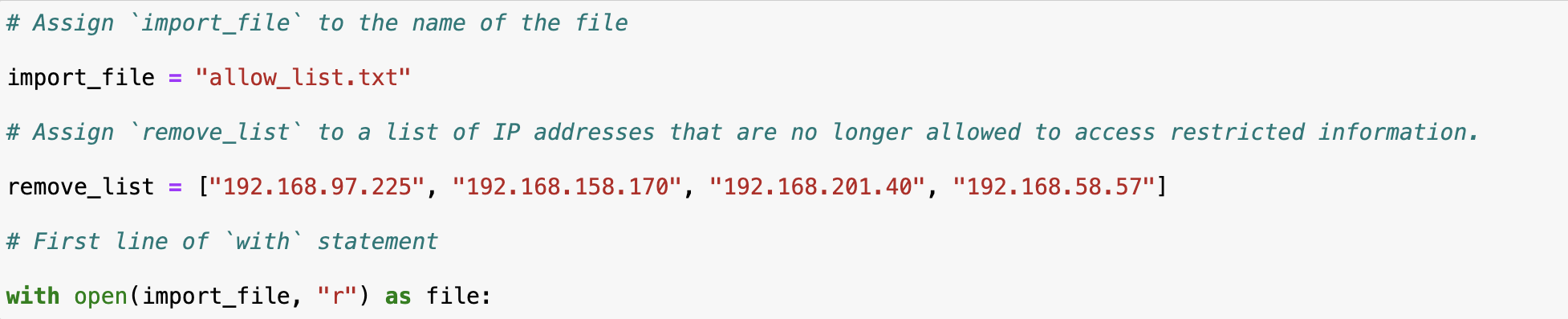
# Algorithm for file updates in Python

## Project description

As a security professional working at a health care company, I am tasked with creating an algorithm with Python to check whether the allow list contains IP addresses that exist on the remove list. The algorithm will work in such a way that any IP addresses on the remove list will be removed from the allow list so as to restrict access to certain content from the employees that don’t need to access it.

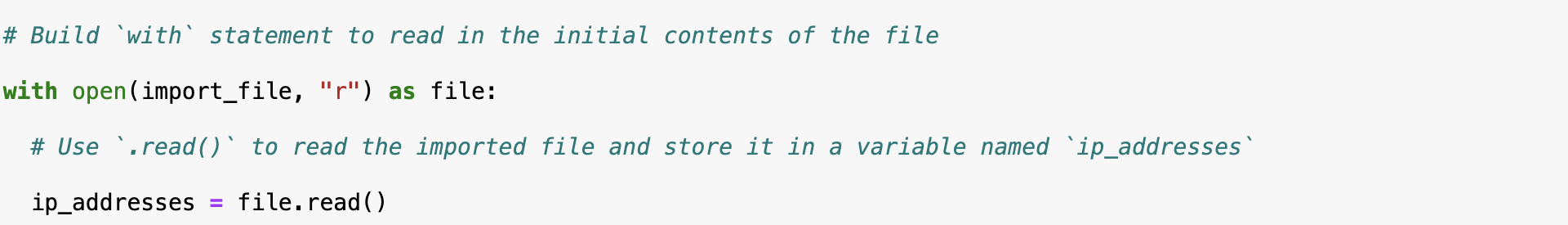
## Open the file that contains the allow list

To initiate the algorithm I assigned “allow\_list.txt” as a string to the variable import\_file. I then assigned the corresponding IP addresses to remove\_list. I used a with statement because it will allow me to manage the resources within the statement after closing it. In order to open the file containing the allow list I used a with statement paired with the .open() function as well as “r” to put it in read mode since we don’t want to make any changes to it just yet. The .open() function contains two parameters. The variable import\_file is the file I want to import and the second parameter “r” is stating I want to read the file. Then using “as” to assign a variable named file, I am able to store the output of the .open() function while working inside the with statement.



