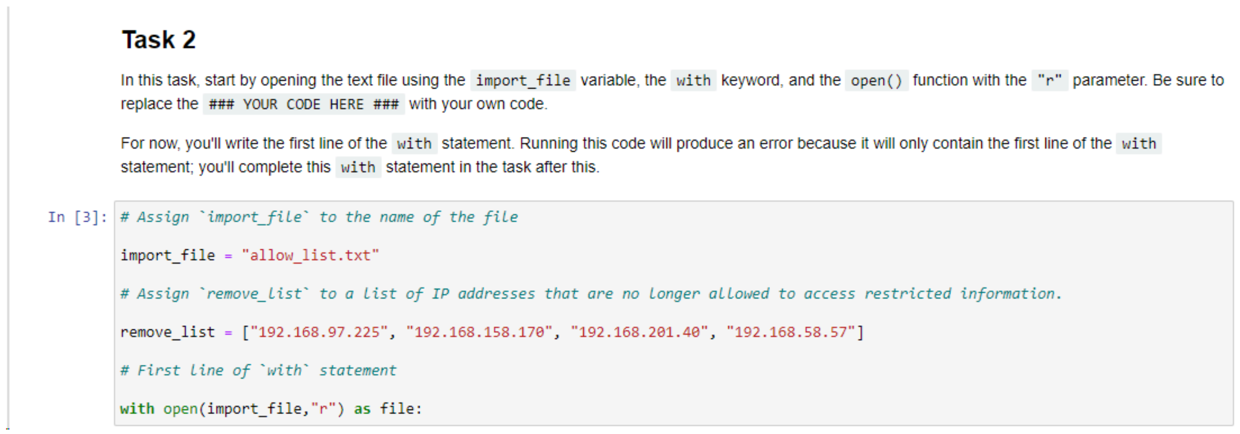
Algorithm for file updates in Python

Project description

To create an algorithm that uses Python code to check whether the allow list contains any IP addresses identified on the remove list. If so, those IP addresses must be removed from the file containing the allow list.

Open the file that contains the allow list



Task 2

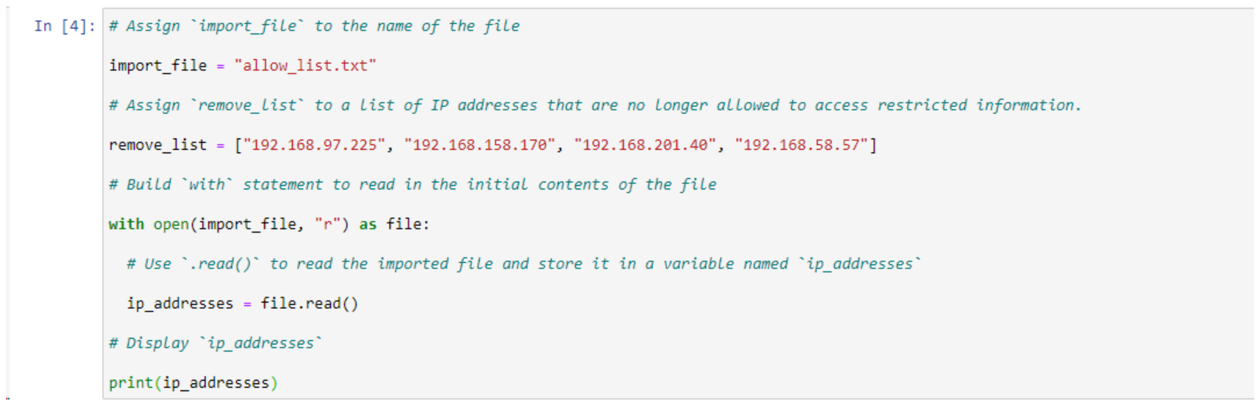
In this task, start by opening the text file using the `import_file` variable, the `with` keyword, and the `open()` function with the `"r"` parameter. Be sure to replace the `### YOUR CODE HERE ###` with your own code.

