Algorithm for file updates in Python

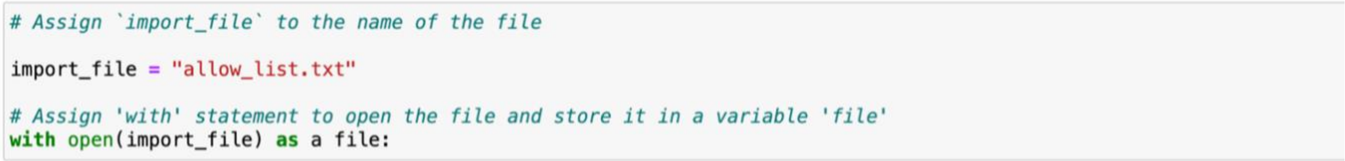
Project description

This task requires me to regularly update a file that identifies the employees who can access restricted content for a health care company. The contents of the file are about personal patient records and there are IP addresses that should have no access to this information but still have the access to sign into the restricted subnetwork.

I was **provided with a remove list which specifies which employees must remove from this allow list** and my task is **to create an algorithm that uses Python code to check whether the allow list contains any IP addresses identified on the remove list**

1) Open the file that contains the allow list

First, I assign the string “allow_list.txt” to the variable import_file
Then I use ‘with’ statement to open the file and assign the variable file to store the file

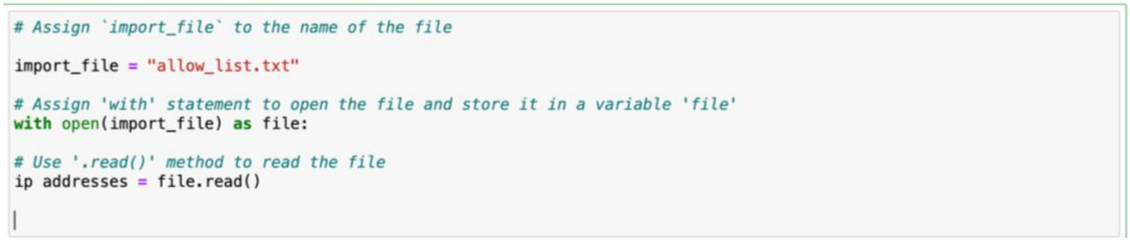


```
# Assign `import_file` to the name of the file

import_file = "allow_list.txt"

# Assign 'with' statement to open the file and store it in a variable 'file'
with open(import_file) as a file:
```

Note(s): As a beginner in Python, it might be confusing to see the ‘with open (import_file) as file:’ and ‘with open(import_file, ‘r’) as file:’. But they are producing the same output, the difference between the two of them is only the ‘r’, it indicates read mode.

2) Read the file contents

After converting the file into a string, use ‘.read()’ method to read the file and then store the file into a variable called ‘ip_addresses’



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

import_file = "allow_list.txt"

