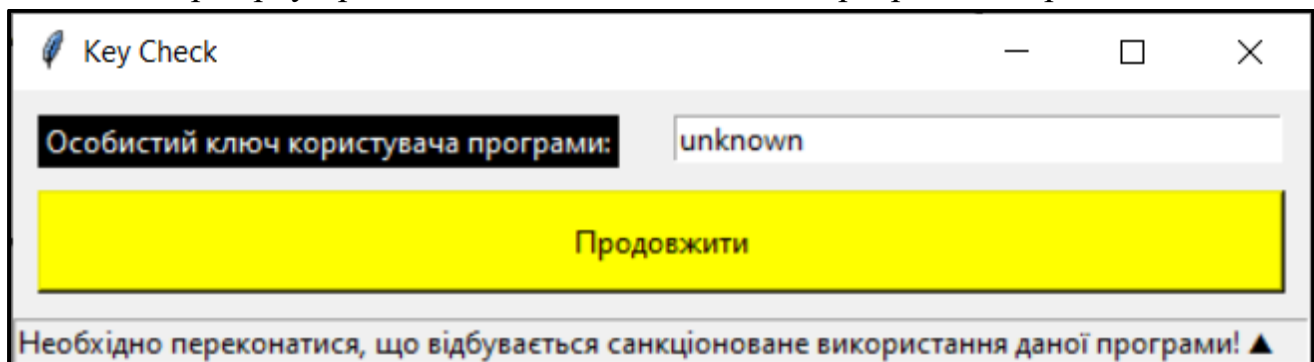
```
program_drive = os.path.splitdrive(install_dir)[0] # `!` (C: or D:)
```

3. При спробі виконати програмний застосунок, перевіряється відповідність середовища виконання і інформації, що записано в реєстрі шляхом збору інформації про комп'ютер, її хешування і порівнянню хешей. При невдалій перевірці робота програми, що захищається повинна завершуватися з видачею відповідного повідомлення.

- Якщо перевірку пройти вдало, то отримаємо доступ до програми



- Якщо перевірку пройти не вдало, то виконання програми завершиться



* Відповідний фрагмент коду, що забезпечує перевірку коду доступу та середовища виконання

```

if value == f"{signature}-{data_hash}":
    messagebox.showinfo("Success", "Значення коду доступу та інформація про середовище запуску програми відповідають запису у реєстрі")
    sign_in(root)
    return True
else:
    messagebox.showerror("Error", "Неправильний особистий код доступу або недійсна інформація про середовище запуску програми")
    root.destroy()
    return False

```