# Algorithm for file updates in Python

## Project description

I am working in a health-care company. One of my day to day activities at work is to regularly update a file that identifies the employees who can access restricted content. Employees are restricted access based on their IP address. My task is to create an algorithm that uses python code to update the file.

## Open the file that contains the allow list

First, I started by assigning the text file using the `import_file` variable to the name of the file in string.
The code I used is `import_file = "allow_list.txt"`
Then, I used the `with` statement to open it, also using the letter `"r"` as the second parameter to read the file.
The code I used is `with open(import_file, "r") as file:`

## Read the file contents

To read the file, I used the `.read()` method to read the `import_file` and store the string in the variable named `ip_addresses`.
The code I used is `ip_addresses = file.read()`

## Convert the string into a list

In order to remove individual ip addresses from the allow list, I used the `.split()` method to convert the `ip_addresses` from string to list.
The code I used is `ip_addresses = ip_addresses.split()`

## Iterate through the remove list

Here, I assigned the second list called the `remove_list` that contains all the ip addresses that should be removed from the `ip_addresses` list
The code I used is `remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]`
Then next I used a `for` loop to iterate through the `ip_addresses` to remove list Using `element` as a loop variable and also using the `in` as a loop condition.
The code I used is `for element in ip_addresses:`

## Remove IP addresses that are on the remove list

In this step, I used the `for` loop to iterate through the `remove_list` and then remove ip addresses there. Here, I used `element` as a loop variable and used the `in` as a loop condition
The code I used is `if element in remove_list:`
Then I applied the `.remove` method to the `ip_addresses` list to remove ip addresses identified in the loop variable `element`,
The code I used is `ip_addresses.remove(element)`

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

Lastly, I converted the `ip_addresses` list back to a string using the `.join()` method so that it can be written into the text form
The code I used is `ip_addresses = " ".join(ip_addresses)`
Then, I used another `with` statement and also letter "w" to write over the file assigned to the `import_file` variable
The code I used is `with open(import_file, "w") as file:`

## Summary

I was able to use different python codes using Algorithm to swiftly perform tasks which includes updating the file that identifies the employee who can access the restricted file, and also use remove_list to identify which employee to remove from the allow list. Thereby, protecting the content of the file and allowing the employees who are working with personal patient records access.