Automation Assignment : 10

1).Design automation script which accept directory name and file extension from user. Display all files with that extension.
Usage : DirectoryFileSearch.py "Demo" ".txt"
Demo is name of directory and .txt is the extension that we want to search.

Script:

in this program we are taking two command line argv as directory name and extention of file given helper fuction will give us all .extention files from the currect directory

exm: dir
a.c
b.c
d.c

a.js
b.js
c.js

our extionsion is .c
Search--> output: a.c b.c d.c

'''

```
from sys import *
import os

def sameExtension(path, extension):


        flag = os.path.isabs(path);     # this fun us to check given parameter has abs path or
not


        if flag == False:                       # if flag is False then given path is not abs then
we need to modify with below
                path = os.path.abspath(path); #path conversion

        exists = os.path.isdir(path); # after modification if path is conts directory it contaisn
True else false

        if exists:
                print("Given path is valid");
```

```
                for folder, subFolder, fileList in os.walk(path):
                        #print("Current Directory Name : ",folderLst);
                        for filename in fileList:
                                if filename.endswith(extension):
                                        print(os.path.join(folder,filename));

        else:
                print("Given path is invalid");



def main():

        print("Application Name:"+argv[0]); #this will display our file name
        print("Length of given cmd arg",len(argv));

        if len(argv)!=3:
                print("Given input is Invalid");
                exit();

        if argv[1]=="-h" or argv[1]=="-H":
                print("This script is used to display perticular format files");
                exit();

        if argv[1]=='-u' or argv[1]=='-U':
                print("Usage: Application is used to Display same format files from given
Directories");
                exit();

        try:
                sameExtension(argv[1],argv[2]);

        except ValueError :
                        print("File Nof found");

        except Exception:
                print("invalid input",Exception);


if __name__ == "__main__":
        main();

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

'''
```

2. Design automation script which accept directory name and two file extensions from user. Rename all files with first file extension with the second file extenntion.

Usage : DirectoryRename.py 'Demo' '.txt' '.doc'
Demo is name of directory and .txt is the extension that we want to search and rename with .doc.
After execution this script each .txt file gets renamed as .doc.

to write script we need to take 3 paramerts

Demo directory
.txt file extension
.doc file extension

wherever we get file with .txt file that we have to convert in .doc format

'''

```
import os
from sys import *


def File_Extension_Modification(path, old_ext, new_ext):

        flag = os.path.isabs(path);

        if flag == False:
                path = os.path.abspath(path);

        exists = os.path.isdir(path);

        if exists:

                for folder, subFolder, fileList in os.walk(path):
                        for filename in fileList:
                                if filename.endswith(old_ext):
                                        #nf = Path(filename).stem + new_ext;
                                        if not os.path.isdir(filename):
                                                filename = os.path.join(path,filename);

                                        newpath = os.path.abspath(filename);
                                        naav, sheput = os.path.splitext(newpath);
                                        os.rename(newpath, str(naav)+new_ext);
        else:
                        print("invalid input");

        print("Successfully Changed")


def help():
        print("Input should like: DirName .old_ext .new_ext");
```

```
def main():
	print("Application Name: ",argv[0]);

	if len(argv)!= 4:
		print("Invalid input");
		help();
		exit();

	if argv[1]== 'h' or argv[1]== 'H':
		print("This script is use to rename the file extension");
		print("Input should like: DirName .txt .pdf");
		exit();

	if argv[1]== 'u' or argv[1]== '-U':
		print("Usage: Application Req Abspath then we gets files to modify");
		exit();

	try:

		File_Extension_Modification(argv[1],argv[2],argv[3]);

	except Exception:
		print("Error found",Exception);


if __name__ == "__main__":
	main();
```

//

'''

3.Design automation script which accept two directory names. Copy all files from first directory
into second directory. Second directory should be created at run time.
Usage : DirectoryCopy.py Demo Temp
Demo is name of directory which is existing and contains files in it. We have to create new Directory as Temp and copy all
files from Demo to Temp.

input : DirName1 DirName2
DirName2 shoude create at run time and copy files from DirName1

'''

```
import os
import sys
import shutil
```

```
def copytree(src, dst, symlinks=False, ignore=None):
    if not os.path.exists(dst):
        os.makedirs(dst)
    for item in os.listdir(src):
            if os.path.isfile(item):
            s = os.path.join(src, item)
            d = os.path.join(dst, item)
            if os.path.isdir(s):
                copytree(s, d, symlinks, ignore)
            else:
                if not os.path.exists(d) or os.stat(s).st_mtime - os.stat(d).st_mtime > 1:
                    shutil.copy2(s, d);


def main():

            print("Application Name: ",sys.argv[0]);

            try:
                        copytree(sys.argv[1],sys.argv[2]);
            except Exception as e:
                        print("Error Found",e);


if __name__ == "__main__":
            main();

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
'''
```

4. Design automation script which accept two directory names and one file extension. Copy all files with the specified
extension from first directory into second directory. Second directory should be created at run time.
Usage : DirectoryCopyExt.py Demo Temp .exe
Demo is name of directory which is existing and contains files in it. We have to create new Directory as Temp and copy all files with extension .exe from Demo to Temp.

in this script we have to take first directory name which is existing in, and one non-existing dire name and take one .extension name and copy all files from existing dir to non-existing dir with given .extension

Avail DirName
a.c
b.c
k.js

Non Avail Dir **Extension .cpp**
a.cpp
b.cpp
k.cpp
conversion should be taken place after running our script

'''

```
import sys
import shutil
import os



def copytree(src, dst, symlinks=False, ignore=None):
    if not os.path.exists(dst):
        os.makedirs(dst)
    for item in os.listdir(src):

        s = os.path.join(src, item)
        d = os.path.join(dst, item)
        if os.path.isdir(s):
            copytree(s, d, symlinks, ignore)
        else:
            if not os.path.exists(d) or os.stat(s).st_mtime - os.stat(d).st_mtime > 1:
                    if not os.path.isabs(s):
                                            s = os.path.abspath(s);
                                            shutil.copy2(s, d);




def Extension_Changer(path, ext):
        flag = os.path.isabs(path);
        if flag == False:
                path = os.path.abspath(path);

        os.chdir(path);

        for f in os.listdir(path):
                file_name, file_ext = os.path.splitext(f);
                f = os.path.join(path,f);
                os.rename(f, str(file_name)+ext);



def main():

        print("Application Name:",sys.argv[0]);

        if len(sys.argv) != 4:
                print("Invalid input");
        try:
                copytree(sys.argv[1],sys.argv[2]);
                Extension_Changer(sys.argv[2], sys.argv[3]);
        except Exception as e:
                print("Error Found ",e);
```

```
if __name__ == "__main__":
        main();
```