Hello again, I’m Iustinian Andronie and I’ve finished the task. Now I want to explain some things.

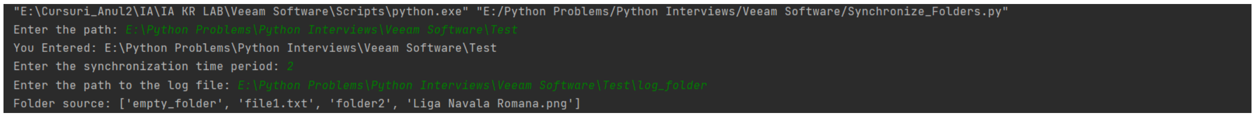
The task I chose is from Internal Development in QA (SDET) Team, and I solved it in Python. The task was to implement a program that will synchronize two folders named ‘source’ and ‘replica’. The synchronization must come from the source folder to the replica folder, meaning that replica folder’s contents must be deleted and updated as such this folder will look like the ‘source’ folder.

It said that synchronization must happen periodically at a specified time that will be given in the input by the user. And lastly, all the operations: creation/copying/removal that can happened to files and folders must be logged out in a file who will be created on the path given by the user.

The libraries that I used in solving this task are: “os” (for files and folders operations), “shutil” (for copying files) and “time” (for the interval)

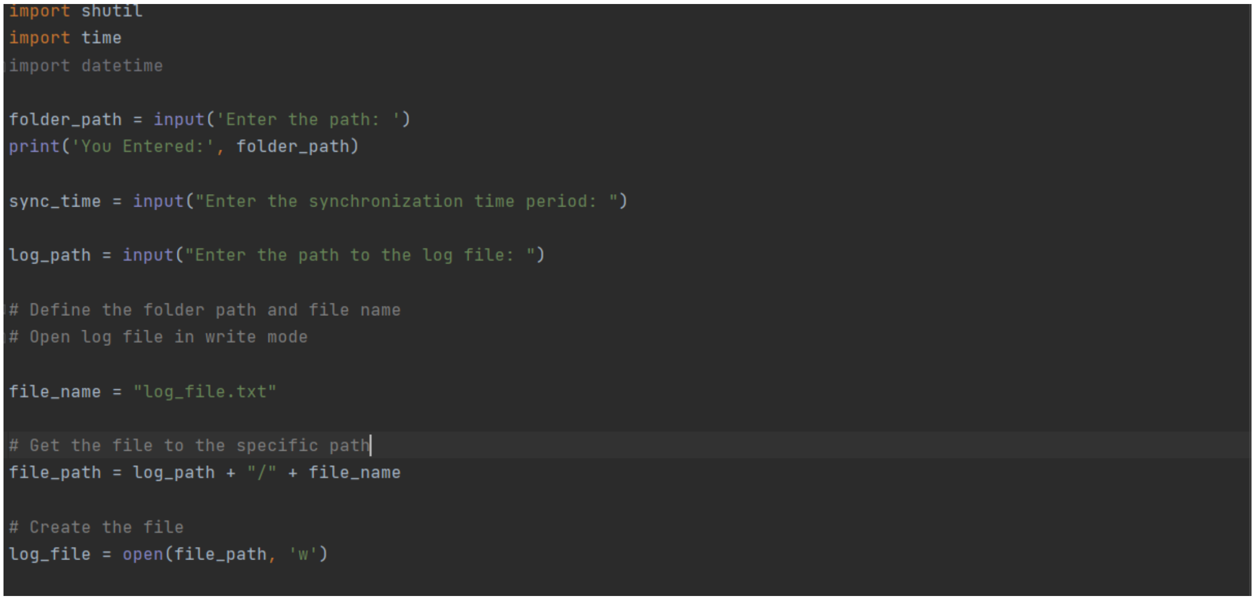
The program that I created runs correctly for cases when we have multiple types of files (excel, images, txt etc) and folders (empty or with subfolders). Even when we have content in a subfolder of the “replica” directory which should not be there, the program will delete from the subfolders what doesn’t need to be there, and copy and paste what should be.

When we run the program, we need to copy the path to the ‘source’ and ‘replica’ folders, enter the synchronization time and the path in which we will have our log file, like in this picture:



```
"E:\Cursuri_Anul2\IA\IA KR LAB\Veeam Software\Scripts\python.exe" "E:/Python Problems/Python Interviews/Veeam Software/Synchronize_Folders.py"
Enter the path: E:\Python Problems\Python Interviews\Veeam Software\Test
You Entered: E:\Python Problems\Python Interviews\Veeam Software\Test
Enter the synchronization time period: 2
Enter the path to the log file: E:\Python Problems\Python Interviews\Veeam Software\Test\log_folder
Folder source: ['empty_folder', 'file1.txt', 'folder2', 'Liga Navala Romana.png']
```

All these operations happen in the code in these first lines:



```python
import shutil
import time
import datetime

folder_path = input('Enter the path: ')
print('You Entered:', folder_path)

sync_time = input("Enter the synchronization time period: ")

log_path = input("Enter the path to the log file: ")

# Define the folder path and file name
# Open log file in write mode

file_name = "log_file.txt"

# Get the file to the specific path
file_path = log_path + "/" + file_name

# Create the file
log_file = open(file_path, 'w')
```

Now things get interesting, the program will solve the synchronization and output in the console, and in the log file what has been deleted/created/copied, and from where.

When I tried to do these operations at first, I made a for loop, to get every content from the source directory and copy at the replica directory. It was a good start to see that the program works till now. I created in the source directory one txt file and one folder, and they were copied to the replica directory as I predicted.

But then, I made a change that it was no longer solved by the program, and that was creating another folder in that first folder in the source directory. This made me brainstorm the endless possibilities that could arise such as: having an enormous number of subfolders in subfolders of the folder in the source directory. The fact that I would need a program that will go to every subfolder of the first folder and check it for other subfolders made me think of a recursive function.

So I was thinking that now I need that for loop to happen at every sublevel of the source directory.

I do this by the function named "folder_synchronization" which has two parameters: 'source_dir' and 'replica_dir' which have the paths of the source folder and replica folder. I want to have these parameters in the function because I need to keep track of the path in which I check for subfolders. For example, if I have a subfolder with content in the source directory, for this content to be copied in the replica directory, I need to have the path created first for the subfolder and then to move the content in it.

In the function, when looping through every item from the source folder, I need to check if it's a file or a folder, because copying each to the replica directory will be different. After checking, I need to find out if that item isn't already in the replica folder, because this folder can have it already from the first time it was created.

If the item is a folder, I need to check if it is empty or not, because for an empty folder, things are easy, I just create an empty folder and go to the next item from the same level. If it has content, then I need to call the function again, but with updated paths for the "source" and "replica" directories.

Another thing that I've almost forgot to do when I solved this problem was checking if the "replica" folder has already content which needs to be deleted, meaning that this content doesn't exist in the "source" folder, or isn't at the right level in the folder (has different path after "replica/"). For the delete, I just verify if it's a file or folder, because for the folder case, we need to remove the entire "tree" with "shutil.rmtree(path)".

When running the program, the console will notify the user if the synchronization time passed so we know when the operations happened. To stop the infinite loop, we must write 'f' in the console, and to let it run we must press 'Enter'.