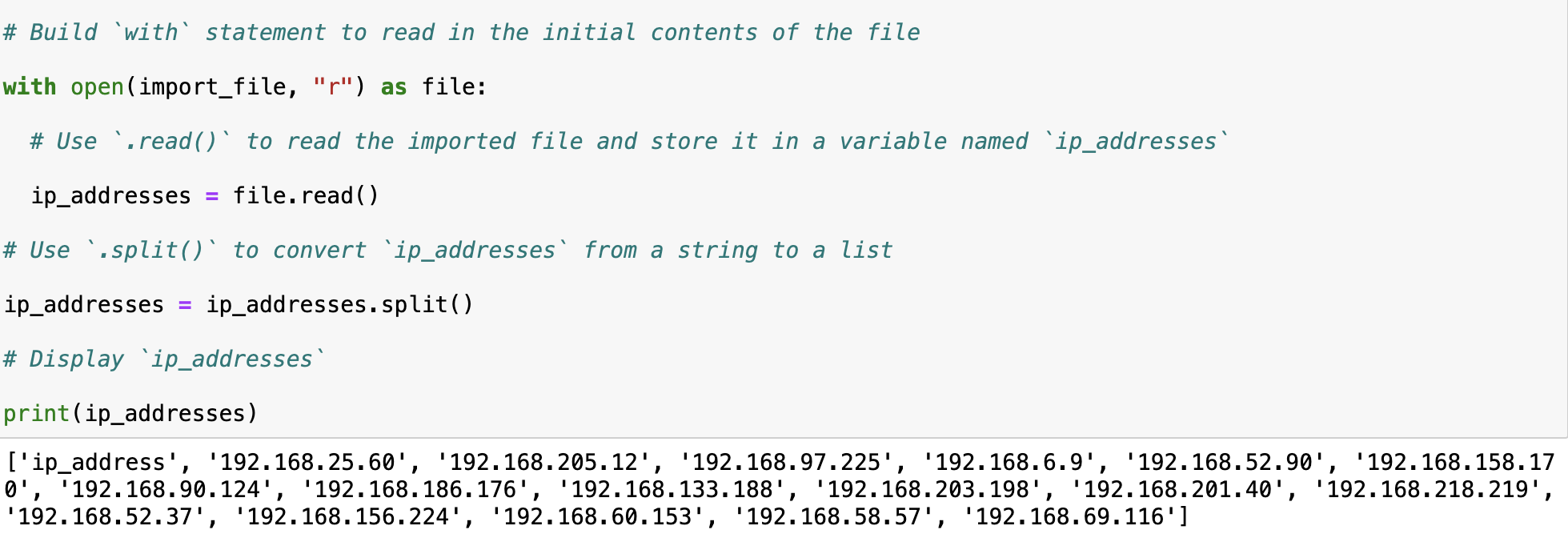
## Read the file contents

To read the file contents I used the .read() method to convert the file into a string. Since I had assigned the contents of the with statement to the variable named file, I paired this with the .read() method to output those files as a string to use later in my algorithm under the variable ip\_addresses.



