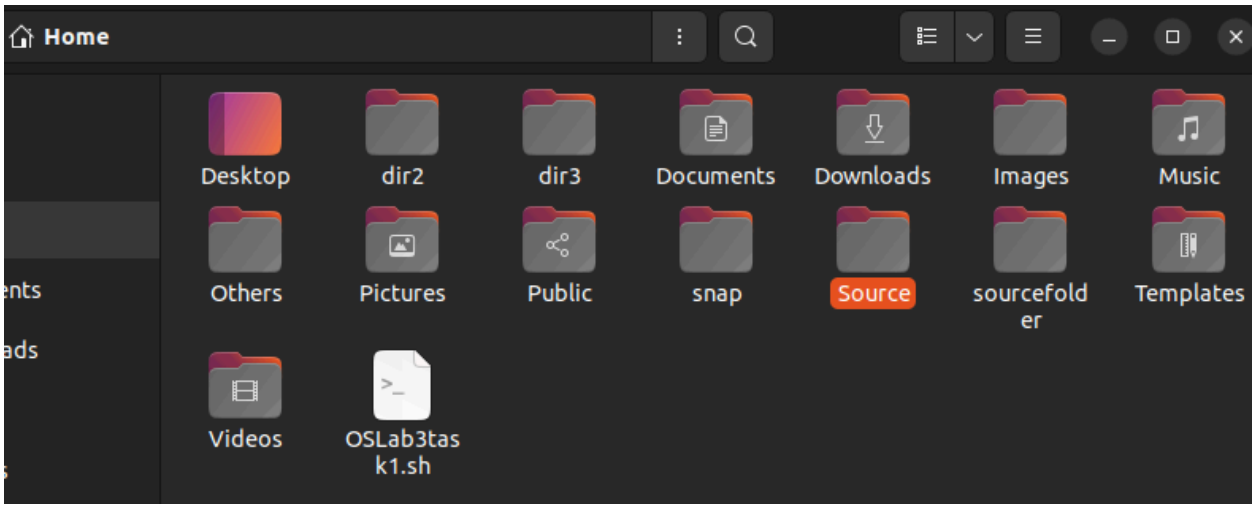
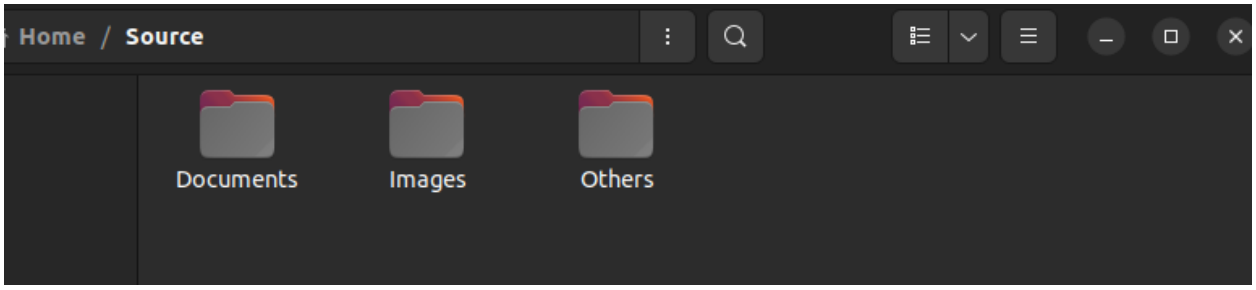
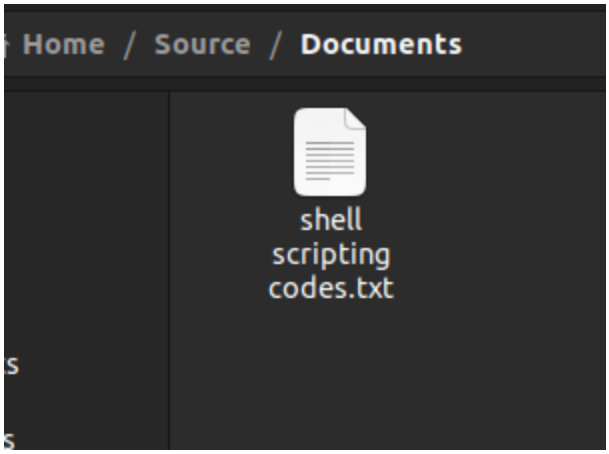
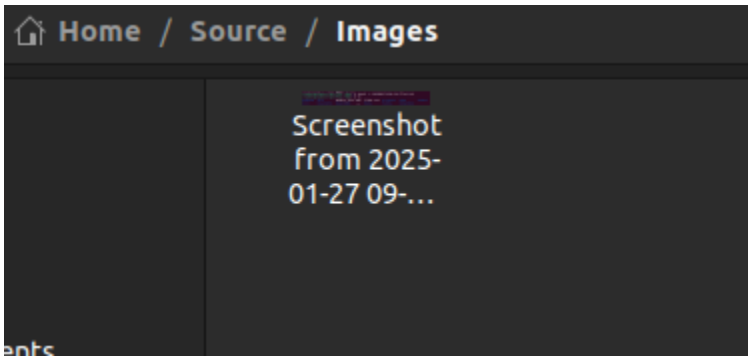
1. Write a script that moves files into separate folders based on their extensions (e.g., .jpg files into an Images folder, .txt files into a Documents folder).

```bash
#!/bin/bash

SOURCE_FOLDER="/home/student/Source"

mkdir -p "$SOURCE_FOLDER/Images" "$SOURCE_FOLDER/Documents" "$SOURCE_FOLDER/Others"

for file in "$SOURCE_FOLDER"/*; do
    if [ -f "$file" ]; then
        extension="${file##*.}"
        case "$extension" in
            jpg|jpeg|png|gif)
                mv "$file" "$SOURCE_FOLDER/Images/"
                ;;
            txt|doc|docx|pdf)
                mv "$file" "$SOURCE_FOLDER/Documents/"
                ;;
            *)
                mv "$file" "$SOURCE_FOLDER/Others/"
                ;;
        esac
    fi
done

echo "Files have been sorted."
```










```
student@student-OptiPlex-7000:~$ mkdir Source
student@student-OptiPlex-7000:~$ cd