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 [3]: # 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:
```

To open a file, using the "with" keyword and the 'open()' function. The former ensures that resources are made available and closes the file after the intended operation is done. The later does the opening. The name of the file to be opened is included in the parentheses of the function. The option "r" indicates that we are reading the content of the file. The "as" keyword depicts that we are storing the opened file as "file".

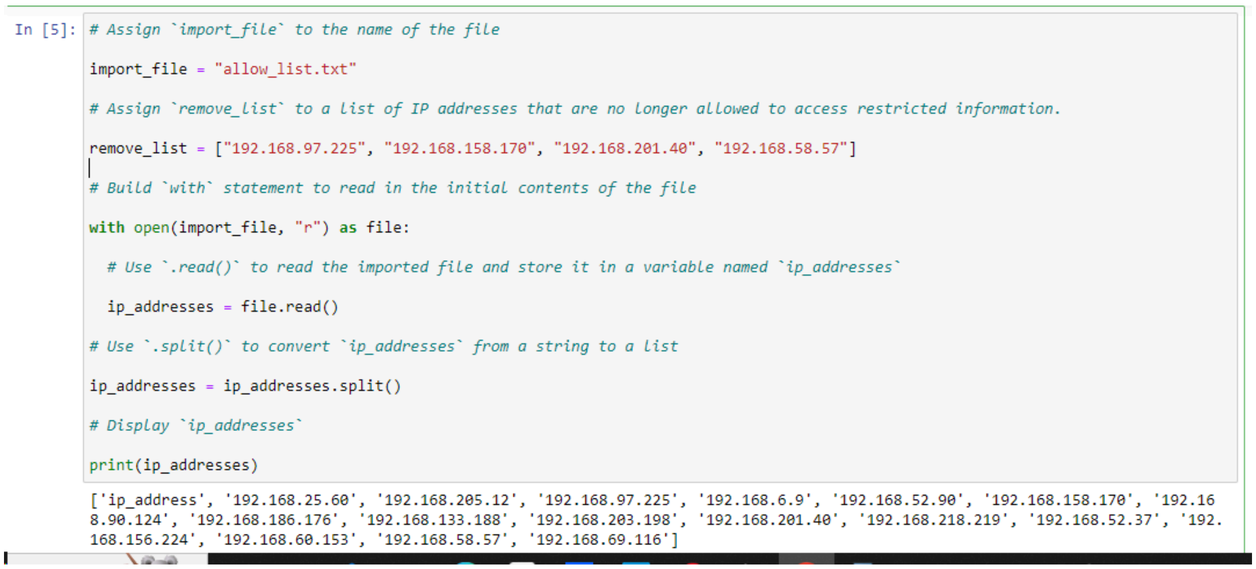
Read the file contents



```
In [4]: # 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()

