Project Title: Directory Organizer

This project is about using bash to organize your directories.

Basic Requirements:

You have multiple files scattered in different directories on your computer. Your task is to create a bash script (organizer.sh) that will organize these files into folders based on file type.

1. The bash script will take 2 inputs: a source directory path and a destination directory path. It will scan the given source directory and its subdirectories to identify different file types based on their file extensions (e.g., .txt, .jpg, .pdf, etc.) and then create separate folders in the destination path for each file type and move the respective files into their corresponding folders.
2. If a file with the same name already exists in the destination folder, the script should rename the file by appending an incrementing number to avoid overwriting.
   Example: If you have 2 or more files named create.txt in different sub folders in the source directory then one would be stored as create.txt and second would be stored as create_1.txt , third would be create_2.txt and so on.
3. **Exception handling should be done wherever necessary** (ex. you can check whether the destination folder exists. If not create a destination folder).
4. After running the script, a short summary of the files(number of folders created, files transferred, number of files in each folder) should be printed.
5. You have organized files in the earlier part based on file extension. Extend it to handle organization based on creation date (day, not time!). Again carefully handle file name repetition.
6. Provide an option to delete the original files after they have been organized (this should be an optional feature). This option should be taken as a command line flag while running the script. This flag can be named -d. See this link for more details.
7. You need to make sure that your code works for filenames such as <filename>.<ext> (where is ext is non-empty string) and <filename> (where filename does not have any dot). We would be checking your code only for the above mentioned filenames. Having said that, if you have already handled some different format of filenames, you can add it in as a customization.

Customization:

You can extend the bash script to add further features. Here are some examples, but really you should come up with your own. Marks will be given per YOUR creativity.

1. Add a feature to generate a log file that captures the details of every moved file (such as the source path, destination path, and timestamp of movement). Log file name should be taken by the user using -l flag.

2. Add an option to exclude certain file types or directories from being organized, based on user input or predefined rules. This can be done by say using -e flag.
3. During scanning, Zip files should be unzipped and examined and saved accordingly.
4. Files having exactly the same hash (copy of files) should be deleted.

Process:

Your script should take relevant arguments to run as per user specification.
A few examples are given below of expected usage.

1. *bash organizer.sh <srcdir> <destdir> -s ext* should organize the files based on extension
2. *bash organizer.sh <srcdir> <destdir> -s date* should organize the files based on their creation date
3. *bash organizer.sh <srcdir> <destdir> -d -s ext* should organize based on extension and also delete the original files after moving them.

4. bash organizer.sh <srcdir> <destdir> -s ext -e txt,doc should exclude all the txt and doc files. Even if -d option is given to delete the original files you should not delete the excluded extension based files

5. If the user does not give any style, the default style should be extension based

You should create the test cases yourself for all these options and test everything carefully. Note when evaluating, TAs will use their own test cases, your code should work with that!

Marks Distribution (20 marks): These marks will be awarded via a viva, where you demonstrate the project and TAs will ask questions.

- Organization based on extension: 4 marks
- Organization based on date: 2 marks
- Deleting original files: 2 marks
- Exception handling and printing of user friendly messages: 2 marks
- Customization: 4 marks
- Viva: 4 marks
- Code quality, organization, and comments, latex based report: 2 marks

Note: All submitted code in a given project will be checked for plagiarism and if caught, your case will be forwarded to DDAC.

Upload: Please upload all relevant files, including report as a zip file on BodhiTree before the deadline. NO CHANGES TO THE FILES WILL BE PERMITTED POST THIS, INCLUDING DURING EVALUATION. During evaluation, TAs will download the zip from Bodhitree, set it up on their machine and then interact with you for evaluation. They will use their own input/testcases during this time.