student@student-OptiPlex-7000:~$ gedit OSLab3task1.sh
^Z
[1]+  Stopped                 gedit OSLab3task1.sh
student@student-OptiPlex-7000:~$ ./OSLab3task1.sh
bash: ./OSLab3task1.sh: Permission denied
student@student-OptiPlex-7000:~$ chmod +x OSLab3task1.sh
student@student-OptiPlex-7000:~$ ./OSLab3task1.sh
student@student-OptiPlex-7000:~$ gedit OSLab3.sh
[1]+  Killed                  gedit OSLab3task1.sh
student@student-OptiPlex-7000:~$ rm OSLab3.sh
rm: cannot remove 'OSLab3.sh': No such file or directory
student@student-OptiPlex-7000:~$ gedit OSLab3task1.sh
student@student-OptiPlex-7000:~$ chmod +x OSLab3task1.sh
student@student-OptiPlex-7000:~$ ./OSLab3task1.sh
student@student-OptiPlex-7000:~$ gedit OSLab3task1.sh
^Z
[1]+  Stopped                 gedit OSLab3task1.sh
student@student-OptiPlex-7000:~$ ./OSLab3task1.sh
bash: ./OSLab3task1.sh: Permission denied
student@student-OptiPlex-7000:~$ chmod +x OSLab3task1.sh
student@student-OptiPlex-7000:~$ ./OSLab3task1.sh
Files have been sorted.
```

