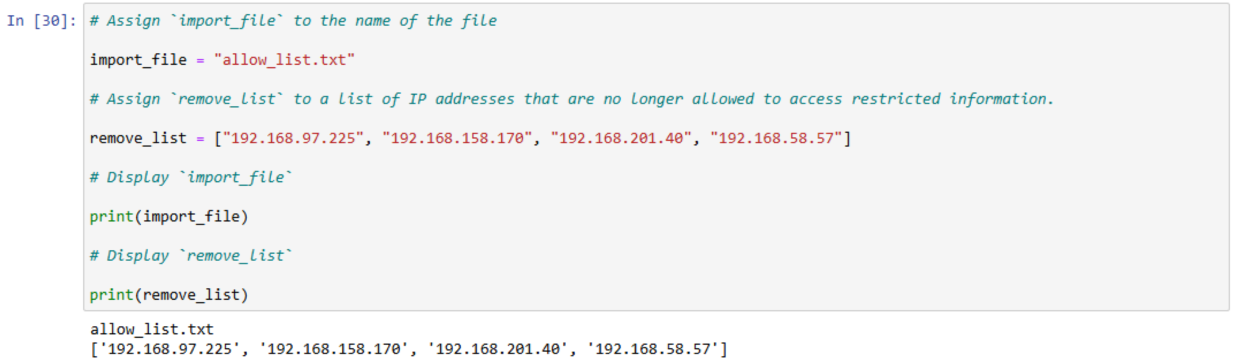
Algorithm for file updates in Python

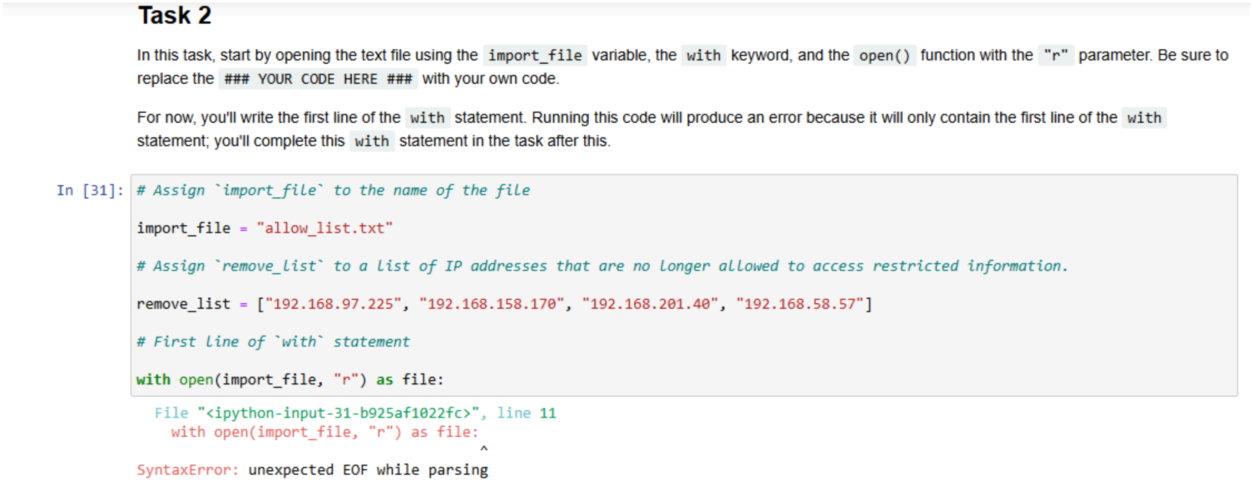
Project description

I played the part of a security analyst at a healthcare company. My task was to update a list that identifies which employees have access. This is based off of the employees IP address. I created an algorithm to check if the user should have access and if not remove their name from the list.

Open the file that contains the allow list



```
In [30]: # Assign `import_file` to the name of the file

         import_file = "allow_list.txt"

         # Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

         remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

         # Display `import_file`

         print(import_file)

         # Display `remove_list`

         print(remove_list)

         allow_list.txt
         ['192.168.97.225', '192.168.158.170', '192.168.201.40', '192.168.58.57']
```



