What’s the Hottest City?

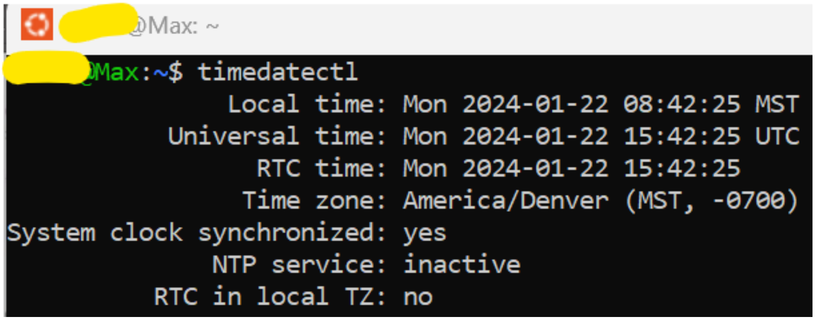
- **Note**: This project is provided by **Codecademy** (https://www.codecademy.com)
- **Note**: I did this project in the **Ubuntu** terminal.

Project description

The purpose of the project is to determine the hottest city. I will:

- Fetch forecast temperatures for multiple cities.
- Save these temperatures to a file.
- Sort the cities based on their temperature.

Start Project Timestamp

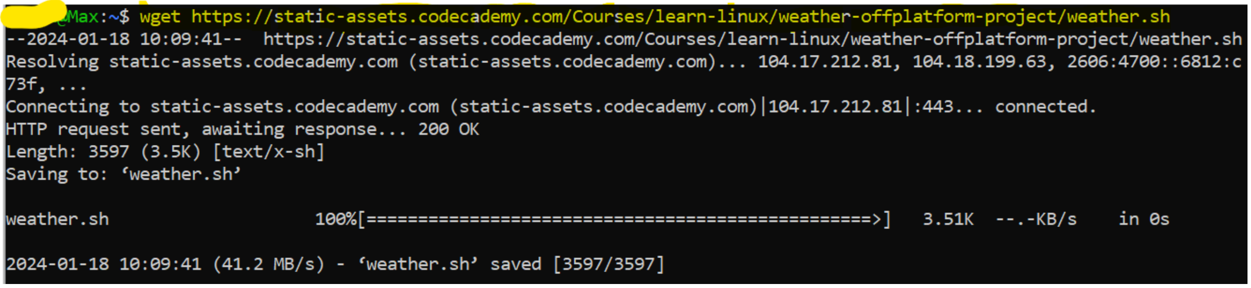


[redacted]@Max: ~

```
[redacted]@Max:~$ timedatectl
               Local time: Mon 2024-01-22 08:42:25 MST
           Universal time: Mon 2024-01-22 15:42:25 UTC
                 RTC time: Mon 2024-01-22 15:42:25
                Time zone: America/Denver (MST, -0700)
System clock synchronized: yes
              NTP service: inactive
          RTC in local TZ: no
```

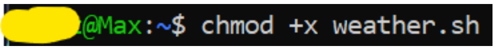
Project Setup

1. I first downloaded a modified **weather** script offered by Codecademy. This script is a modification from **Bash-Snippets** at https://github.com/alexanderepstein/Bash-Snippets. The command (highlighted in yellow) that I used for downloading the Codecademy version is shown below:
    - **Optional**: Installing the original **weather** from **Bash-Snippets is optional**. However, it is recommended if you are familiar with **git** and if you wish to further improve on the script. The API link for this project is https://wttr.in/.



