# Algorithm for file updates in Python

## Project description

My organization uses an allowed list of IP addresses to access restricted content. The file is called “allow\_list.txt” and helps identify approved IP addresses. A secondary list is used to identify ip addresses that need to be removed and can no longer gain access to restricted content. I have crafted an algorithm to automate the process of updating “allow\_list.txt” and remove IP addresses that should no longer have access.

## Open the file that contains the allow list

To start the algorithm, I opened the “allow\_list.txt” file. I then assigned this file name as a string to the import\_file variable:



Second, I used a with statement to open the file:



The with statement is used with the open.() function in read mode to open the allow list file and read it. Doing this allows me to access the list of IP addresses contained in the file. The with keyword helps manage the contents by closing the file once the with statement has been exited in the code. In the code with open(“import\_file”, “r”) as file: the open() function contains two parameters. The first parameter is “import\_file” which indicates the file to import, the second is “r” which indicates that the file is to be read. The code also has the as keyword to assign a variable named file: this variable stores the output of the .open() function.

## Read the file contents

To be able to read the file contents, I used the .read() method to convert it into the string.



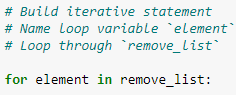
The .read() method converts the file into a string and allows me to read it. I used the .read() method on the file variable identified in the with statement. Lastly, I assigned the string output to the ip\_addresses variable.

## Convert the string into a list

In order to remove single IP addresses from the allow list, I needed to have them in list format. The .split() method is used to convert ip\_addresses into a list:  


The .split() function is called by appeneding it to a string variable. The purpose of doing this is to make it easier to remove IP addresses from the allow list. The .split() function splits the text into list elements. In this algorithm, the .split() uses the data stored in the ip\_addresses variable and converts the string into a list of IP addresses.

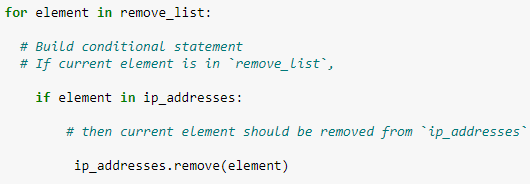
## Iterate through the remove list

A large part of the algorithm involves iterating through the IP addresses that are in remove\_list.  


A for loop in Python repeats code for a specified sequence. The for keyword begins the for loop and it is followed by the loop variable. In this case the loop variable is element. Followed by the keyword in, this indicates to iterate through the sequence ip\_addresses and assign each value to the loop variable element.

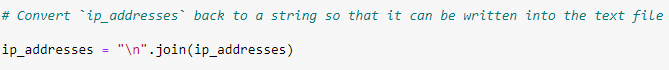
## Remove IP addresses that are on the remove list

The algorithm requires removing IP addresses from the allow list, ip\_addresses, that are also found in the remove\_list. Since no duplicates are found in ip\_addresses I was able to use the following code:



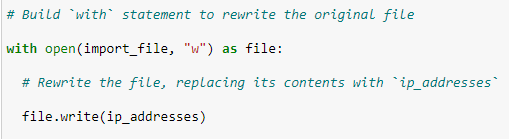
Firstly, within the for loop, I created a conditional statement that checked if the variable element was found in the ip\_addresses list. Within that conditional statement I applied .remove() to ip\_addresses. I passed in the loop variable element as the argument so that each IP address that belongs to remove\_list would be removed from ip\_addresses.

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

The final step of the algorithm revises the list of IP addresses. I needed to update the allow list file. To do this, I needed to first convert the last back into string format. I used the .join() method:   


The .join() method combines items in an iterable into a string. In this algorithm, I used the .join() method to create a sting out of ip\_addresses so then I could have an argument to pass into the .write() method when writing to the “allow\_list.txt” file. The (“\n”) string is used as a separator to instruct Python to place each element on a new line.

Then I used another with statement with the .write() method to update the file:



This time I used a second argument of “w” with the open() function in my with statement. This argument states I want to write over the file contents.  
  
I wanted to write the updated allow list as a string to the file “allow\_list.txt”. The restricted content could no longer be accessed by any IP addresses that were removed from the allow list. I appended the .write() function to the file object file that I identified in the with statement. I passed the ip\_addresses variable as the argument to specify that the contents of the file in the with statement should be replaced with the data in the ip\_addresses varialbe.

## Summary

I crafted an algorithm that removes IP addresses that are specified in a remove\_list variable from “allow\_list.txt” file of approved IP addresses. This algorithm opened the file, converting it to a string to be read. After that, it was converted to a list stored in the variable ip\_addresses. I then iterated through all of the IP addresses in remove\_list. As iteration took place, each ip address was evaluated to see if it was an element of the ip\_addresses list. If it was, I applied the .remove() method to remove it from the ip\_addresses list. Lastly, I used the .join() method to convert the ip\_addresses back into a string so that I could write over the contents of the “allow\_list.txt” file with the revised list of ip addresses.