2. Write a script that compresses a specified directory into a ".tar.gz" archive with a timestamp and allows restoring from a backup.

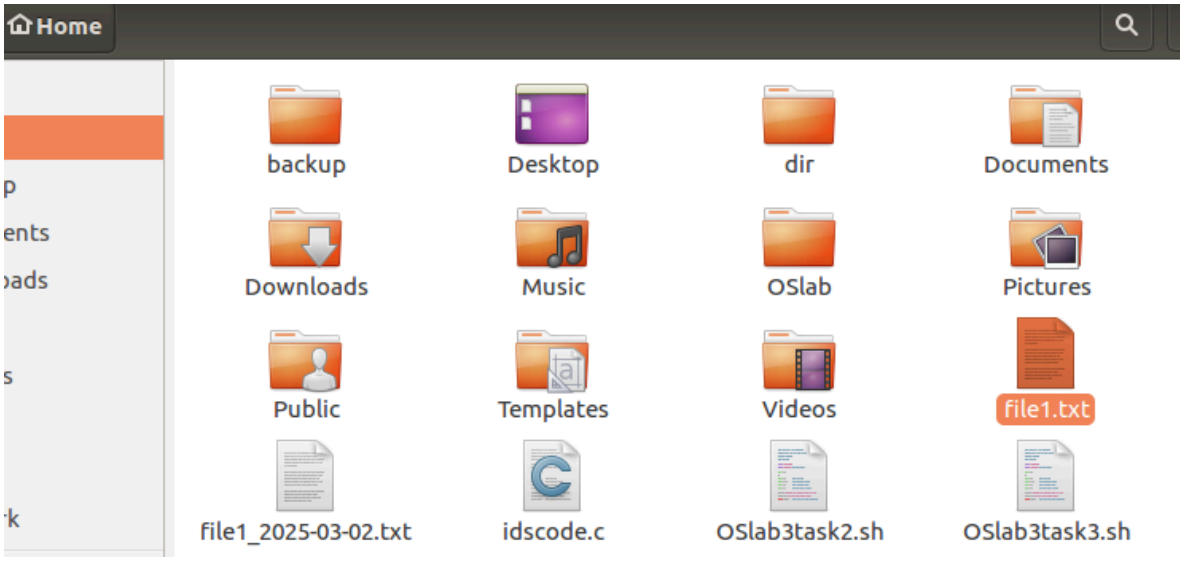
```
0281kainat@k200281kainat-VirtualBox:~$ OSlab3task2.sh
ab3task2.sh: command not found
0281kainat@k200281kainat-VirtualBox:~$ touch OSlab3task2.sh
0281kainat@k200281kainat-VirtualBox:~$ gedit OSlab3task2.sh
```

```
k200281kainat@k200281kainat-VirtualBox:~$ gedit OSlab3task2.sh
k200281kainat@k200281kainat-VirtualBox:~$ chmod +x OSlab3task2.sh
k200281kainat@k200281kainat-VirtualBox:~$ ./OSlab3task2.sh
k200281kainat@k200281kainat-VirtualBox:~$ mkdir dir
k200281kainat@k200281kainat-VirtualBox:~$ chmod +x OSlab3task2.sh
k200281kainat@k200281kainat-VirtualBox:~$ ./OSlab3task2.sh
```

```
#!/bin/bash
compress_dir(){
local dir=$*
if [ ! -d "$dir" ]; then
        echo "Error: $dir doesnot exist."
        exit 1
fi

local curr_date=$(date '+%d-%m-%Y_%H-%M-%S')
if tar czf "./$dir ($curr_date).tar.gz" $dir; then
        echo "$dir directory compressed successfully!"
else
        echo "Error: Couldnot compress $dir directory"
fi
}

extract_archive(){
local archive_name=$1
if [ ! -e "$archive_name" ]; then
        echo "ERror: $archive_name doesnot exist."
        exit 1
fi

if tar xzf $archive_name &> /dev/null; then
        echo "$archive_name archive extracted successfully!"
else
        echo "Error: Couldnot extract $archive_name archive."
fi
}
```

3. Create a script that takes a filename as input and creates a backup of the file with the current date appended to the filename.

```
k200281kainat@k200281kainat-VirtualBox:~$ touch OSlab3task3.sh
k200281kainat@k200281kainat-VirtualBox:~$ gedit OSlab3task3.sh
k200281kainat@k200281kainat-VirtualBox:~$ chmod +x OSlab3task3.sh
k200281kainat@k200281kainat-VirtualBox:~$ touch file1.txt
k200281kainat@k200281kainat-VirtualBox:~$ chmod +x OSlab3task3.sh
k200281kainat@k200281kainat-VirtualBox:~$ ./OSlab3task3.sh file1.txt
Backup created: 'file1_2025-03-02.txt'
```

