# Physics 2G03 – Homework 1

## 1 - File Wildcard Expressions

1. ***ls b*3***
2. ***cp *.??? backup/***
3. ***rm -r [A-Z]****

## 2 - Options For Unix Commands

1. The flag `***-1***` will list all files in a single column (i.e. ***ls -1***)
2. The flags, `***-i***` & `***--ignore-case***` will tell 'grep' to ignore case (i.e. ***grep -i***)
3. The flag `***-r***` will tell 'grep' to search files in all subdirectories (i.e. ***grep -r***). You can also use the flag `-d` and specify the 'recurse' action to achieve the same effect (i.e. ***grep -d recurse***)
4. The flags `***-i***` and `***-I***` will make 'rm' prompt you before deleting a file (i.e. ***rm -i***)
5. `***tail -30 fileName.txt***` will show the last 30 lines of 'fileName.txt'
6. The flag `***-r***` will tell 'sort' to sort in reverse order (i.e. ***sort -r***)

## 3 - Shells & Commands

1. The default value of my `PATH` variable is:
   ***.:/home/chowdhaj/bin:/usr/local/openmpi-2.0.1/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/usr/local/abinit/bin***
2. The PATH variable is an environment variable on *nix operating systems that specifies a set of directories where executable programs (i.e. man, cd, ls, etc.) are located. This is where the shell goes to when it executes commands (i.e. man, cd, ls, etc).
3. If you typed `***xeyes***` at the command prompt, the shell (i.e. ***tcsh***) would run the `***xeyes***` program in your current directory, and NOT the standard `***xeyes***` program. This is because the shell checks the current director for executable programs before checking the `PATH` variable. So, if it finds an executable called `***xeyes***` in your current directory, then the shell will execute that one. If it cannot find `***xeyes***` in your current directory, then it will check the `PATH` variable. This is why you should never name your programs/scripts with identical names to other executables/programs.
4. The first word on a command prompt is an executable program. Hence, the shell checks all executable programs to attempt to complete the current word being typed. The shell first checks the current directory for a match before moving onto the `***PATH***` variable.
5. For the second word on a command line, the shell examines all files in the current directory, including executables. However, for a command like `cd` it examines the files in the directory you are pointing to. For example: ***cd ~/Desktop/<TAB>*** would list all files in the `Desktop` directory. Either way, the shell examines files in the directory you are in OR are pointing to.

## 4 - Shell Start-up Scripts

1. I would modify the '**.cshrc**' file, located in the home directory, and add the following command: ***alias rm 'rm -i'***
2. I would add the following line in the '.cshrc' file, located in the home directory:
   ***alias xterm 'xterm -cr red &'***
3. I would use the command `***set noclobber***` to prevent overwriting of existing files with a redirection. And `***unset clobber***` reverses this and allows me to overwrite files with redirections.
4. To make the prompt show the time, you need to use the '***%t***' specifier. For example:
   ***set prompt = '{%t}$ '***
5. The prompt I came up with looks like the following:
   ***{8:16pm Of Sun, Oct 04}(chowdhaj@phys-ugrad)[~/homework]***
   ***>>>***

It has basic 4 parts to it. The first part shows the time and date in curly braces (i.e. {8:16pm Of Sun, Oct 04}). The second part shows the username and hostname of the terminal in round brackets - who is logged in and to what machine (i.e. (chowdhaj@phys-ugrad)). The third part shows what directory you are currently in, in square brackets (i.e. [~/homework]). The fourth part is three greater than symbols on the next line, and these indicate where the user needs to start typing (i.e. >>>). You can achieve this with the following command:

***set prompt = '{%t Of %d, %w %D}(%n\@%m)[%~]\n>>> '***

## 5 - Pipes & Redirection

ps aux | grep "tcsh" | wc -l