Skript systému pro rozpoznání řeči a identifikace nesprávného zpětného potvrzení

```
import csv
import re
import threading
import queue

import speech_recognition as sr
import whisper
from pynput import keyboard



audio_data_queue = queue.Queue()


r = sr.Recognizer()
active_source = None


pressed = False



def on_press(key):
    global pressed
    if not pressed and key == keyboard.Key.f1:  # only if key is not held
        print('Key %s pressed' % key)
        pressed = True  # key is held
        active_source.CHUNK = 0



def on_release(key):
    global pressed
    if key == keyboard.Key.f1:
```

```
        print('Key %s released' %key)
        pressed = False  # key is released


def listen_to_audio():
    # initialize the recognition
    microphone_1 = sr.Microphone
    microphone_2 = sr.Microphone

    with microphone_1(device_index=0) as source_1, microphone_2(device_index=0) as source_2:
        print("Adjusting for ambient noise...")
        r.adjust_for_ambient_noise(source_1)
        r.pause_threshold = 2
        r.dynamic_energy_ratio = 4
        print("Adjusted.")

        global active_source
        active_source = source_1
        idx = 0

        while True:
            print(f"[{idx}] [Source {1 if active_source == source_1 else 2}] You can speak now:")
            audio_data = r.record(active_source)
            active_source.CHUNK = 1024
            audio_data.get_raw_data()
            audio_data_queue.put(audio_data)

            active_source = source_2 if active_source == source_1 else source_1
            idx += 1
```

```
def convert_to_text():
    idx = 0
    prev_text = None

    while True:
        audio_data = audio_data_queue.get(timeout=60)

        print(f"Recognizing {idx}. recording...")
        text = r.recognize_whisper(audio_data)

        if len(text) < 10:
            continue
        if text == "thank you.":
            continue

        print(text)
        with open("TextKeeper.csv", "a+", encoding="UTF8", newline="\n") as file:
            writer = csv.writer(file)
            writer.writerow([str(idx)])
            writer.writerow(text)

        idx += 1

        if idx % 2 == 0:
            numbers = re.findall(r'\d+', text)
            prev_numbers = re.findall(r'\d+', prev_text)
```

```
            not_numbers = ''.join(re.findall(r'[^\d+]', text, re.IGNORECASE))
            prev_not_numbers = ''.join(re.findall(r'[^\d+]', prev_text, re.IGNORECASE))

            Text_compare = not_numbers == prev_not_numbers
            numbers_compare = numbers == prev_numbers

            print(numbers_compare)
            print(Text_compare)

            with open("TextKeeper.csv", "a+", encoding="UTF8", newline="\n") as file:
                writer = csv.writer(file)
                writer.writerow([str(numbers_compare)])
                writer.writerow([str(Text_compare)])

        else:
            prev_text = text


def main():
    model = whisper.load_model("base.en")

    # Initialize keyboard listener in order to be able to switch input sources.
    # It's just a thread listening for keyboard strokes.
    listener = keyboard.Listener(on_press=on_press, on_release=on_release)
    listener.start()

    # Background thread converting audio to text. It waits for the audio data to
    # appear in a shared queue.
    thread = threading.Thread(target=convert_to_text, args=())
```

```
    thread.start()

    # Main thread, which listens to audio and puts it into a shared queue.
    listen_to_audio()

    # Wait for both background threads to finish.
    listener.stop()
    thread.join(timeout=10)


if __name__== "_main_":
    main()
```