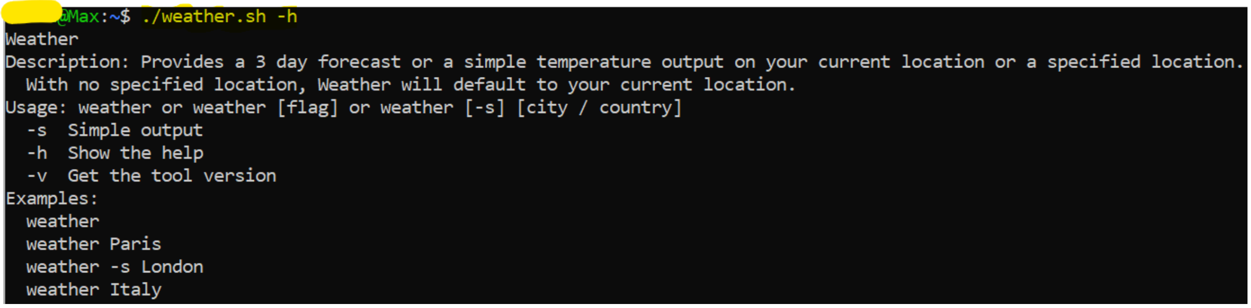
```
[redacted]@Max:~$ wget https://static-assets.codecademy.com/Courses/learn-linux/weather-offplatform-project/weather.sh
--2024-01-18 10:09:41--  https://static-assets.codecademy.com/Courses/learn-linux/weather-offplatform-project/weather.sh
Resolving static-assets.codecademy.com (static-assets.codecademy.com)... 104.17.212.81, 104.18.199.63, 2606:4700::6812:c
73f, ...
Connecting to static-assets.codecademy.com (static-assets.codecademy.com)|104.17.212.81|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3597 (3.5K) [text/x-sh]
Saving to: ‘weather.sh’

weather.sh                    100%[=================================================>]   3.51K  --.-KB/s    in 0s

2024-01-18 10:09:41 (41.2 MB/s) - ‘weather.sh’ saved [3597/3597]
```

2. I then gave the script executable permissions with the following command.



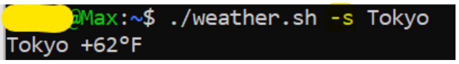
```
[redacted]@Max:~$ chmod +x weather.sh
```

3. To se the usage of the script, I used the following command (highlighted in yellow).



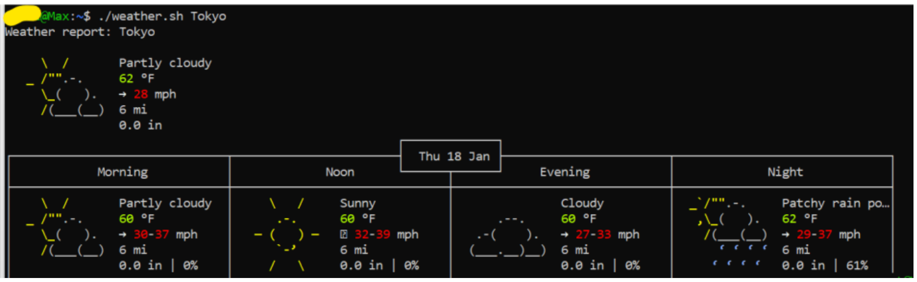
```
[redacted]@Max:~$ ./weather.sh -h
Weather
Description: Provides a 3 day forecast or a simple temperature output on your current location or a specified location.
  With no specified location, Weather will default to your current location.
Usage: weather or weather [flag] or weather [-s] [city / country]
  -s  Simple output
  -h  Show the help
  -v  Get the tool version
Examples:
  weather
  weather Paris
  weather -s London
  weather Italy
```

Note: Using the **-s** argument when using the **./weather.sh** command will output a simpler version of the weather. An example of this is shown below:



```
[redacted]@Max:~$ ./weather.sh -s Tokyo
Tokyo +62°F
```