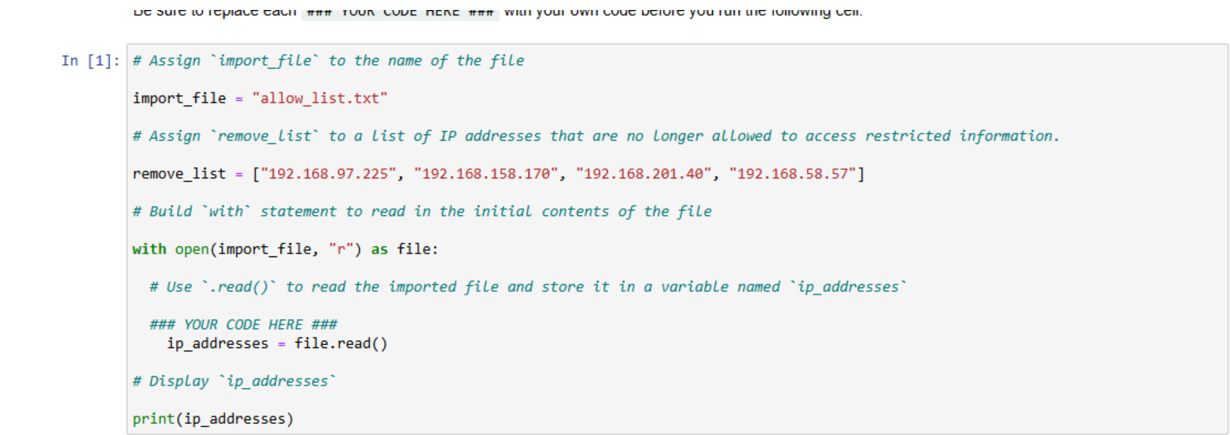
Task 2

In this task, start by opening the text file using the `import_file` variable, the `with` keyword, and the `open()` function with the `"r"` parameter. Be sure to replace the `### YOUR CODE HERE ###` with your own code.

For now, you'll write the first line of the `with` statement. Running this code will produce an error because it will only contain the first line of the `with` statement; you'll complete this `with` statement in the task after this.

```
In [31]: # Assign `import_file` to the name of the file

         import_file = "allow_list.txt"

         # Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

         remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

         # First line of `with` statement

         with open(import_file, "r") as file:

           File "<ipython-input-31-b925af1022fc>", line 11
             with open(import_file, "r") as file:
                                                 ^
         SyntaxError: unexpected EOF while parsing
```

Read the file contents

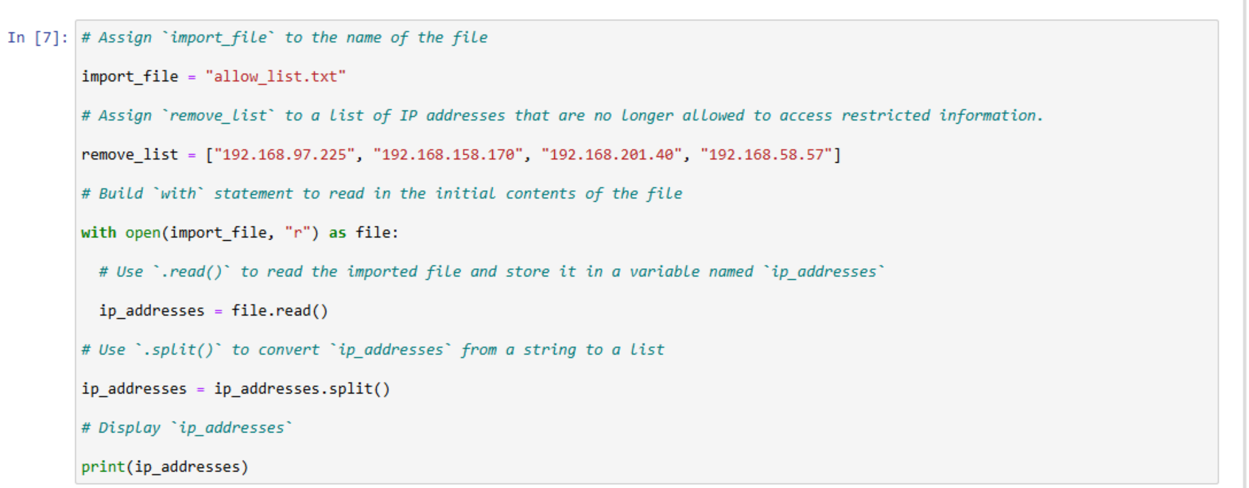


Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [1]:

```
# Assign `import_file` to the name of the file

import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# Build `with` statement to read in the initial contents of the file

with open(import_file, "r") as file:

  # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`

  ### YOUR CODE HERE ###
    ip_addresses = file.read()

# Display `ip_addresses`

print(ip_addresses)
```