# Assign 'with' statement to open the file and store it in a variable 'file'
with open(import_file) as file:

# Use '.read()' method to read the file
ip addresses = file.read()

|
```

Note(s): The general syntax for Python is variable = expression.

Variable: The variable on the left side is where the result of the expression will be stored
Expression: The expression on the right side is evaluated, and its result is assigned to the variable

In this case, applying ‘file.read()’

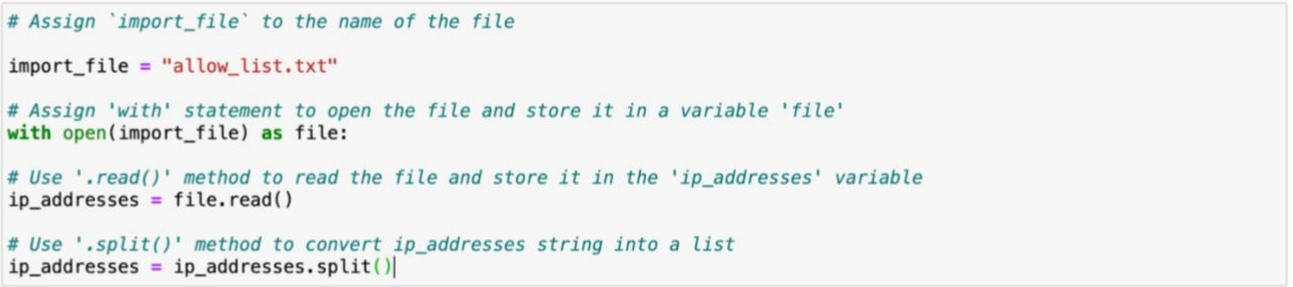
- Ip_addresses: The variable where you want to store the contents of the file
- file.read(): The expression that reads the entire content of the file and returns it as a string

In addition, ‘,read()’ method and the ‘r’ mode serve different purposes in file handling in Python.

‘r’ mode: specifies how the file should be opened
‘.read()’ method: is used to actually read the contents of the file once it is opened

3) Convert the string into a list

Use the ‘.split()’ method to convert the ip_adressess string into a list to remove individual IP addresses from the allow list.

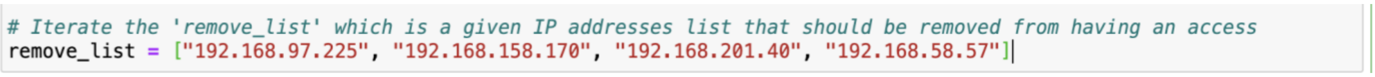


```
# Assign `import_file` to the name of the file

import_file = "allow_list.txt"

# Assign 'with' statement to open the file and store it in a variable 'file'
with open(import_file) as file:

# Use '.read()' method to read the file and store it in the 'ip_addresses' variable
ip_addresses = file.read()

# Use '.split()' method to convert ip_addresses string into a list
ip_addresses = ip_addresses.split()
```

Note(s): .split method is helpful to break a string into a list

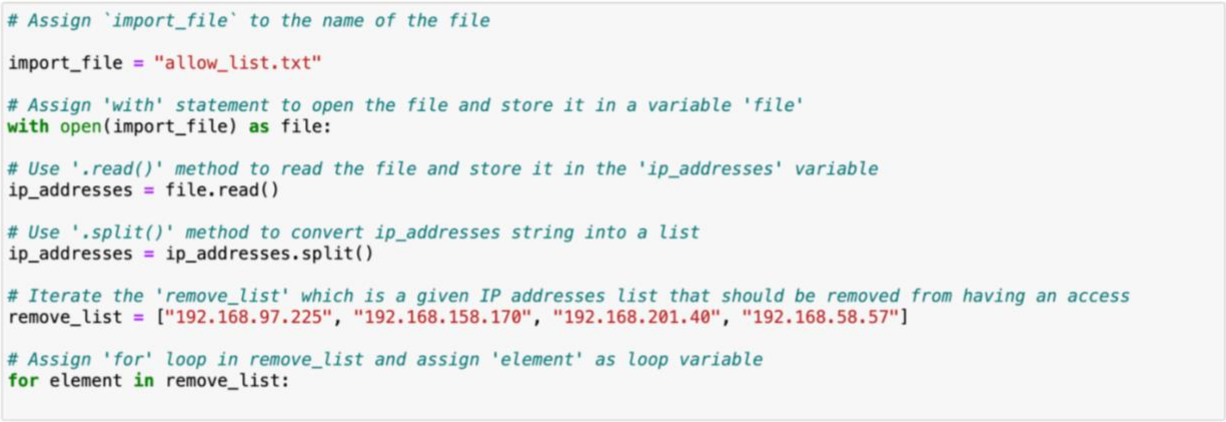
4) Iterate through the remove list

Given the ‘remove_list’ is the list of IP addresses that should be removed



