CSI 3336 Systems Programming
Project 5

Bash History Processor

For this problem you are going to write a C program called **bhp**. This stands for bash history processor. The job of bhp is to read the contents of .bash_history and produce a summary. This will require you to use null-terminated character array strings, dynamic memory allocation and the C standard library for file I/O. You should be able to run your command as follows.

```
cat .bash_history | bhp
```

When you run like this, your command should produce output like the following.

```
command      freq
cat           300
ls           1234
grep           20
```

Your bash history process will read output from cat (which will show up in standard input) and will print a summary of how many times a specific command has been executed (in any order you like). Each line will display the command's name, left justified in a 12-column field. This will be followed by a space and the number of execution times right justified in a 4-column field.

Use instances of the following structure to keep up with information about each command. From this, you can see that each command is limited to 12 characters in length. Your main program will use a dynamically allocated array of CmdRec to keep up will all the commands it has seen. You will need to resize (grow) this array as you encounter new commands in the output of cat. Make sure to start off with a small amount of memory to verify that your dynamically resizing array is working properly.

```
/** struct used to keep a count of commands encountered in .bash_history **/
struct CmdRec {
/** Name of the command **/
char cmdName[ 13 ];

/** Frequency of commands encountered **/
int cmdCount;
};
```

Call your source file **bhp.c**. When you are done, turn in a printout and submit an electronic copy using submit.systems.