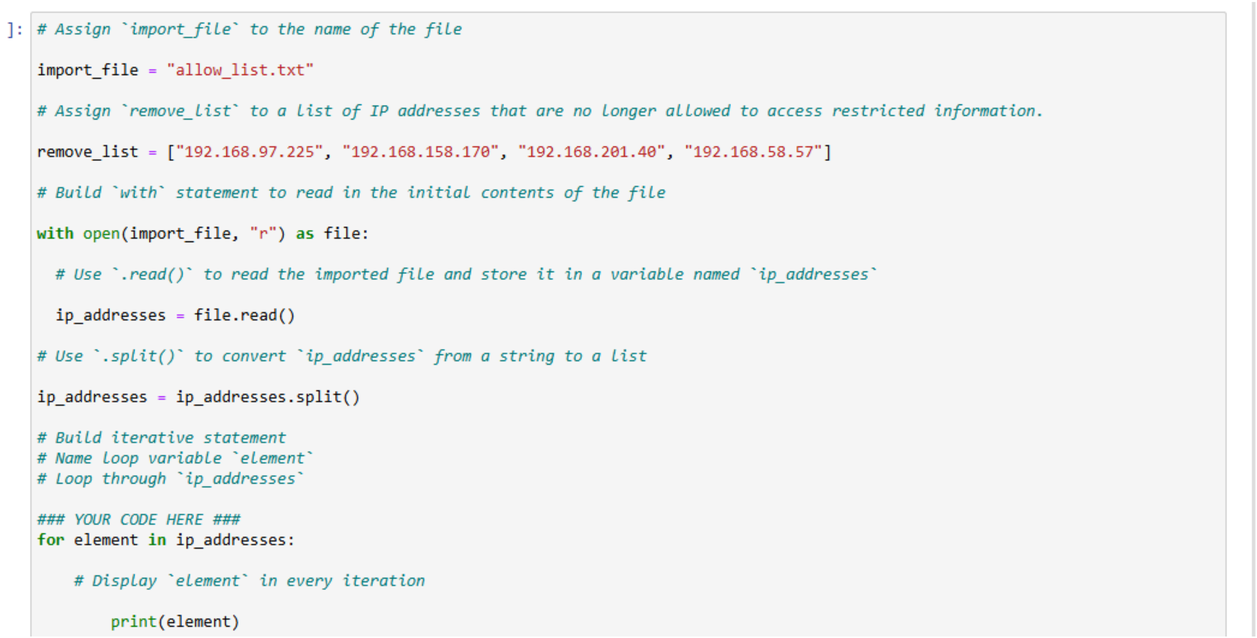
Convert the string into a list



In [7]:

```
# Assign `import_file` to the name of the file

import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# Build `with` statement to read in the initial contents of the file

with open(import_file, "r") as file:

  # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`

  ip_addresses = file.read()

# Use `.split()` to convert `ip_addresses` from a string to a list

ip_addresses = ip_addresses.split()

# Display `ip_addresses`

print(ip_addresses)
```