```
# Iterate the 'remove_list' which is a given IP addresses list that should be removed from having an access
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]
```

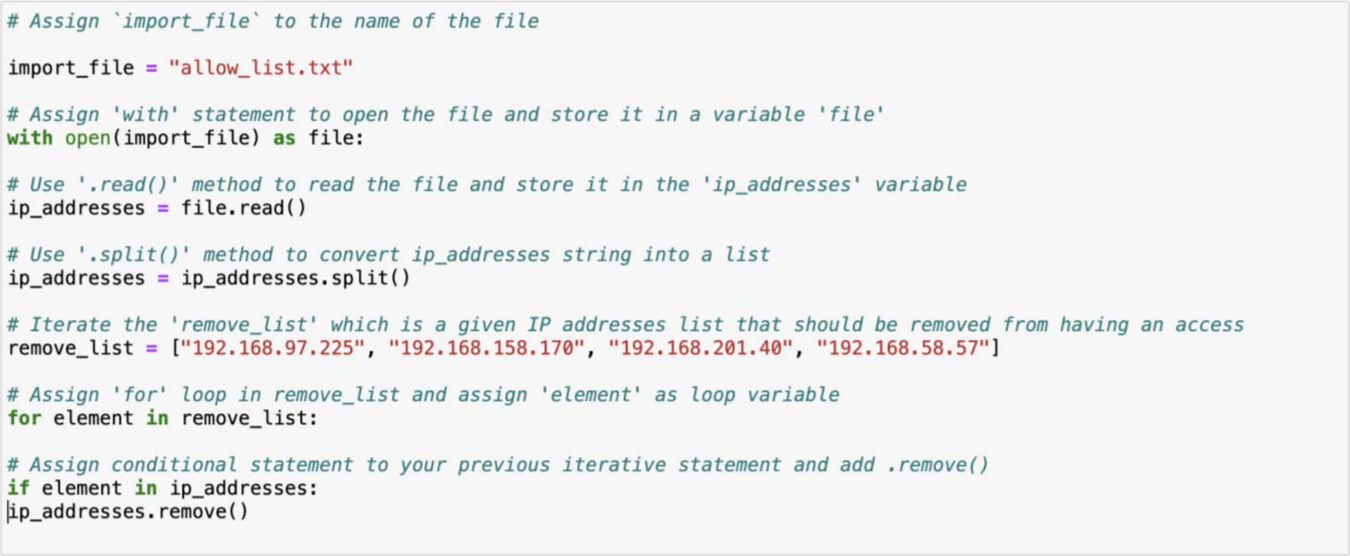
So, I was tasked to set up the header of a ‘for’ loop that will iterate through the ‘remove_list’ and use ‘element’ as the loop variable



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

import_file = "allow_list.txt"

# Assign 'with' statement to open the file and store it in a variable 'file'
with open(import_file) as file:

# Use '.read()' method to read the file and store it in the 'ip_addresses' variable
ip_addresses = file.read()

# Use '.split()' method to convert ip_addresses string into a list
ip_addresses = ip_addresses.split()

# Iterate the 'remove_list' which is a given IP addresses list that should be removed from having an access
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# Assign 'for' loop in remove_list and assign 'element' as loop variable
for element in remove_list:
```