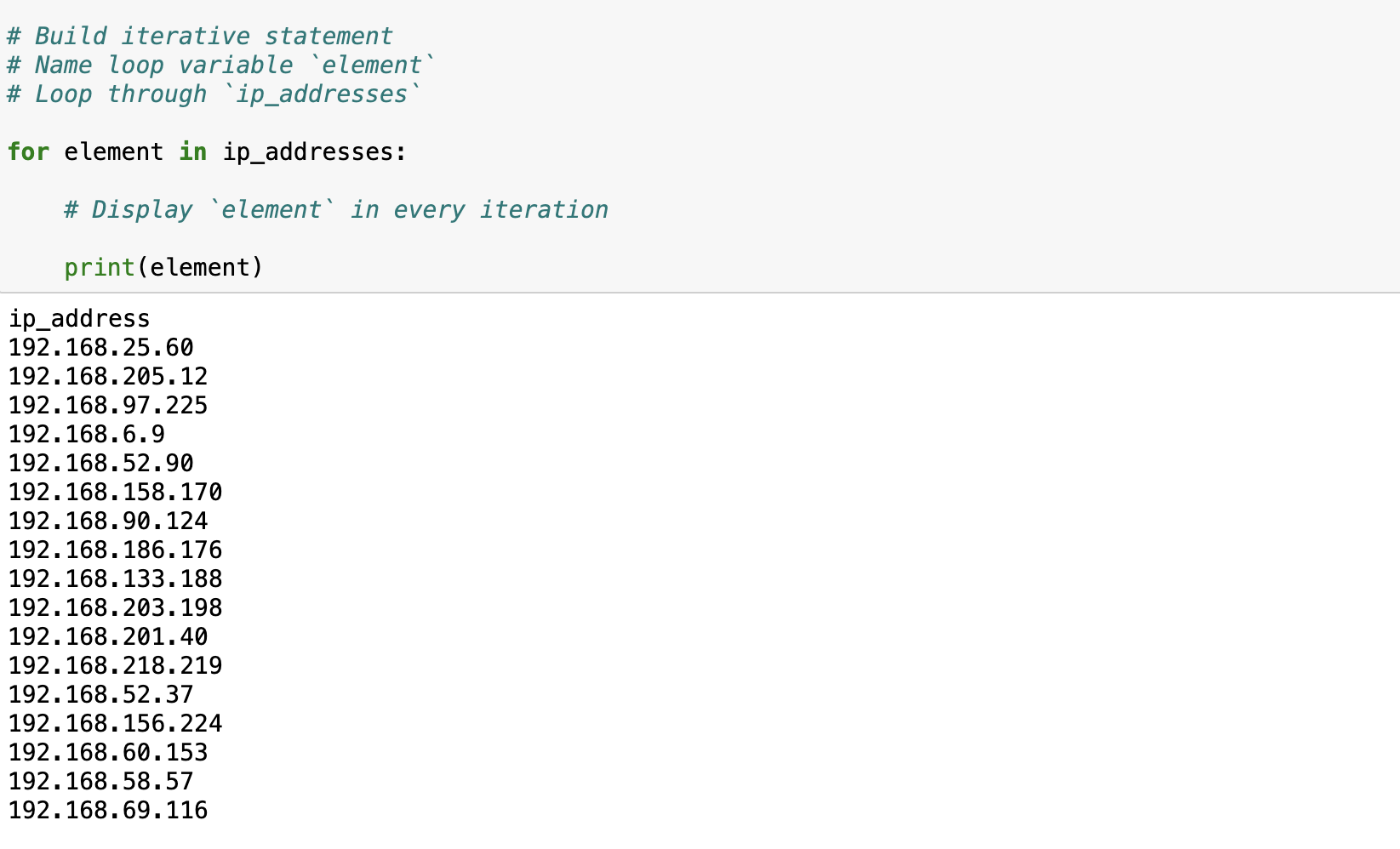
## Convert the string into a list

In order to make the contents of the allow\_list.txt more readable I used the .split() method to convert the string into a list. You can use the .split() function by adding it to the end of a variable, in this case ip\_addresses, so that we can call upon the list later on in the algorithm.



## Iterate through the remove list

In order to iterate through the remove\_list I used a for loop. The purpose of the for loop is to apply the same code sequence to each item, or element in this case, in the sequence. The in operator is used to indicate that the loop will iterate through every item in ip\_addresses and assign a value to the loop variable element. As you can see below it prints each IP address in the list.