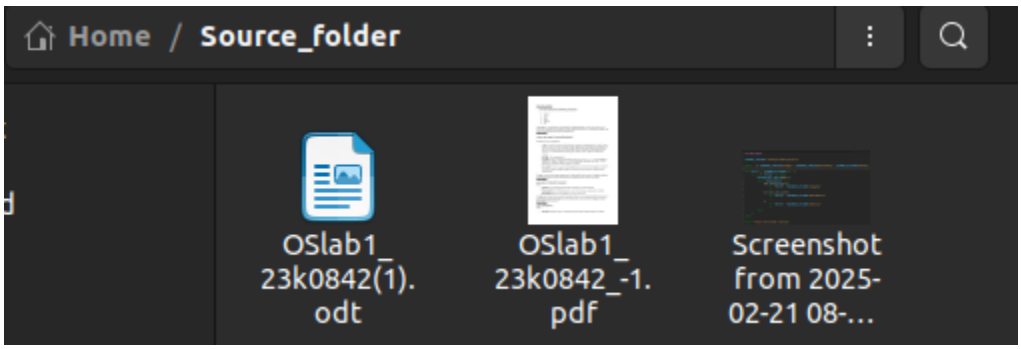



```bash
#!/bin/bash
if [ "$#" -ne 1 ]; then
    echo "Usage: $0 <filename>"
    exit 1
fi

filename="$1"

if [ ! -f "$filename" ]; then
    echo "Error: File '$filename' not found."
    exit 1
fi

base_name="${filename%.*}"
ext="${filename##*.}"

date_suffix="$(date +%Y-%m-%d)"

backup_file="${base_name}_${date_suffix}.${ext}"

cp "$filename" "$backup_file"

if [ $? -eq 0 ]; then
    echo "Backup created: '$backup_file'"
else
    echo "Error creating backup."
fi
```

4. Create a script that lists all files in a directory sorted from smallest to largest.

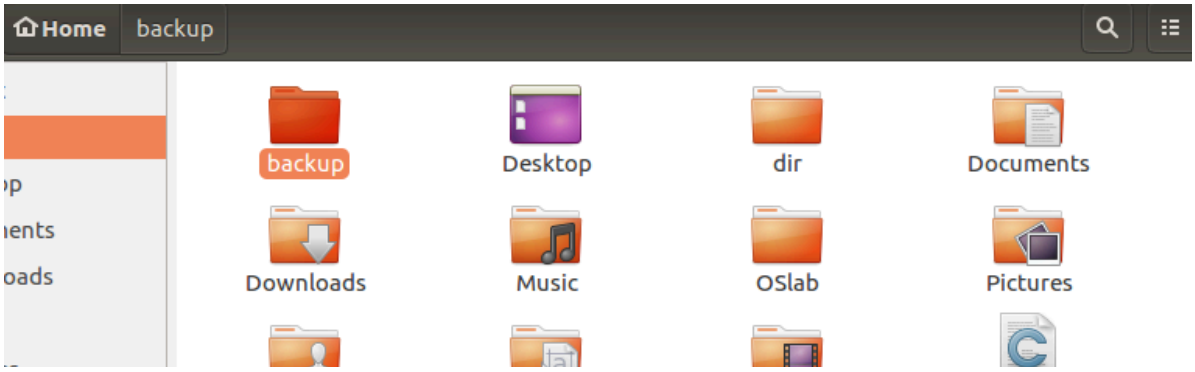
```
student@student-OptiPlex-7000:~$ touch OSLab3task4.sh
student@student-OptiPlex-7000:~$ gedit OSLab3task4.sh
student@student-OptiPlex-7000:~$ chmod +x OSLab3task4.sh
student@student-OptiPlex-7000:~$ ./OSLab3task4.sh
/home/student/Source_folder/OSlab1_23k0842(1).odt: 445K
/home/student/Source_folder/OSlab1_23k0842_-1.pdf: 303K
/home/student/Source_folder/Screenshot: 57K
```

```
#!/bin/bash

TARGET_DIR="/home/student/Source_folder"

find "$TARGET_DIR" -type f -exec ls -lhS {} + | awk '{ print $9 ": " $5 }'
```