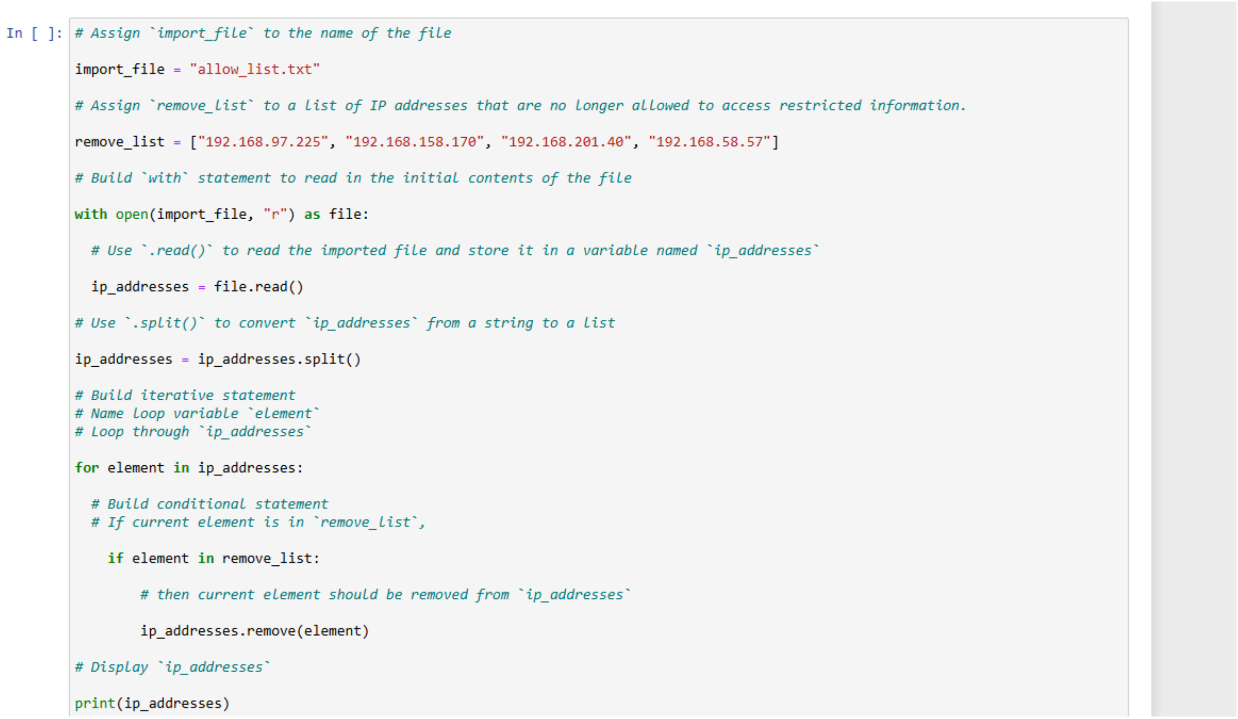
Iterate through the remove list



]:

```
# Assign `import_file` to the name of the file

import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# Build `with` statement to read in the initial contents of the file

with open(import_file, "r") as file:

  # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`

  ip_addresses = file.read()

# Use `.split()` to convert `ip_addresses` from a string to a list

ip_addresses = ip_addresses.split()

# Build iterative statement
# Name loop variable `element`
# Loop through `ip_addresses`

### YOUR CODE HERE ###
for element in ip_addresses:

    # Display `element` in every iteration

        print(element)
```

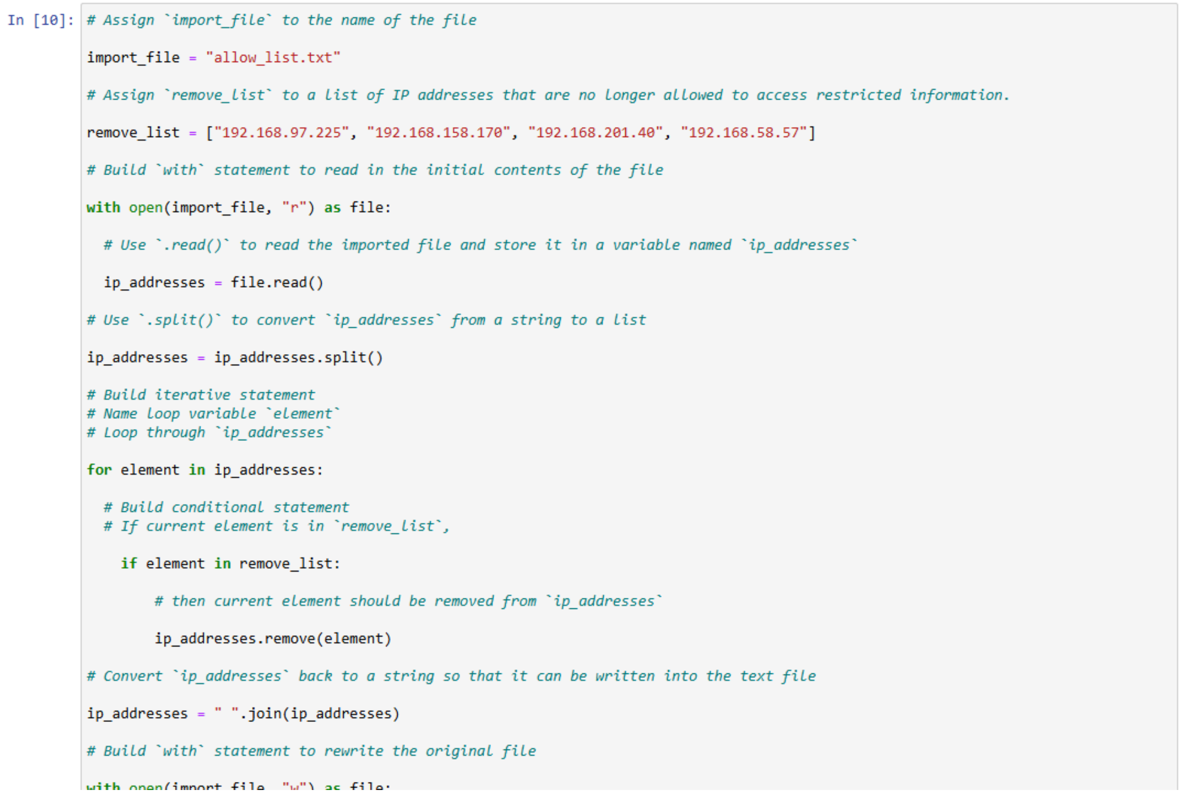
Remove IP addresses that are on the remove list