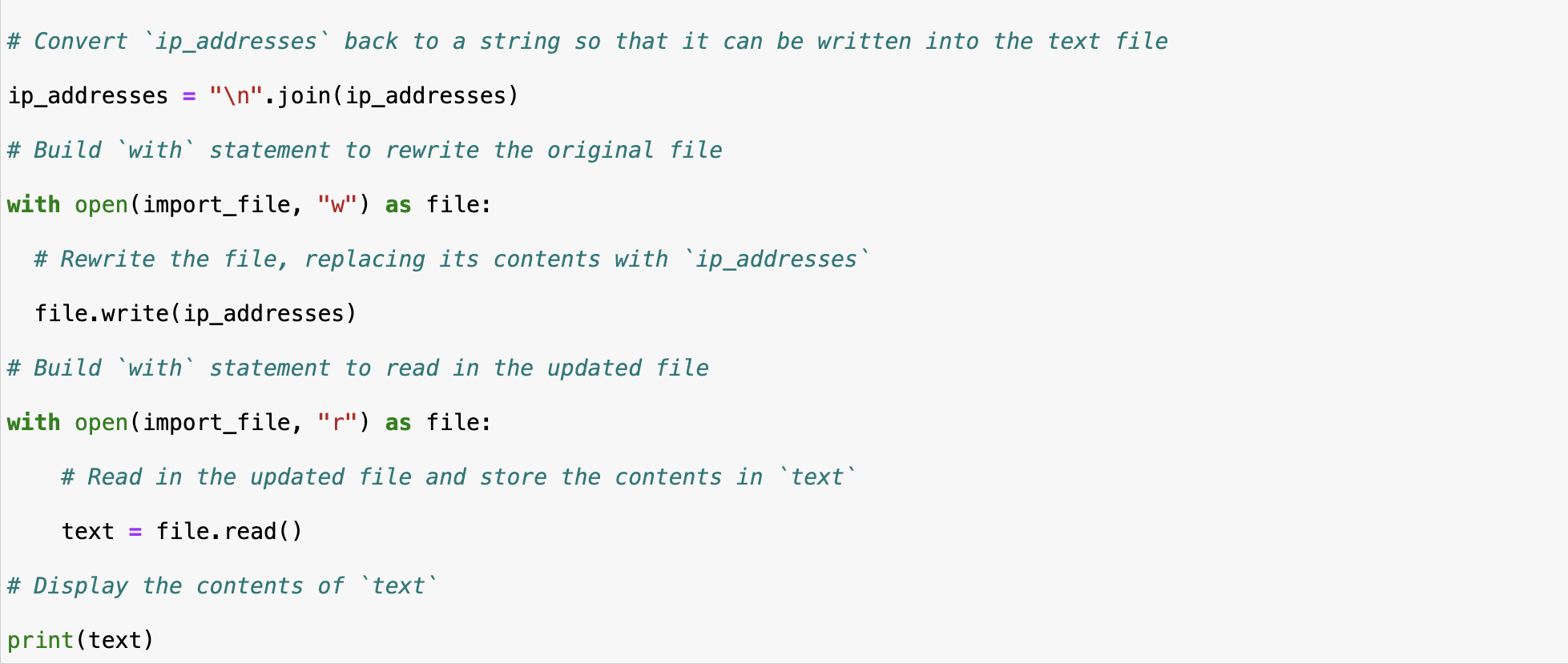
## Remove IP addresses that are on the remove list

Since we want to remove certain IP addresses from the allow\_list we have to create an if statement to tell the code to remove any IP address that exists within the remove\_list that is also in the variable ip\_addresses. For the first line of code we first state to look for the loop variable element within ip\_addresses. For the second line of the code we are essentially telling the code to do something if the loop variable element exists within the remove\_list. For the third line of code we use the .remove() function to remove any element that exists within ip\_addresses that is also in the remove\_list. We must put the third line after the if statement because if not it would produce an error since it would be looking to remove an element before a rule was set.



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

In order to update the file with the revised list of IP addresses I first started by converting ip\_addresses into a string so that I could write into the text file. To do this I used the .join() method and using the string “\n” to separate elements in the file by placing them on a new line. The .join() method is used to take elements from an iterable into a string. Since we want to rewrite the file we need to convert it into a string. The next line of code we used a with statement and referenced back our original file import\_file along with “w” and the open() function to indicate that we want to write over the contents of the file. Since we identified the file we want to write over as “file” I then used the .write() function on the next line with ip\_addresses as the argument to replace the files that were originally in the allow\_list.txt.



## Summary

To summarize, I created an algorithm that removed IP addresses from allow\_list.txt that existed within the remove\_list variable. I first opened the original file and converted it into a string so we could see the contents of the list. I then converted it into a list and assigned it to the variable ip\_addresses so that we could iterate through the IP addresses in the remove\_list. If an IP address was in the remove\_list I then used the .remove() method to remove the element from ip\_addresses. Finally I used the .join() method to convert the list back into a string so that I could write over the original file and update allow\_list.txt to the newly revised list of approved IP addresses.