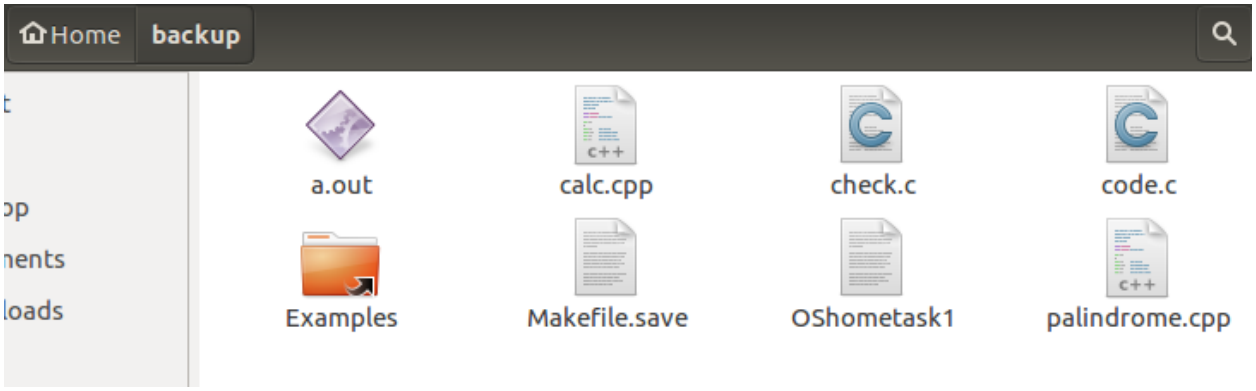


5. Write a script that moves all files older than 7 days from the current directory to a backup folder.

```
k200281kainat@k200281kainat-VirtualBox:~$ touch OSlab3task5.sh
k200281kainat@k200281kainat-VirtualBox:~$ gedit OSlab3task5.sh
k200281kainat@k200281kainat-VirtualBox:~$ chmod +x OSlab3task5.sh
k200281kainat@k200281kainat-VirtualBox:~$ ./OSlab3task5.sh
Created backup folder: backup
Moved files older than 7 days to 'backup'
```



```bash
#!/bin/bash

backup_folder="backup"

if [ ! -d "$backup_folder" ]; then
    mkdir "$backup_folder"
    echo "Created backup folder: $backup_folder"
fi

find . -maxdepth 1 -type f -mtime +7 -exec mv {} "$backup_folder" \;

if [ $? -eq 0 ]; then
    echo "Moved files older than 7 days to '$backup_folder'"
else
    echo "No files older than 7 days found."
fi
```



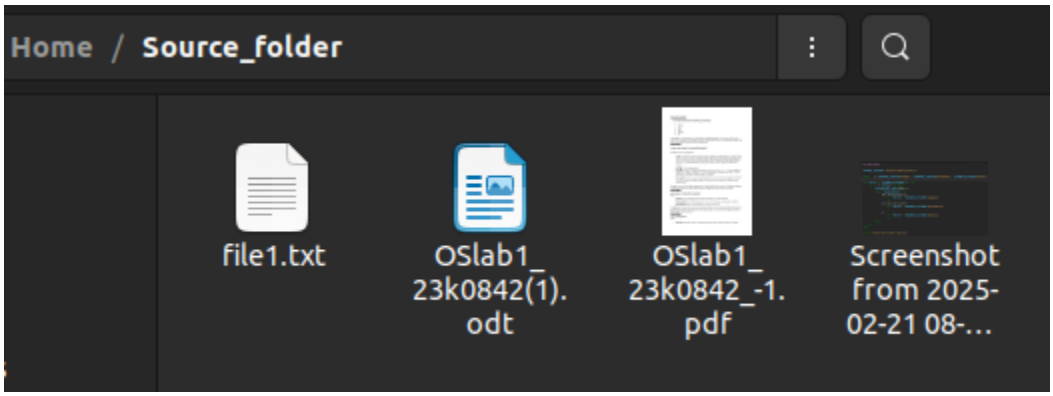
6. Create a script that finds and deletes all empty files in a directory.

