# Python Algorithm for file updates

## Project description

For this project, I created an algorithm that automates updating files used to control access to resources. “allow\_list.txt” identifies IP addresses that are allowed access while “remove\_list” contains restricted IP addresses.

## Open the file that contains the allow list.

A close-up of a computer screen

Description automatically generated

A close-up of a white background

Description automatically generated

First, I assigned the *“allow\_list.txt”* as a string to the variable *import\_file.* I opened the file by running a *with* statement with the *open()* function command as shown above. In the code, the first parameter import\_file identifies the file to import, and then the second “r” indicates what I want to do with the file which in this case is read. The code also used the *as* keyword to store the output of the open() function as a variable named *file* while within the with statement. The *with* statement will help manage the resources by closing the file after exiting the statement.

## Read the file contents.

A screenshot of a computer

Description automatically generated

In order to read the contents of the file outside the with statement, I used the *.read()* function in the body of the with statement to convert the content into a string and assign it to the variable *ip\_addresses.*  I used the *print()* function to display the contents as seen in the screenshot above.

## Convert the string into a list

A white screen with black text

Description automatically generated

I used the *.split()* function to convert the string output into a list by extracting individual IP addresses. In this code, the *.split()* function converts the string of IP addresses separated by whitespaces stored in the variable *ip\_addresses*, into a list of IP addresses and reassigns them to the same variable. This conversion makes it easier to work with the data in the algorithm.

## Iterate through the remove list

A screenshot of a computer

Description automatically generated

An essential part of my algorithm involved iterating through the remove\_list for restricted IP addresses. To achieve this, I used the *for* statement and *in* keyword to iterate every loop variable assigned as *element* in the remove\_list in a sequence. This function iterates and assigns a value to every loop variable in the list sequentially.

## Remove IP addresses that are on the remove list

A screenshot of a computer

Description automatically generated

My algorithm requires removing any IP address contained in the ‘remove\_list’ from the ‘allow\_list’ which has been assigned the variable ‘ip\_addresses’. I used a conditional *if* statement within the body of the *for* loop function to evaluate the loop variable ‘element’ against the contents of the ‘ip\_addresses’ variable. Then, within the conditional, I applied the .*remove()* function to the ‘ip\_asddresses’ variable to remove any ‘element’ found in it that is also in the ‘remove\_list’.

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

## A screenshot of a computer Description automatically generated

Finally, I updated the allow list file with the revised list of IP addresses. To achieve this, I first converted the content of the ‘ip\_addresses’ list back to a string using the .*join()* method. The function “ip\_addresses = “ “.join(ip\_addresses)” seen in the screenshot above iterates through the list, joins all the elements into a long string with whitespaces in between the elements and reassigns the string to the variable ‘ip\_addresses’. Using the *with open()* statement and *as* keyword, I reopened the original file and assigned the variable *file* to the output within the statement. The parameter “w” indicates that I want to overwrite the content of the original file. I used the code “file.write(ip\_­addresses)” within the body of the *with* statement to overwrite the content of the opened original file with the revised list assigned the variable ‘ip\_addresses’.

Then, I used another *with open()* statement to read the updated file and assigned the output the variable *text.* The updated string is seen displayed as the output of the *print() function.*

## Define a function named `update\_file` that takes in two parameters: `import\_file` and `remove\_list`

A screenshot of a computer

Description automatically generated

To finish the project, I defined a function for the algorithm named ‘*update\_file’*  that takes in two parameters ‘import\_file’ and ‘remove\_list’.

## Summary

I created an algorithm that removes IP addresses identified in a remove\_list variable from the "allow\_list.txt" file of approved IP addresses. This algorithm involved opening the file, converting it to a string to be read, and then converting this string to a list stored in the variable ip\_addresses. I then iterated through the IP addresses in remove\_list and evaluated the element against the content of the ip\_addresses list. I applied the .remove() method to it to remove the elements common to both lists. After this, I used the .join() method to convert the ip\_addresses back into a string so that I could write over the contents of the original "allow\_list.txt" file with the revised list of IP addresses.