Week 5

Question 1 : Using command-line arguments involves the sys module. Review the docs for this module and using the information in there write a short program that when run from the command-line reports what operating system platform is being used.

```
import sys

def main():
 print(f"The operating system platform is: {sys.platform}")

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

```
C:\Users\karna>python platform.py
The operating system platform is: win32
```

Question 2: Write a program that, when run from the command line, reports how many arguments were provided. (Remember that the program name itself is not an argument).

```
import sys

def main():
 num_args = len(sys.argv)-1
 print(f"Number of arguments: {num_args}")

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

```
C:\Users\karna>python args_count.py
Number of arguments: 0

C:\Users\karna>python args_count.py arg1 arg2 arg3
Number of arguments: 3
```

Question 3: Write a program that takes a bunch of command-line arguments, and then prints out the shortest. If there is more than one of the shortest length, any will do.

Hint: Don't overthink this. A good way to find the shortest is just to sort them.

```
import sys

def main():
 args = sys.argv[1:]

 if not args:
  print("No arguments were provided.")
  return

 shortest = min(args, key=len)
 print(f"The shortest argument is: {shortest}")

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

```
C:\Users\karna>python short_arg.py computer cool autumn
The shortest argument is: cool
```

Question 4: Write a program that takes a URL as a command-line argument and reports whether or not there is a working website at that address.

Hint: You need to get the HTTP response code.

Another Hint: StackOverflow is your friend.

```
import requests
import sys

def check_website(url):
    try:
        response = requests.get(url)
        if response.status_code == 200:
```

```
            print(f"The website at {url} is working!")
        else:
            print(f"The website at {url} returned status code: {response.status_code}")
    except requests.exceptions.RequestException as e:
        print(f"Failed to reach the website at {url}. Error: {e}")

if __name__ == "__main__":
    if len(sys.argv) != 2:
        print("Usage: python check_website.py <URL>")
        sys.exit(1)

    url = sys.argv[1]
    check_website(url)
```

```
C:\Users\karna>notepad url_check.py

C:\Users\karna>python url_check.py https://www.traveloka.com
The website at https://www.traveloka.com returned status code: 202

C:\Users\karna>python url_check.py http://www.exemp.com
The website at http://www.exemp.com is working!
```

Question 5: Last week you wrote a program that processed a collection of temperature readings entered by the user and displayed the maximum, minimum, and mean. Create a version of that program that takes the values from the command-line instead. Be sure to handle the case where no arguments are provided!

```
import sys

def process_temperatures(temperatures):
    temperatures = [float(temp) for temp in temperatures]

    max_temp = max(temperatures)
    min_temp = min(temperatures)
    mean_temp = sum(temperatures) / len(temperatures)
```

```
    print(f"Maximum Temperature: {max_temp}")
    print(f"Minimum Temperature: {min_temp}")
    print(f"Mean Temperature: {mean_temp:.2f}")

if __name__ == "__main__":

    if len(sys.argv) < 2:
        print("No temperature readings provided. Please enter at least one temperature.")
        sys.exit(1)

    temperature_readings = sys.argv[1:]

    process_temperatures(temperature_readings)
```

```
C:\Users\karna>python temp_args.py 65 70 82 36
Maximum Temperature: 82.0
Minimum Temperature: 36.0
Mean Temperature: 63.25

C:\Users\karna>python temp_args.py
No temperature readings provided. Please enter at least one temperature.
```