


## CSE 232, Lab Exercise #2

### Partner

Remember we are working in pairs. Choose a partner and trade off the typing chores.

### Command Line: Copying, Moving and Deleting Files

**A Warning**  Deleting or overwriting important operating system files can ruin the operation of your computer. Be sure where you are (not in a system directory) when you do work like this. Check your prompt, it tells you where you are.

### cp

The `cp` command "copies" files from one path (location) to another. The `cp` is used with two arguments as follows: `cp source destination`

1. The source file: This is the file you want to copy.
2. The destination file: This is where you want your copy to be placed

You copy files that are in your working directory or by specifying a fully qualified path to the file. Examples:

```
cp a.txt b.txt
```

copies `a.txt` to file `b.txt`. `a.txt` is unchanged, `b.txt` is created (if it doesn't exist) or overwritten (if it does exist). Both files have the same contents after.

```
cp ../c.txt ../../c.txt
```

This copies `c.txt` (`..` means parent directory, so `c.txt` is in the parent to the current directory) to a file called `c.txt` (which is two directories up from current)

```
cp ~/Downloads/d.txt .
```

This copies the `d.txt` file from my `home_directory/Downloads` (`~` means your home directory) folder to the current directory (the `.` means current directory). If you specify **only** a directory as a destination, it will create a file with the same name in that directory.

The `cp` can also copy directories if you use the `-r` flag. The "r" stands for recursive, meaning to copy the directory and all sub-directories of it. Example:

```
cp -r other other2
```

copies the directory `other` and all its contents (files, directories, everything) into a directory called `other2`.

1. Save the file `lab02_folder.zip` from the Weekly lab web page to your Desktop. You may need to right click the link and select "Save Link As ..."

2. Right click the zipped file on the Desktop and select "Extract Here". You should now have a folder called "lab02\_folder" on your Desktop now.
3. In your terminal change directories to the Desktop (`cd ~/Desktop`).
4. `cd` to the folder named "lab02\_folder", then to the subfolder named "b\_folder", then to the folder named "start".
5. If you run `ls`, you should see a folder named "other" and a file named "f.txt".
6. Copy the file named "f.txt" to the file "f2.txt".
7. Copy the folder named "a\_folder" to the working directory. The "a\_folder" is in the same folder as "b\_folder".

★ Please show your TA the output from `ls`. There should be four items in the folder named "start":

- `f.txt`
- `f2.txt`
- `other/`
- `a_folder/`

### **mv**

The `mv` command is short for move and is identical to `cp`, except the original source doesn't exist after it is moved to the destination. Also, `mv` doesn't need the `-r` flag to move folders, it is happy to move folders just like files. Example:

```
mv other other2
```

This is basically a renaming of the directory. Run the above command, and confirm the the "other" folder no longer exists, and there's now a "other2" folder present. Be sure your working directory is the "start" folder.

The `mv` command is often used to rename directories and folders.

### **rm**

The `rm` command is short for remove (delete). **WARNING: files deleted with `rm` are effectively gone forever. They don't go to the TrashCan/RecycleBin. So please be careful with `rm`.**

The `rm` command takes one argument: the file or folder you wish to delete. Note: if you wish to delete a folder, you need to supply the `-r` flag. Example:

```
rm f.txt
```

Deletes the file `f.txt`

```
rm -r a_folder
```

Deletes the folder `a_folder` and everything within it. Without a backup it is effectively gone forever!

## Assignment Overview

The aim of this assignment is practice the use of while loops and conditionals statements. You are going to write a program that prompts the user for an integer and then determines the **additive persistence** and corresponding **additive root**. You will continue prompting the user for an integer until they enter a negative number which indicates they are finished

## Background

There are many properties of numbers that one can investigate. The ancient Greeks were fascinated with the properties of integers, even ascribing them mystical properties.

One such property is an integer's **additive persistence** and its resulting **additive root** (<http://mathworld.wolfram.com/AdditivePersistence.html>). Additive persistence is a property of the sum of the digits of an integer. The sum of the digits is found, and then the summation of digits is performed creating a new sum. This process repeats until a single integer digit is reached. Consider the following example:

1. The beginning integer is 1234
2. Sum its digits is  $1+2+3+4 = 10$
3. The integer is now 10
4. The sum of its digits is  $1 + 0 = 1$
5. The integer is 1. When the value reaches a single digit, we are finished. This final integer is the additive root

The number of cycles is the additive persistence. The integer 1234 has an additive persistence of 2 (first sum was 10, then the second sum was 1). The final digit reached is called the integer's additive root. The additive digital root of 1234 is 1.

## Program Specifications

The program should run as follows.

- 1) Program takes in a single long from input, the number being checked.
  - a. if the input long is 0 or less, print the single work "Error" and end the program.
- 2) Otherwise, the output should be two space separated longs
  - a. the persistence
  - b. the additive root
- 3) If the input long is a single digit, report its additive persistence as 0 and its additive root is the input number.
- 4) Otherwise calculate the additive persistence.



Please show your TA your working program

## Testing

Mimir has a directory Lab02:Persistence and an empty file lab02.cpp that you can use for testing.

## Getting Started

- 1) Create the lab02.cpp in your favorite editor or work on Mimir
- 2) Break the problem down into parts. Some obvious parts:
  - a. gather input from the file, and check for negative numbers (the quit condition)
  - b. check if the input is between 0 and 9.
    - i. that's a special case as indicated
  - c. otherwise write a loop that calculates the persistence
    - i. track the count through this loop
  - d. inside that write a loop that can sum the digits of an integer until it reaches a single digit.
- 3) How do you get the digits of an integer? Look at a combination of division (/) and remainder(%) operators on integers.
- 4) I would add some “diagnostic output” so you can be sure things are working as they should. For each pass through the loop of the additive persistence, print each new integer created. Feel free to always do this as you need to work on your ability to debug problems. You can always fix it to give the exact, required output later.

## Want to do more?

There is also a multiplicative persistence. Put that in your loop and calculate it for each number. No Mimir checks for this but you can do it anyway if you have time.

## The multiplicative persistence

(<http://mathworld.wolfram.com/MultiplicativePersistence.html>) and resulting **multiplicative root** are determined the same way, only multiplying the digits of an integer instead of adding. For example

1. The beginning integer is 1234
2. The product of  $1*2*3*4 = 24$
3. The integer is now 24
4. The product of  $2*4 = 8$
5. The integer is now 8. When the value reaches a single digit, we are finished. This final integer is the multiplicative root.

As before, the number of cycles is the multiplicative persistence and the final number the multiplicative root. For 1234, the multiplicative persistence is 2, and its multiplicative root is 8.

## Things to think about

- what type should the integers be (int, long)
  - why
- what is the largest long you can have

- look it up
- what happens when you enter a number that is too large? For example, the smallest number with an additive persistence of 4 is 19999999999999999999 . Can you make your calculation with that number?