# Algorithm for file updates in Python

## Project description

I am a security professional working for a healthcare organization and I need to update a file regularly. This file lists employees who are allowed to access restricted information. These employees are restricted from logging in based on their IP which is in an allow list. There is also a remove list which lists IP’s to be removed from the allow list.

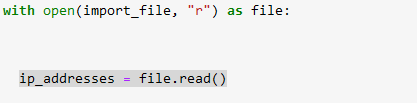
## Open the file that contains the allow list

To access the allow list, the file must first be opened by assigning it to a variable named 'import\_file' for ease of use. The 'with open(import\_file, 'r'):' statement is employed, utilizing the 'with' context manager to handle resources efficiently, ensuring the file is properly opened and automatically closed afterward. The 'open()' method takes two parameters: the first specifies the file to be opened ('import\_file'), and the second ('r') indicates the intention to read the file.

The 'as' statement is then used to assign the file's contents to a variable named 'file.'  
  


## Read the file contents

Next, to read the file contents, the variable used for opening the file is reassigned to a new variable named 'ip\_addresses.' This step ensures the data can be utilized outside the 'with' statement. The assignment employs the '=' operator, and the 'ip\_addresses' variable is populated with the file's contents using the '.read()' method, transforming it into usable data.



## Convert the string into a list

To enhance the readability of the allow\_list variable, we convert its string data into a list using the '.split()' method. This method divides the information into segments, and by default, it uses whitespaces and other characters as delimiters.



## Iterate through the remove list

To automate the removal of IPs from the allow list, a 'for' loop is employed to iterate through the 'ip\_addresses' list. Using the 'for' keyword initiates the loop, where a new variable, 'element,' is assigned to each item in the 'ip\_addresses' list.

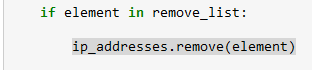


## Remove IP addresses that are on the remove list

With the variable and 'for' loop established, we proceed to utilize the loop for removing IP addresses from the list stored in the 'ip\_addresses' variable. The '.remove()' method, suitable for a list without duplicates, is employed on the allow list within the loop to eliminate any matching IP addresses.



We instruct the loop to remove an element from the allow list if it is present in the remove list, utilizing the '.remove()' method for this purpose.

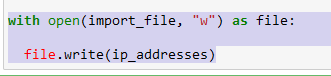


## Update the file with the revised list of IP addresses

With the removal of IPs from the 'remove\_list' in the 'for' loop, the next step is to update the allow list with the modified data. The '.join()' method is employed for this task.



In this example, we employed the '.join()' method with a space as the separator to transform the modified 'ip\_addresses' list into a string. The result is then assigned to the variable labeled 'ip\_addresses.'



Utilizing the 'with' and 'open' statements, we reopen the file in write mode ('w') to update its contents. The '.write()' method is then employed, with 'ip\_addresses' as the variable specifying the data to be written. It's crucial to direct the write operation to the 'file' variable, which represents the opened file.

## Summary

In this algorithm designed for managing a file of allowed IP addresses, the process begins with opening the file using the 'with' and 'open' statements, assigning the file to the variable 'import\_file.' The file's contents are read and stored in the 'ip\_addresses' variable after converting them into a list for easier manipulation. A 'for' loop is then employed to iteratively remove IP addresses listed in the 'remove\_list' from the 'ip\_addresses' list. The modified list is subsequently joined back into a string using the '.join()' method with a space as the separator, and the updated data is written back to the file in write mode ('w') using the '.write()' method within the same 'with' context manager.