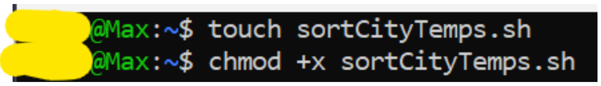
The fancy version without the **-s** argument is shown below:



```
[redacted]@Max:~$ ./weather.sh Tokyo
Weather report: Tokyo

     \  /       Partly cloudy
   _ /"".-.     62 °F
     \_(   ).   → 28 mph
     /(___(__)  6 mi
                0.0 in
                                                      ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤  Thu 18 Jan ├───────────────────────┬──────────────────────────────┐
│            Morning           │             Noon      └──────┬──────┘     Evening           │             Night            │
├──────────────────────────────┼──────────────────────────────┼──────────────────────────────┼──────────────────────────────┤
│    \  /       Partly cloudy  │     \   /     Sunny          │               Cloudy         │ _`/"".-.     Patchy rain po… │
│  _ /"".-.     60 °F          │      .-.      60 °F          │      .--.     60 °F          │  ,\_(   ).   62 °F           │
│    \_(   ).   → 30-37 mph    │   ― (   ) ―   ⍰ 32-39 mph    │   .-(    ).   → 27-33 mph    │   /(___(__)  → 29-37 mph     │
│    /(___(__)  6 mi           │      `-'      6 mi           │  (___.__)__)  6 mi           │     ‘ ‘ ‘ ‘  6 mi            │
│               0.0 in | 0%    │     /   \     0.0 in | 0%    │               0.0 in | 0%    │    ‘ ‘ ‘ ‘   0.0 in | 61%    │
```

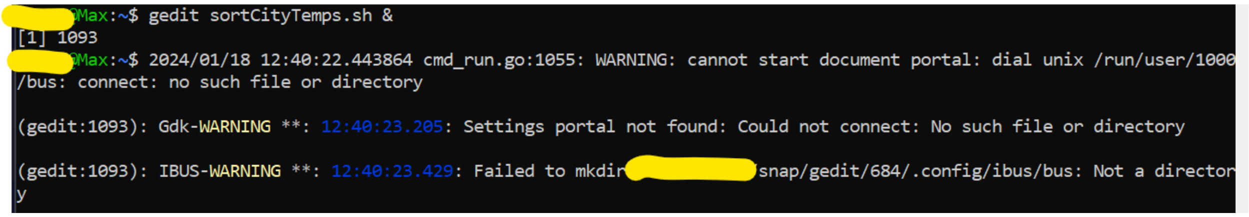
Create Script File

1. I created a shell script file with the **touch** command called **sortCityTemps.sh** and gave it permissions with **chmod +x** command to execute:

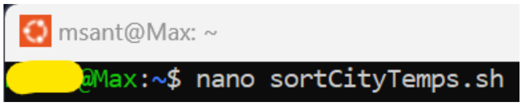


```
[redacted]@Max:~$ touch sortCityTemps.sh
[redacted]@Max:~$ chmod +x sortCityTemps.sh
```

