

Name: Md. Waliul Islam Rayhan

ID: B190305034

```
# Function to display the menu options
```

```
display_menu() {  
    clear  
    echo "Welcome to File and Directory Management Shell Scripting System!"  
    echo "=====  
    echo "1. Move Files"  
    echo "2. Rename Files"  
    echo "3. Create Directories"  
    echo "4. Delete Directories"  
    echo "5. Delete Files"  
    echo "6. Delete Old Files"  
    echo "0. Exit"  
    echo  
}
```

```
# Function to handle moving files
```

```
move_files() {  
    echo "Move Files"  
    echo "~~~~~"  
    echo "Please enter the source file path:"  
    read source_file  
    echo "Enter the destination directory path:"  
    read destination_dir
```

```
# Check if source file exists
```

```
if [ ! -f "$source_file" ]; then  
    echo "Ops! Source file '$source_file' not found!"  
    return  
fi
```

```
# Check if destination directory exists
```

```
if [ ! -d "$destination_dir" ]; then  
    echo "Ops! Destination directory '$destination_dir' not found!"  
    return  
fi
```

```
# Move the file
```

```
mv "$source_file" "$destination_dir"
```

```

    echo "Congratulations! File has been moved successfully!"
}

# Function to handle renaming files
rename_files() {
    echo "Rename Files"
    echo "~~~~~"
    echo "Enter the current file name:"
    read current_name
    echo "Please enter the new file name:"
    read new_name

    # Check if the current file exists
    if [ ! -f "$current_name" ]; then
        echo "Ops! File '$current_name' not found!"
        return
    fi

    # Rename the file
    mv "$current_name" "$new_name"
    echo "Nice! File has been renamed successfully!"
}

# Function to handle creating directories
create_directories() {
    echo "Create Directories"
    echo "~~~~~"
    echo "Enter the directory name please:"
    read directory_name
    echo "Enter the parent directory path (Leave blank for current directory):"
    read parent_directory

    # Set the parent directory to current directory if not provided
    if [ -z "$parent_directory" ]; then
        parent_directory="./"
    fi

    # Create the directory

```

```
mkdir -p "$parent_directory/$directory_name"

echo "Bravo! Directory has been created successfully!"

}
```

Function to handle deleting directories

```
delete_directories() {

    echo "Delete Directories"

    echo "~~~~~"

    echo "Please enter the directory path:"

    read directory_path

    # Check if the directory exists

    if [ ! -d "$directory_path" ]; then

        echo "Ops! Directory '$directory_path' not found!"

        return

    fi

    # Confirm directory deletion

    echo "Are you sure you want to delete the directory '$directory_path'? (y/n)"

    read confirm

    if [ "$confirm" != "y" ] && [ "$confirm" != "Y" ]; then

        echo "OK, Directory deletion canceled!"

        return

    fi

    # Delete the directory

    rm -rf "$directory_path"

    echo "Directory has been deleted successfully!"

}
```

Function to handle deleting files

```
delete_files() {

    echo "Delete Files"

    echo "~~~~~"

    echo "Please enter the file name or path:"

    read file_path

    # Check if the file exists
```

```

if [ ! -f "$file_path" ]; then
    echo "Ops! File '$file_path' not found!"
    return
fi

# Confirm file deletion
echo "Are you sure you want to delete the file '$file_path'? (y/n)"
read confirm
if [ "$confirm" != "y" ] && [ "$confirm" != "Y" ]; then
    echo "OK, File deletion canceled."
    return
fi

# Delete the file
rm "$file_path"
echo "File has been deleted successfully!"
}

# Function to delete old files
delete_old_files() {
    echo "Delete Old Files"
    echo "~~~~~"

    echo "Enter the directory path:"
    read directory_path

    echo "Enter the number of days for file inactivity:"
    read inactive_days

    # Check if the directory exists
    if [ ! -d "$directory_path" ]; then
        echo "Ops! Directory '$directory_path' not found!"
        return
    fi

    # Confirm deletion of old files
    echo "Are you sure you want to delete files in '$directory_path' that haven't been accessed for $inactive_days days? (y/n)"
    read confirm

```

```

if [ "$confirm" != "y" ] && [ "$confirm" != "Y" ]; then
    echo "OK, File deletion canceled!"
    return
fi

# Find and delete old files
find "$directory_path" -type f -atime +$inactive_days -delete

echo "Old files has been deleted successfully!"
}

# Function to display help information
display_help() {
    echo "File and Directory Management Script"
    echo "=====
echo
echo "This script provides options for managing files and directories."
echo
echo "Menu Options:"
echo "1. Move Files: Move files from one directory to another."
echo "2. Rename Files: Rename files."
echo "3. Create Directories: Create new directories."
echo "4. Delete Directories: Delete existing directories."
echo "5. Delete Files: Delete specific files."
echo "6. Delete Old Files: Delete files that haven't been accessed for a specified period."
echo "7. Help: Display this help information."
echo "0. Exit: Quit the script."
echo
echo "Usage: ./file_directory_management.sh"
echo
echo "Usage Examples:"
echo "~~~~~"
echo "- Move Files:"
echo "  Enter the source file path and the destination directory path."
echo
echo "- Rename Files:"
echo "  Enter the current file name and the new file name."
echo
echo "- Create Directories:"

```

```
echo " Enter the directory name and the parent directory path (optional)."  
echo  
echo "- Delete Directories:"  
echo " Enter the directory path. You will be prompted to confirm the deletion."  
echo  
echo "- Delete Files:"  
echo " Enter the file name or path. You will be prompted to confirm the deletion."  
echo  
echo "- Delete Old Files:"  
echo " Enter the directory path and the number of days for file inactivity."  
echo " You will be prompted to confirm the deletion of old files."  
echo  
}
```

Main script execution

```
while true; do  
    display_menu  
    echo "Enter your choice (0-6):"  
    read choice  
  
    case $choice in  
        0) # Exit  
            echo "Good Bye!"  
            echo "Exiting..."  
            break  
            ;;  
        1) # Move Files  
            move_files  
            ;;  
        2) # Rename Files  
            rename_files  
            ;;  
        3) # Create Directories  
            create_directories  
            ;;  
        4) # Delete Directories  
            delete_directories
```

```
;;  
5) # Delete Files  
    delete_files  
;;  
6) # Delete Old Files  
    delete_old_files  
;;  
*) # Invalid choice  
    echo "Invalid Input. Please try again."  
;;  
esac  
  
echo  
read -n 1 -s -r -p "Press enter any key to continue..."  
done
```