        # Display `ip_addresses`

        print(ip_addresses)
```

The .read() method is used to read the file and store it in the variable name - ip_addresses, to acceessed later.

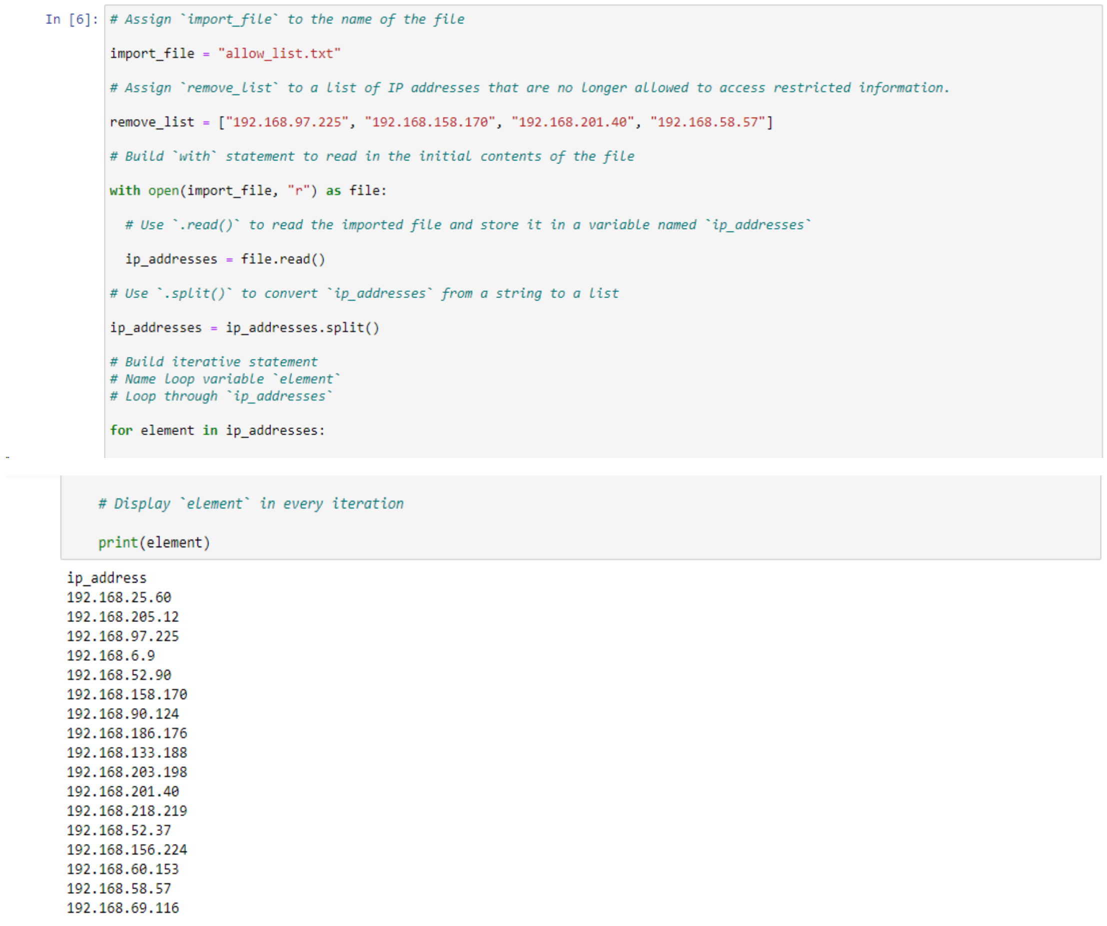
Convert the string into a list



```
In [5]: # 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)

        ['ip_address', '192.168.25.60', '192.168.205.12', '192.168.97.225', '192.168.6.9', '192.168.52.90', '192.168.158.170', '192.16
        8.90.124', '192.168.186.176', '192.168.133.188', '192.168.203.198', '192.168.201.40', '192.168.218.219', '192.168.52.37', '192.
        168.156.224', '192.168.60.153', '192.168.58.57', '192.168.69.116']
```

To convert the string in the imported file to a list we use the .split() method. This involves reassigning the ip_addresses variable. This converts the string to list as shown in the result above.

Iterate through the remove list

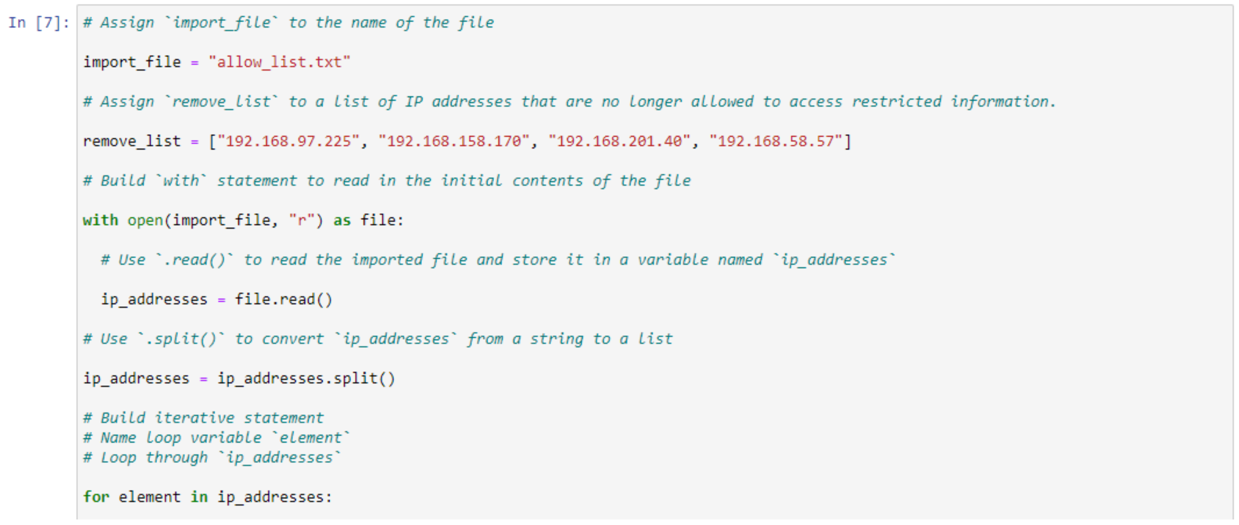


