Instructions

1. Please visit the following web page and sign up for an API key. Be patient, it may take a day or two to get approved. Keep checking your junk mail folder too:
    1. https://api.winnipegtransit.com
2. Once you have an API key, navigate to the api overview page and familiarize yourself with the api. Pay particular attention to the JSON section:
    1. https://api.winnipegtransit.com/home/api/v3
3. Visit the following web page and become familiar with how to use the API:
    1. https://api.winnipegtransit.com/home/api/v3/example
4. Visit the following web page and become familiar with how to work with JSON based web services (don't do the project, just read it). This will introduce you to some basic Python structures like lists and dictionaries and how to fetch json data:
    1. https://projects.raspberrypi.org/en/projects/where-is-the-space-station/0
5. Write a simple terminal app that lists all the bus stops within a certain radius of a set of GPS coordinates you supply. The user should be able to choose one bus stop which then lists all the scheduled and estimated arrival times. See instructor demo for an example of what this should look like.
6. Import all the required libraries for this assignment at the top of your file:

```
from requests import get
```

7. You can find GPS coordinates on Google Maps. If you search for a specific address, you can either right click on the location "pin" and see the coordinates, or look in the address bar for the coordinates.
8. You can choose how big a radius you'd like to use. Start off with 100m for testing purposes.
9. You can fetch a URL and retrieve the data like this, which will return a native Python dictionary, don't forget to paste the URL in your browser to examine what it returns:

```
API_KEY = “paste_your_api_key_string_here”
lon = -97.138  # GPS longitude of location
lat = 49.895  # GPS latitude of location
distance = 100  # radius in meters to search around GPS coordinates

# url to request stops
url_stops = f"https://api.winnipegtransit.com/v3/stops.json?lon={lon}&lat={lat}&distance={distance}&api-key={API_KEY}"

# request bus stops nearby
resp_stops = get(url_stops).json()
```

Bonus

Convert the timestamp strings returned from the API to datetime objects and then just display the properly formatted time on the terminal, instead of the whole timestamp. Here is an example:

```
from dateutil.parser import parse
my_datetime_obj = parse(my_timestamp)
print(my_datetime_obj.strftime("%H:%M:%S"))
```

Color code the output. Compare the scheduled arrival time to the estimated arrival time. If the bus will be on time, color the times green. If it will be late, color the times red. If it will be early, color the times blue. You can find how to do that here:

https://pypi.org/project/colorama/

For example:

```
from colorama import just_fix_windows_console, Fore, Style
just_fix_windows_console()
print(Fore.RED + 'some red text')
print(Back.GREEN + 'and with a green background')
print(Style.DIM + 'and in dim text')
print(Style.RESET_ALL)
print('back to normal now')
```