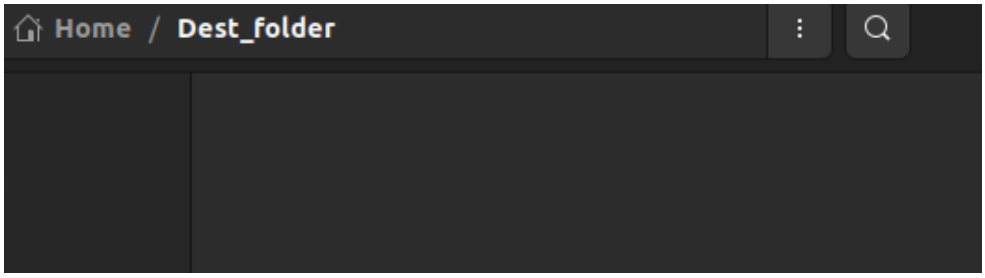
```
student@student-OptiPlex-7000:~$ touch OSLab3task6.sh
student@student-OptiPlex-7000:~$ gedit OSLab3task6.sh
student@student-OptiPlex-7000:~$ chmod +x OSLab3task6.sh
student@student-OptiPlex-7000:~$ ./OSLab3task6.sh
All empty files in /home/student/Source have been deleted.
```

```bash
#!/bin/bash

TARGET_DIR="/home/student/Source"

find "$TARGET_DIR" -type f -empty -delete

echo "All empty files in $TARGET_DIR have been deleted."

```

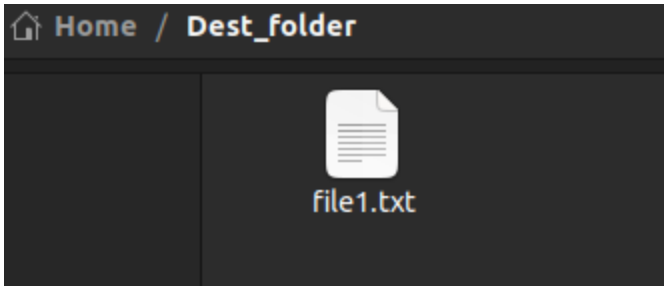
7. Create a script that copies all .txt files from one directory to another specified directory.

```
student@student-OptiPlex-7000:~$ touch OSLab3task7.sh
student@student-OptiPlex-7000:~$ gedit OSLab3task7.sh
^Z
[1]+  Stopped                 gedit OSLab3task7.sh
student@student-OptiPlex-7000:~$ chmod +x OSLab3task7.sh
student@student-OptiPlex-7000:~$ ./OSLab3task7.sh
cp: cannot stat '/home/student/Source_folder/*.txt': No such file or directory
All .txt files have been copied from /home/student/Source_folder to /home/student/Dest_folder
student@student-OptiPlex-7000:~$ gedit OSLab3task7.sh
^Z[1]   Killed                  gedit OSLab3task7.sh

[2]+  Stopped                 gedit OSLab3task7.sh
student@student-OptiPlex-7000:~$ chmod +x OSLab3task7.sh
student@student-OptiPlex-7000:~$ ./OSLab3task7.sh
All .txt files have been copied from /home/student/Source_folder to /home/student/Dest_folder
student@student-OptiPlex-7000:~$
```

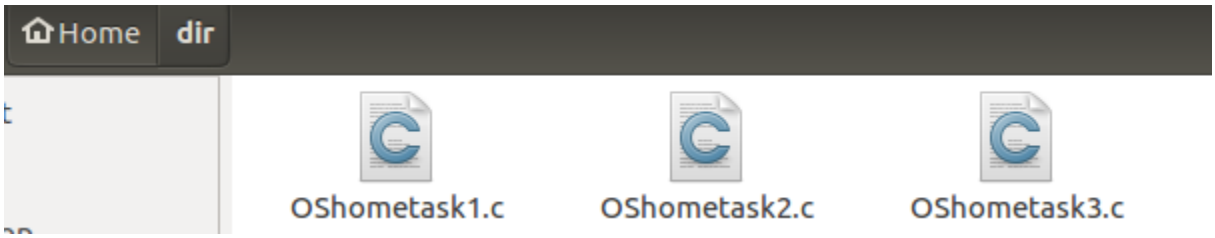





```bash
#!/bin/bash

SOURCE_DIR="/home/student/Source_folder"

DEST_DIR="/home/student/Dest_folder"

mkdir -p "$DEST_DIR"

cp "$SOURCE_DIR"/*.txt "$DEST_DIR"

echo "All .txt files have been copied from $SOURCE_DIR to $DEST_DIR"

