# Algorithm for file updates in Python

## Project description

Access to restricted content is controlled by using an allow list of IP addresses. The “allow\_list.txt” files identifies these IP addresses. A separate remove list identifies IP addresses that should no longer have access to this content. I created an algorithm that should automate updating the “allow\_list.txt” file and remove the IP addresses that shouldn’t have access anymore.

## Open the file that contains the allow list

First part of the algorithm I opened the “allow\_list.txt” file. I assigned a file name as a string to the import\_file variable.

A close up of a name

Description automatically generated

I then used a with statement to open the file.

A white background with text

Description automatically generated

In my algorithm the with statement is used with the open() function in read mode to open the allow file list and read it. The purpose of opening the file is to access the IP addresses stored in the allow list file. Using the with keyword will close the file after exiting the with statement. The code also uses the as keyword to assign a variable name of file. file stores the output of the open() function while I work in the with statement.

## Read the file contents

I use the .read() method to covert the file contents into a string.

A close-up of a white background

Description automatically generated

When using the open() function using the argument of “r” for read, I can call the read() function in the with statement. The read() function allows me to convert the file into a string and read it. I applied the read() method to the file variable and then assigned the string output to the variable ip\_addresses.

## Convert the string into a list

In order to remove IP addresses from the allow list, it needed to be in list format. I used the split() function to convert the ip\_addresses string into a list:

A close-up of a computer code

Description automatically generated

The split() function converts the contents of a string to a list. The purpose of splitting ip\_addresses into a list is to make it easier to remove IP addresses from the allow list. The split() function splits the text by whitespace into a list. In this algorithm, the split() function takes the data stored in the variable of ip\_addresses, and converts this string into a list of IP addresses. To store the list I then reassigned it back to the variable ip\_addresses.

## Iterate through the remove list

A key part of the algorithm is to iterate through the IP addresses that are elements in the remove\_list. In order to accomplish this, I used a for loop:

A screen shot of a computer

Description automatically generated

The for loop repeats code for a specified sequence. The purpose of the for loop in this algorithm is to apply specific code statements to all the elements in the sequence. The for keyword starts the loop and is followed by the loop variable element and keyword in. The in keyword iterates through the sequence ip\_addresses and assigns each value to the loop variable element.

## Remove IP addresses that are on the remove list

My algorithm requires to remove any IP address from the allow list, ip\_addresses, that is also in the remove\_list. Since there were no duplicates in ip\_addresses, I was able to use the code to this:

A computer code with black text

Description automatically generated with medium confidence

Within the for loop, I created a conditional statement that evaluated whether the variable element was found in the ip\_addresses list. Then I applied the remove() function to ip\_addresses. I passed in the loop variable element as an argument so that each IP address in remove\_list would be removed from ip\_addresses.

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

For the final step in the algorithm, I needed to update the allow list file with a revised list of IP addresses. In order to do this I converted the list back into a string. I use the join() method for this:

A white background with black text

Description automatically generated

The join() method combines all items in an iterable into a string. I used the join() method to create a string from the list of ip\_addresses so that I could pass it as an argument to the write() method when writing to the file “allow\_list.txt”.I used the string (“\n”) as a separator to place each element on a new line to make it more legible.

I then used another with statement and the write() method to update the file:

A screenshot of a computer program

Description automatically generated

I used a second argument of “w” this time with the open() function in my with statement. This “w” argument means I want to open a file and write over its contents. The write() function writes a string data to a specified file and overwrites any existing file content.

In this case I wanted to write the updated allow list to the file “allow\_list.txt”. This way the content is no longer accessible to any IP addresses that were removed from the allow list. To rewrite the file I used the write() function to the file object file that is identified in the with statement. I passed the variable ip\_addresses as an argument to state that the contents of the file in the with statement should be replaced with the data in this variable.

## Summary

I created an algorithm that removes IP addresses from the remove\_list variable from the “allow\_list.txt” file of approved IP addresses. I then used a for loop to iterate through the IP addresses in remove\_list, which evaluated if the element was part of the ip\_addresses list. If it was then the remove() method was used to remove that element from ip\_addresses. Then I used the join() method to convert the ip\_addresses back into a string so I could write the contents of the “allow\_list.txt” file with revised list of IP addresses.