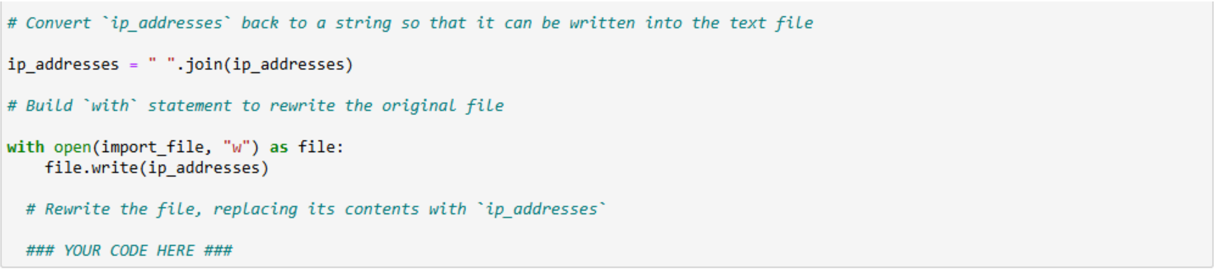
In [ ]:

```
# Assign `import_file` to the name of the file

import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# Build `with` statement to read in the initial contents of the file

with open(import_file, "r") as file:

  # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`

  ip_addresses = file.read()

# Use `.split()` to convert `ip_addresses` from a string to a list

ip_addresses = ip_addresses.split()

# Build iterative statement
# Name loop variable `element`
# Loop through `ip_addresses`

for element in ip_addresses:

  # Build conditional statement
  # If current element is in `remove_list`,

    if element in remove_list:

        # then current element should be removed from `ip_addresses`

        ip_addresses.remove(element)

# Display `ip_addresses`

print(ip_addresses)
```

Update the file with the revised list of IP addresses



```
In [10]: # Assign `import_file` to the name of the file

         import_file = "allow_list.txt"

         # Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

         remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

         # Build `with` statement to read in the initial contents of the file

         with open(import_file, "r") as file:

           # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`

           ip_addresses = file.read()

         # Use `.split()` to convert `ip_addresses` from a string to a list

         ip_addresses = ip_addresses.split()

         # Build iterative statement
         # Name loop variable `element`
         # Loop through `ip_addresses`

         for element in ip_addresses:

           # Build conditional statement
           # If current element is in `remove_list`,

             if element in remove_list:

                 # then current element should be removed from `ip_addresses`

                 ip_addresses.remove(element)

         # Convert `ip_addresses` back to a string so that it can be written into the text file

         ip_addresses = " ".join(ip_addresses)

         # Build `with` statement to rewrite the original file

         with open(import_file, "w") as file:
```



```
# Convert `ip_addresses` back to a string so that it can be written into the text file

ip_addresses = " ".join(ip_addresses)

# Build `with` statement to rewrite the original file

with open(import_file, "w") as file:
    file.write(ip_addresses)

  # Rewrite the file, replacing its contents with `ip_addresses`

  ### YOUR CODE HERE ###
```

Summary

I developed an algorithm to efficiently remove specific IP addresses from a text file containing a list of approved users. The process began by opening the file and reading its contents, which were then split into individual entries for processing. The algorithm iterated through a predefined collection of IP addresses designated for removal, checking each one against the entries from the file. If a match was found, the corresponding IP address was removed from the data. Once all specified IP addresses were processed, the updated data was reformatted and written back to the original file, ensuring the list remained current. This solution highlights my ability to manage data dynamically, maintain file integrity, and apply conditional logic to real-world scenarios.