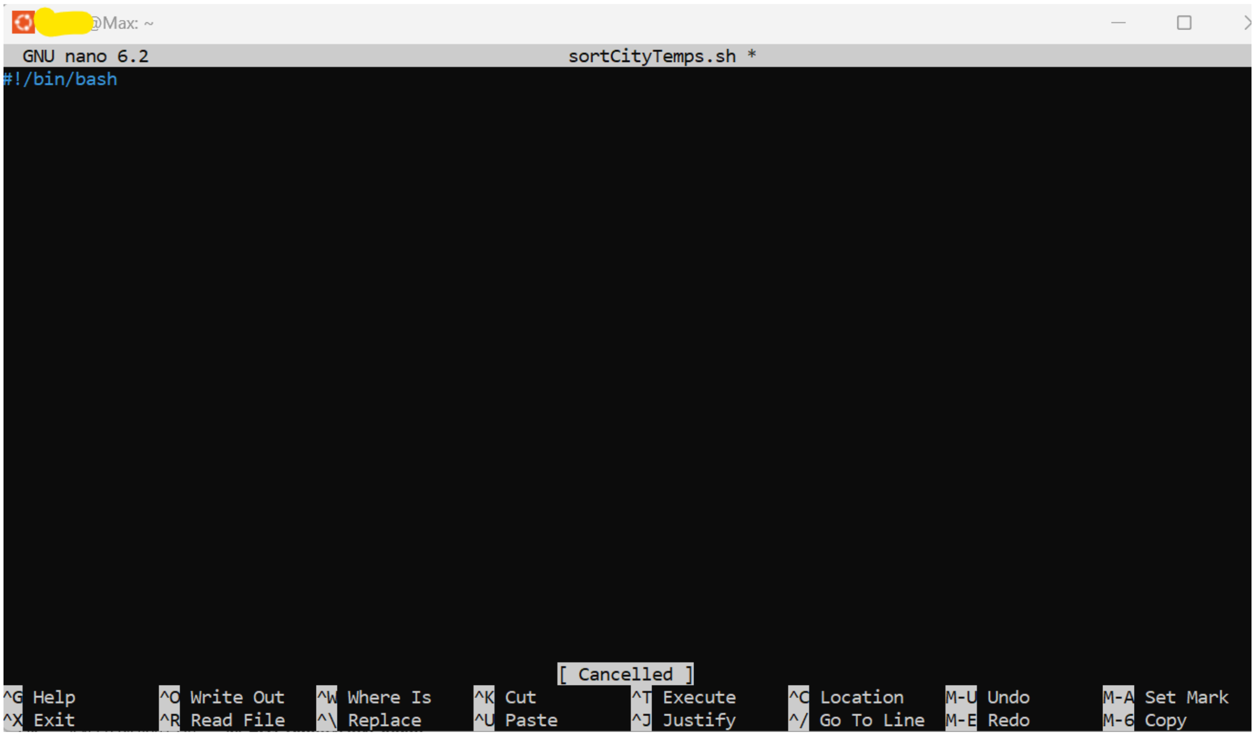
2. Choose a method for working in the file. This file can be worked in either the **Ubuntu Desktop Editor** (which comes pre-installed), a code editor of your choosing, or inside the terminal itself with **nano**. You can open the file in Ubuntu Desktop Editor using the command **gedit sortCityTemps.sh &**. If you decide to use nano, you can save changes to the file using **CTRL/CMD + O** and exit using **CTRL/CMD + X.**
   a. I chose the **nano** code editor with the Ubuntu terminal and added the line **#!/bin/bash** to the top of the script and saved.



```
[redacted]Max:~$ gedit sortCityTemps.sh &
[1] 1093
[redacted]Max:~$ 2024/01/18 12:40:22.443864 cmd_run.go:1055: WARNING: cannot start document portal: dial unix /run/user/1000
/bus: connect: no such file or directory

(gedit:1093): Gdk-WARNING **: 12:40:23.205: Settings portal not found: Could not connect: No such file or directory

(gedit:1093): IBUS-WARNING **: 12:40:23.429: Failed to mkdir[redacted]snap/gedit/684/.config/ibus/bus: Not a director
y
```



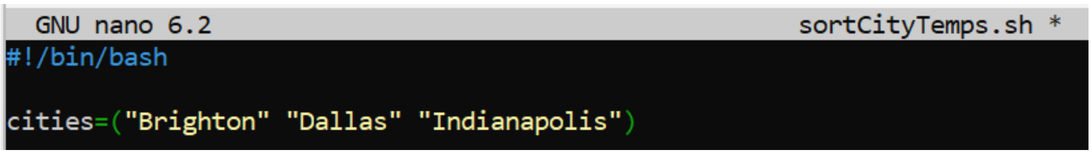
```
[redacted]@Max:~$ nano sortCityTemps.sh
```



```
 GNU nano 6.2                         sortCityTemps.sh *
#!/bin/bash




                                  [ Cancelled ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut       ^T Execute   ^C Location  M-U Undo     M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste     ^J Justify   ^/ Go To Line M-E Redo     M-6 Copy
```

Collect and Sort Temperatures

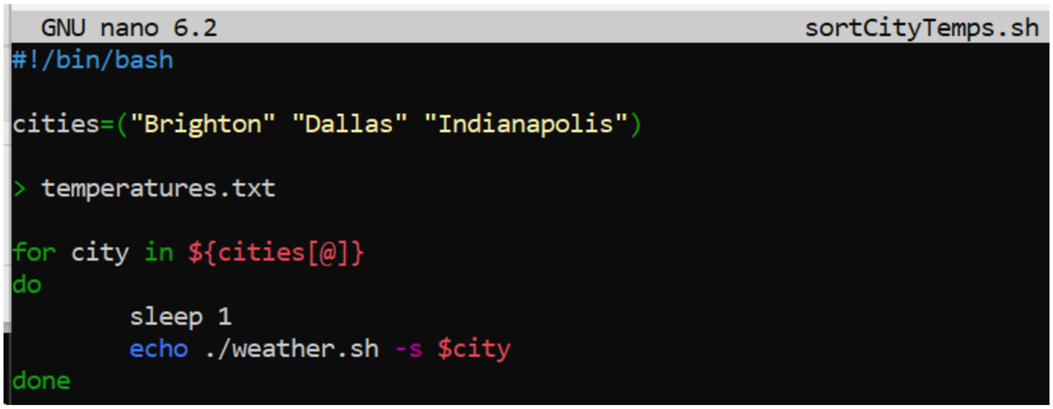
1. I created an array of cities of my own choosing that I will be getting the temperatures from.



```
 GNU nano 6.2                         sortCityTemps.sh *
#!/bin/bash

cities=("Brighton" "Dallas" "Indianapolis")
```