```
In [6]: # Assign `import_file` to the name of the file

        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:

            # Display `element` in every iteration

            print(element)
```

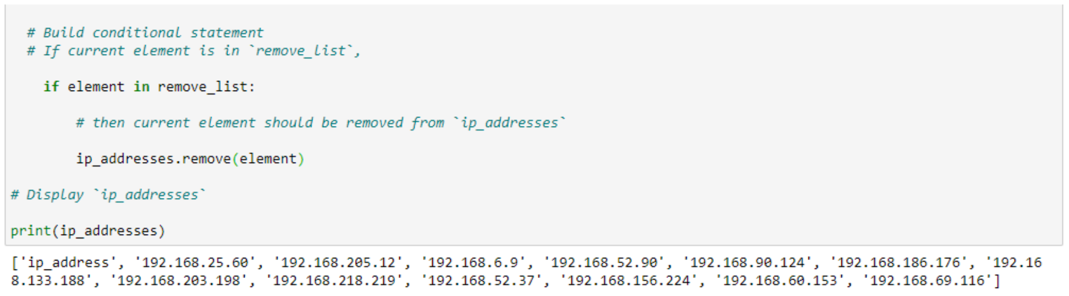
```
ip_address
192.168.25.60
192.168.205.12
192.168.97.225
192.168.6.9
192.168.52.90
192.168.158.170
192.168.90.124
192.168.186.176
192.168.133.188
192.168.203.198
192.168.201.40
192.168.218.219
192.168.52.37
192.168.156.224
192.168.60.153
192.168.58.57
192.168.69.116
```

To iterate through a list, we use an iterative statement with “for” keyword and “element” as the loop variable to look into the iterable list called ip_addresses in this instance. The result is shown above.

Remove IP addresses that are on the remove 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()

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

        for element in ip_addresses:
```

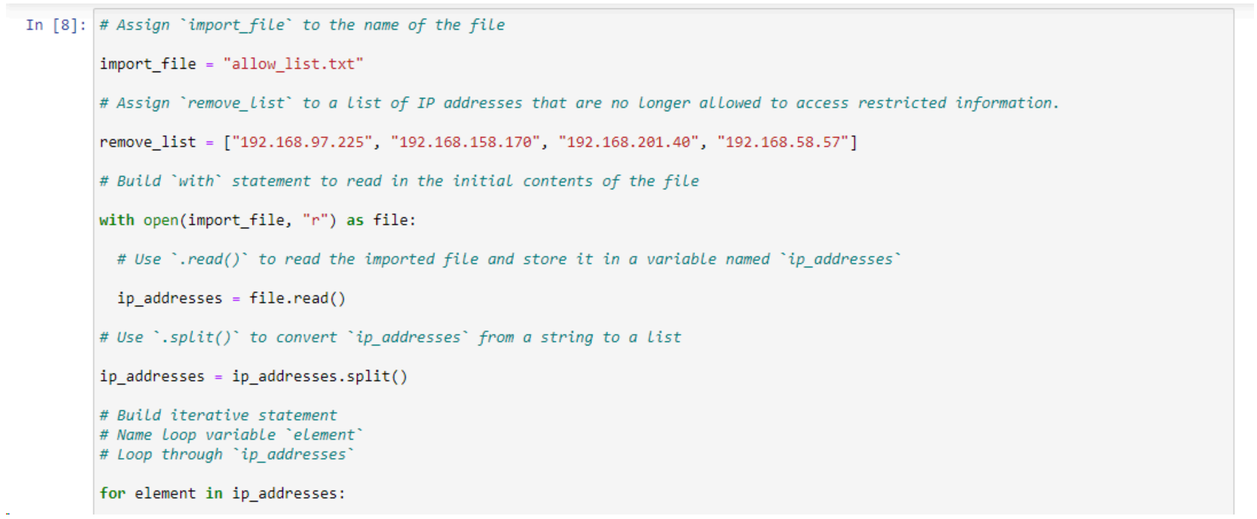


```
  # Build conditional statement
  # If current element is in `remove_list`,

    if element in remove_list:

        # then current element should be removed from `ip_addresses`

        ip_addresses.remove(element)

# Display `ip_addresses`

print(ip_addresses)
```