5) Remove IP addresses that are on the remove list

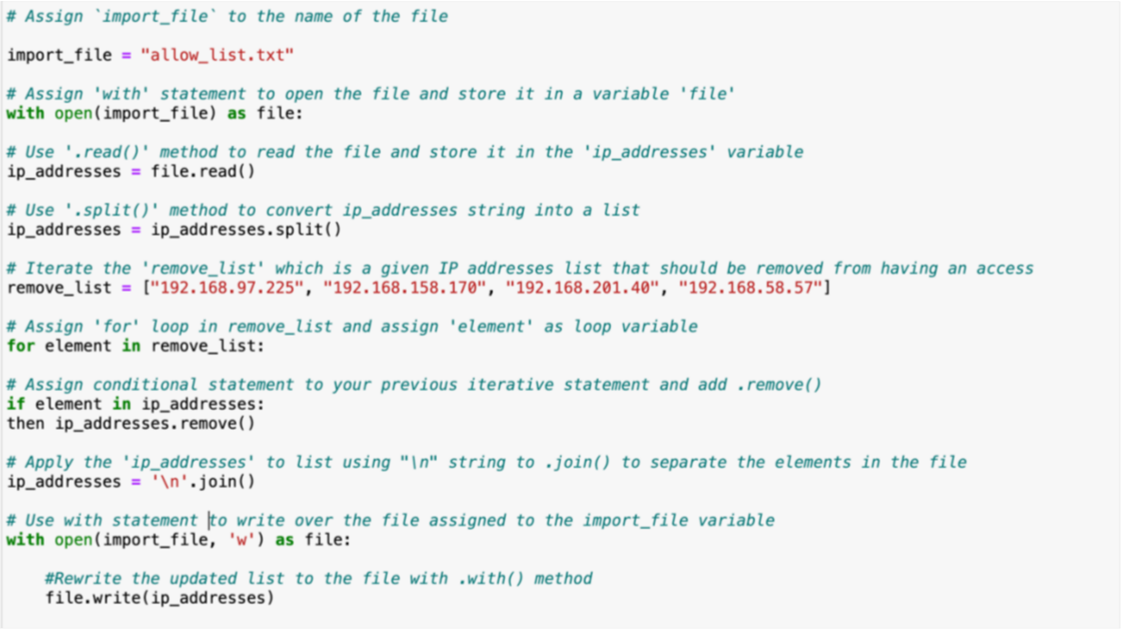
Then I am adding a conditional statement ‘if’ to my previous iterative statement and then apply .remove() method to the list to remove the IP addresses identified in the ‘element’ loop variable.



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

import_file = "allow_list.txt"

# Assign 'with' statement to open the file and store it in a variable 'file'
with open(import_file) as file:

# Use '.read()' method to read the file and store it in the 'ip_addresses' variable
ip_addresses = file.read()

# Use '.split()' method to convert ip_addresses string into a list
ip_addresses = ip_addresses.split()

# Iterate the 'remove_list' which is a given IP addresses list that should be removed from having an access
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# Assign 'for' loop in remove_list and assign 'element' as loop variable
for element in remove_list:

# Assign conditional statement to your previous iterative statement and add .remove()
if element in ip_addresses:
ip_addresses.remove()
```

6) Update the file with the revised list of IP addresses

Finally, the list is updated. For the last step, we need to update the allow_list.txt file with the latest list. So, convert the list ‘ip_addresses’ list back to string using the .join () method and apply ‘.join()’ to the string “\n”. After that, use with statement and .write() method to write over the file with the latest list.



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

import_file = "allow_list.txt"

# Assign 'with' statement to open the file and store it in a variable 'file'
with open(import_file) as file:

# Use '.read()' method to read the file and store it in the 'ip_addresses' variable
ip_addresses = file.read()

# Use '.split()' method to convert ip_addresses string into a list
ip_addresses = ip_addresses.split()

# Iterate the 'remove_list' which is a given IP addresses list that should be removed from having an access
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# Assign 'for' loop in remove_list and assign 'element' as loop variable
for element in remove_list:

# Assign conditional statement to your previous iterative statement and add .remove()
if element in ip_addresses:
then ip_addresses.remove()

# Apply the 'ip_addresses' to list using "\n" string to .join() to separate the elements in the file
ip_addresses = '\n'.join()

# Use with statement to write over the file assigned to the import_file variable
with open(import_file, 'w') as file:

    #Rewrite the updated list to the file with .with() method
    file.write(ip_addresses)
```

Note(s): Using ‘w’ can help to avoid duplicated information. ‘w’ is essential when you want to open a file for writing.

Summary

This task requires me to create a Python algorithm to remove IP addresses from the file and then update the file with the latest list.