# Algorithm for file updates in Python

## Project description

I am creating an algorithm in Python to automate the process of updating a file that contains the allowed IP addresses for accessing restricted content pertaining to sensitive personal patient data of a healthcare company.

## Open the file that contains the allow list

To open the file in Python I would use the following code:

**import\_file = “allow\_list.txt”**

**with open(import\_file, “r”) as file:**

For the purposes of the demonstration the “allow\_list.txt” file is held in the variable **import\_file** to represent what would normally be a passed argument to a function holding the algorithm.

## Read the file contents

For Python to be able to read the contents of the allow list file I would use the following code:

**import\_file = “allow\_list.txt”**

**with open(import\_file, “r”) as file:**

**ip\_addresses = file.read()**

I would assign the file.read() to the variable **ip\_addresses** to make it easier to use the data stored elsewhere in the program.

## Convert the string into a list

To convert the string generated by the read() method into a list I would use the split() method:

**import\_file = “allow\_list.txt”**

**with open(import\_file, “r”) as file:**

**ip\_addresses = file.read()**

**ip\_addresses = ip\_addresses.split()**

Since no argument was passed to the split method it will break up the string into a list by every whitespace it encounters. I reassigned the **ip\_addresses** variable to hold the new generated list.

## Iterate through the IP addresses list and remove IP addresses that are on the remove list

To iterate through the IP addresses list I would utilize a for loop and to ensure that the IP addresses in the remove list are removed I would use a conditional. The following code demonstrates the two processes together:

**for element in ip\_addresses:**

**if element in remove\_list:**

**ip\_addresses.remove(element)**

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

To update the file with the new correct list of allowed IP addresses I would use the following code:

**ip\_addresses = “ “.join(ip\_addresses)**

**with open(import\_file, “w”) as file:**

**file.write(ip\_addresses)**

## Summary

I would combine the algorithm displayed into a function so that it can be used multiple times with different files and remove lists. Produced by this program will be a new updated allow list of IP addresses that can access restricted data and IP addresses that are no longer allowed were automatically removed from the provided file.