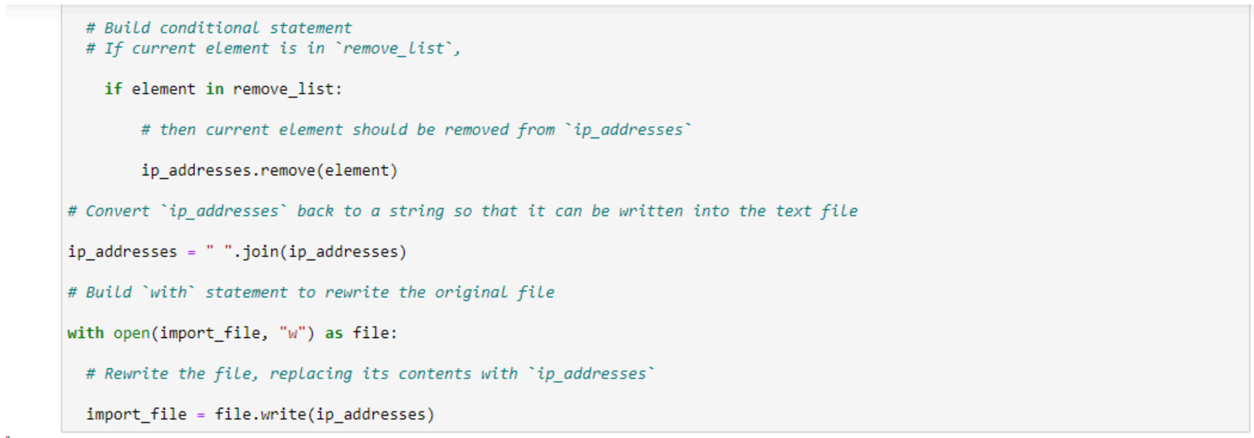
```
['ip_address', '192.168.25.60', '192.168.205.12', '192.168.6.9', '192.168.52.90', '192.168.90.124', '192.168.186.176', '192.16
8.133.188', '192.168.203.198', '192.168.218.219', '192.168.52.37', '192.168.156.224', '192.168.60.153', '192.168.69.116']
```

To remove the forbidden ip_addresses from the allow_list, we use the conditional statement '*if*' to check the *remove_list* and compare with the *allow_list*, if the element is present, the *.remove* method is used and pass the loop variable *element into it.*

Update the file with the revised list of IP addresses



```
In [8]: # 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:

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

  import_file = file.write(ip_addresses)
```

To update the file we need to convert the ip_addresses list back to string format. To accomplish this, we use the .join() method with the " " to signify a space and then reassign the ip_addresses variable and pass it as well as the parameter in the .join() method.
The last is to overwrite the original allow_list with the content of the ip_addresses file. To do this we open the imported_file with "w" option to allow for writing. Then we write the content of the ip_addresses file into the imported _file by using the .write() method with the *file* content of the open function.

Summary

Accessing file in python helps analysts to be able to work strings and convert them to lists in order to iterate through them and do necessary security checks.
In this project, a lot of functions were used such as open(). Also some methods were also employed such as .remove(), .split(), .read(), .write() and so on.
Some keywords in python were also incorporated in the project such as *in, as* and *with.*