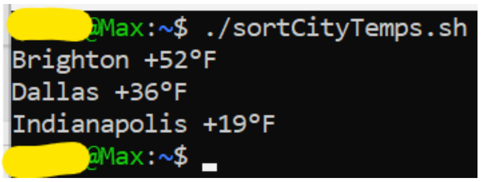
2. In order to save the temperatures, I created an empty file called **temperatures.txt**. The “>” command was used to create an empty file.
3. Next, I set up a loop to iterate through the cities, along with the temperatures, with the **-s** (simple) argument in the array I created. I included a **sleep 1** command within the loop

that pauses the script for 1 second at every iteration to prevent too many requests to the API, if one is being used, in a short amount of time.



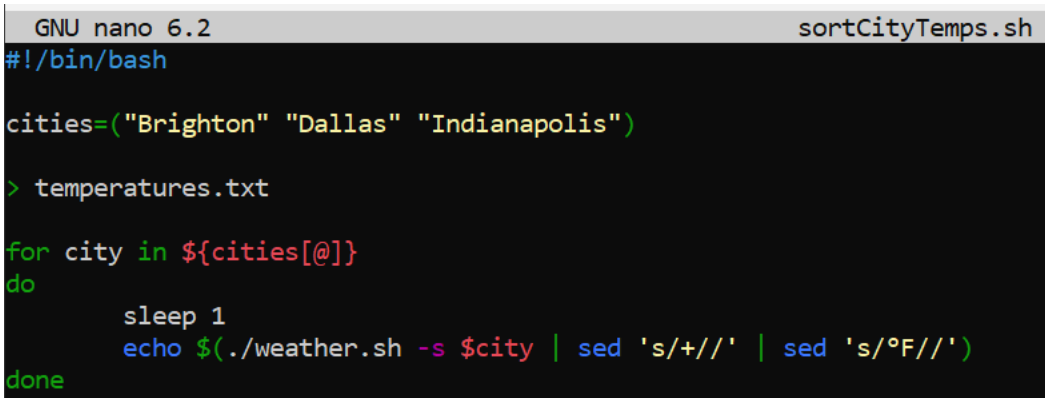
```
  GNU nano 6.2                                          sortCityTemps.sh
#!/bin/bash

cities=("Brighton" "Dallas" "Indianapolis")

> temperatures.txt

for city in ${cities[@]}
do
        sleep 1
        echo ./weather.sh -s $city
done
```

The output of the script in Ubuntu is shown below:



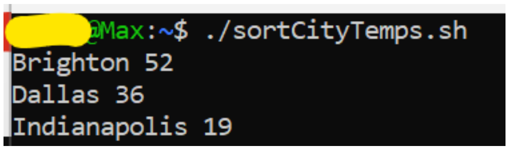
```
[redacted]@Max:~$ ./sortCityTemps.sh
Brighton +52°F
Dallas +36°F
Indianapolis +19°F
[redacted]@Max:~$ _
```

4. I order to avoid having issues with sorting the temperatures, I removed the “+” and “**°F**” symbols. To do this I **piped** the output of the weather script using the **sed** command and replaced each symbol with an empty string. Which is shown below:



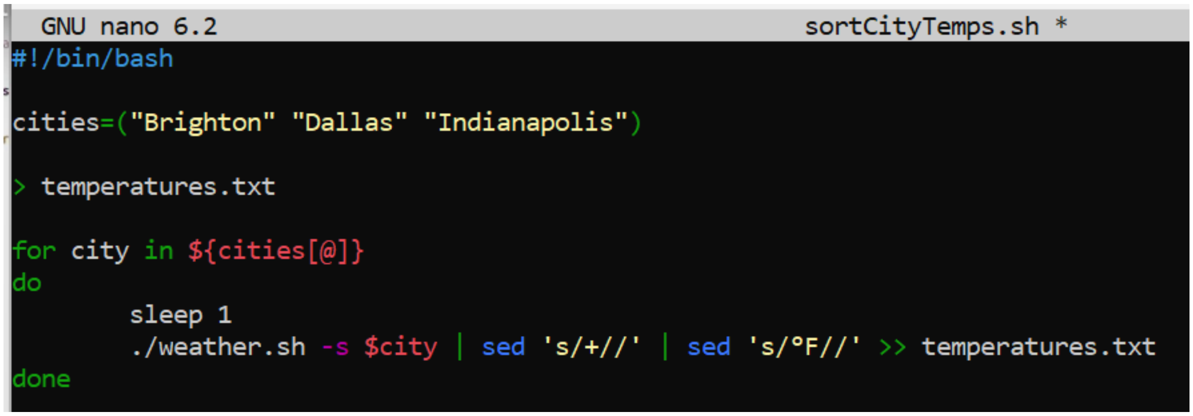
```
  GNU nano 6.2                                          sortCityTemps.sh
#!/bin/bash

cities=("Brighton" "Dallas" "Indianapolis")

> temperatures.txt

for city in ${cities[@]}
do
        sleep 1
        echo $(./weather.sh -s $city | sed 's/+//' | sed 's/°F//')
done
```

Output in Ubuntu without the symbols shown below:



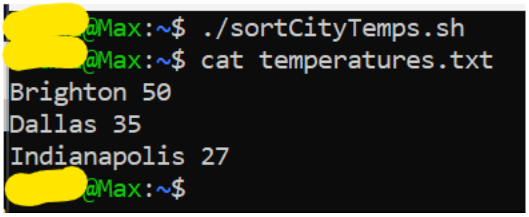
```
[redacted]@Max:~$ ./sortCityTemps.sh
Brighton 52
Dallas 36
Indianapolis 19
```

5. I then redirected the iteration to the **temperatures.txt** file and removed to echo command to avoid overriding on each iteration.



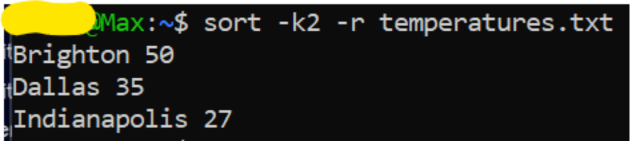
```
  GNU nano 6.2                                          sortCityTemps.sh *
#!/bin/bash

cities=("Brighton" "Dallas" "Indianapolis")

> temperatures.txt

for city in ${cities[@]}
do
        sleep 1
        ./weather.sh -s $city | sed 's/+//' | sed 's/°F//' >> temperatures.txt
done
```