```



8. You are tasked with creating a bash script that renames multiple files in a directory according to
a specified naming convention. The script should:
a. Accept two arguments: the directory path containing the files and the new file name pattern.
b. Rename each file in the directory by appending a sequential number to the new file name
c. pattern (e.g., `file1.txt`, `file2.txt`, etc.).
d. Preserve the original file extension during the renaming process. Provide feedback to the user about the renaming process, including any errors encountered.




```
k200281kainat@k200281kainat-VirtualBox:~$ touch OSlab3task8.sh
k200281kainat@k200281kainat-VirtualBox:~$ gedit OSlab3task8.sh
k200281kainat@k200281kainat-VirtualBox:~$ chmod +x OSlab3task8.sh
k200281kainat@k200281kainat-VirtualBox:~$ ./OSlab3task8.sh /home/k200281kainat/dir
Usage: ./OSlab3task8.sh <directory_path> <new_file_name_pattern>
k200281kainat@k200281kainat-VirtualBox:~$ ./OSlab3task8.sh </home/k200281kainat/dir> <file1>
bash: syntax error near unexpected token `<'
k200281kainat@k200281kainat-VirtualBox:~$ gedit OSlab3task8.sh
k200281kainat@k200281kainat-VirtualBox:~$ chmod +x OSlab3task8.sh
k200281kainat@k200281kainat-VirtualBox:~$ ./OSlab3task8.sh home/k200281kainat/dir OShometask1.c OShometask2.c OShome
task3.c
Usage: ./OSlab3task8.sh <directory_path> <new_file_name_pattern>
k200281kainat@k200281kainat-VirtualBox:~$ ^C
k200281kainat@k200281kainat-VirtualBox:~$ chmod +x OSlab3task8.sh
k200281kainat@k200281kainat-VirtualBox:~$ ./OSlab3task8.sh /home/k200281/dir OShometask
Error: Directory '/home/k200281/dir' not found.
k200281kainat@k200281kainat-VirtualBox:~$ chmod +x OSlab3task8.sh
k200281kainat@k200281kainat-VirtualBox:~$ ./OSlab3task8.sh /home/k200281kainat/dir OShometaskmv: '/home/k200281kaina
t/dir/OShometask1.c' and '/home/k200281kainat/dir/OShometask1.c' are the same file
Error renaming '/home/k200281kainat/dir/OShometask1.c'
mv: '/home/k200281kainat/dir/OShometask2.c' and '/home/k200281kainat/dir/OShometask2.c' are the same file
Error renaming '/home/k200281kainat/dir/OShometask2.c'
mv: '/home/k200281kainat/dir/OShometask3.c' and '/home/k200281kainat/dir/OShometask3.c' are the same file
Error renaming '/home/k200281kainat/dir/OShometask3.c'
Renaming process completed.
```

```bash
#!/bin/bash
if [ "$#" -ne 2 ]; then
    echo "Usage: $0 <directory_path> <new_file_name_pattern>"
    exit 1
fi

dir_path="$1"
new_pattern="$2"

if [ ! -d "$dir_path" ]; then
    echo "Error: Directory '$dir_path' not found."
    exit 1
fi

count=1

for file in "$dir_path"/*; do
    if [ -f "$file" ]; then
        ext="${file##*.}"
        new_name="${new_pattern}${count}.${ext}"
        new_path="$dir_path/$new_name"

        mv "$file" "$new_path"
        if [ $? -eq 0 ]; then
            echo "Renamed '$file' to '$new_path'"
        else
            echo "Error renaming '$file'"
        fi

        ((count++))
    fi
done

echo "Renaming process completed."
```