6. I ran the script and checked out the contents of the **temperatures.txt** file with the **cat** command.



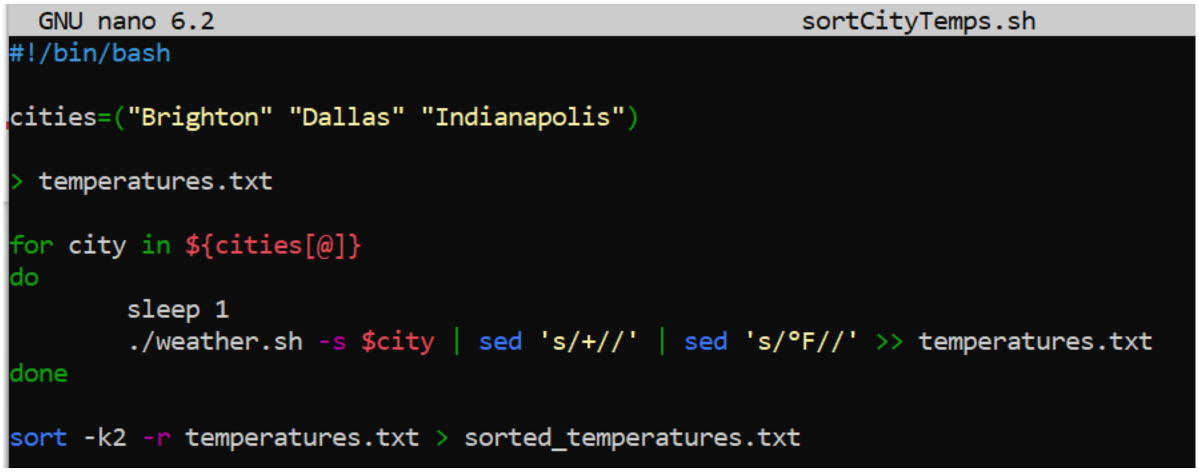
```
[redacted]@Max:~$ ./sortCityTemps.sh
[redacted]@Max:~$ cat temperatures.txt
Brighton 50
Dallas 35
Indianapolis 27
[redacted]@Max:~$
```

7. I sorted the cities based on the temperatures. Using the **sort** command to do this along with the **-k2** argument to sort on the second value in each line (first value is city and second value is the temperature) and the **-r** argument to sort in descending order, this is the output in the Ubuntu terminal:

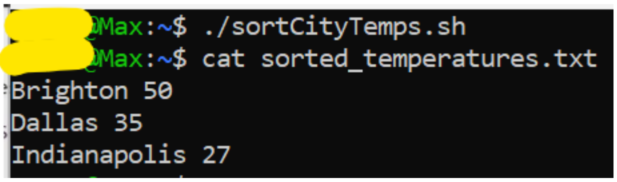


```
[redacted]@Max:~$ sort -k2 -r temperatures.txt
Brighton 50
Dallas 35
Indianapolis 27
```

8. Finally, I added this sorted line to the script after the loop and outputted it to file called **sorted_tepmeratures.txt**, ran the script in Ubuntu, and checked the contents of the new file.

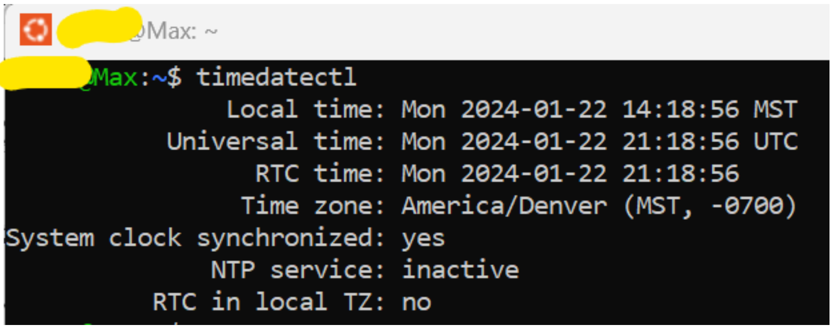


```
  GNU nano 6.2                                         sortCityTemps.sh
#!/bin/bash

cities=("Brighton" "Dallas" "Indianapolis")

> temperatures.txt

for city in ${cities[@]}
do
        sleep 1
        ./weather.sh -s $city | sed 's/+//' | sed 's/°F//' >> temperatures.txt
done

sort -k2 -r temperatures.txt > sorted_temperatures.txt
```



```
[redacted]@Max:~$ ./sortCityTemps.sh
[redacted]@Max:~$ cat sorted_temperatures.txt
Brighton 50
Dallas 35
Indianapolis 27
```

End Project Timestamp



```
[redacted]@Max: ~
[redacted]@Max:~$ timedatectl
               Local time: Mon 2024-01-22 14:18:56 MST
           Universal time: Mon 2024-01-22 21:18:56 UTC
                 RTC time: Mon 2024-01-22 21:18:56
                Time zone: America/Denver (MST, -0700)
System clock synchronized: yes
              NTP service: inactive
          RTC in local TZ: no
```