CSCI 4100
Assignment 3
Writing a `bash` Shell Script

Learning Outcomes

Write a `bash` shell script.

Required Reading

Linux - Chapters 8 and 10

Instructions

For this assignment you must write a bash shell script that keeps track of a list of email contacts by storing them in a text file called `contacts.dat`. The source file should be called `yourlastnameAssign3.sh`, except with your actual last name

Your program should present a menu to the user with the following options:

1. Add a contact
2. Remove a contact
3. Find a contact
4. List all contacts
5. Exit the program

The first option should prompt for a full name and check to see if there is already an entry containing this name in `contacts.dat`. If there is already an entry for this name it should display a message indicating that an entry already exists. If not, it should prompt for the email address, then add a line `contacts.dat` containing the name followed by a comma, followed by the email address.

The second option should prompt for a full name, then remove the any lines from `contacts.dat` containing this name. The third option should prompt for a full name, then display any lines from `contacts.dat` containing this name. The fourth option should display the contents of `contacts.dat` sorted by last name. The fifth option should exit the script

If the user chooses any of the first four options, once the action has been performed the menu should be displayed again and the user should be prompted to make another choice.

Helpful Commands and Options

Your shell script will need the file called `contacts.dat` to exist in order to work. If you execute the following command at the beginning of your script:

```
touch contacts.dat
```

This will create an empty text file the first time the script runs, and verify that it exists when you run the script again.

The `read` command prompts for input and stores it in a variable. You can use the `-p` option to provide the user with a prompt:

```
read -p "Enter your name: " name
```

This will store whatever the user enters in the variable `name`.

The `grep` command will display the lines in a file that match a given pattern. You can suppress the output using the `-q` option if all you want is to determine if such a line exists:

```
if grep -q pattern file
then
  # do stuff
else
  # do other stuff
fi
```

You can display all of the lines that do not match a pattern using the `-v` option:

```
grep -v pattern oldfile > newfile
```

This is one way to remove lines matching a pattern from a file.

The `sort` command sorts a file alphanumerically by line. If you want to sort by a specific field of a file containing a sequence of records, you can use the `-k` option:

```
sort -k 1 file # sorts by first field
sort -k 2 file # sorts by second field
```

Testing and Debugging Shell Scripts

The easiest way to debug a shell script is to test each command individually using the shell. If you want to know the exit status of a command, you can display the special variable `$?`, which evaluates to the exit status of the most recently executed command:

```
$ ls nosuchfile.txt
ls: nosuchfile.txt: No such file or directory
$ echo $?
1
```

If more substantial debugging is required you can run `bash` directly using the `-x` option:

```
$ bash -x hello.sh
+ echo 'Hello World!'
Hello World!
```

All executed commands will appear in a line with a `+` at the beginning before they are executed.

Sample Run

Your output should look similar to the following:

```
Select one of the following options:
-----------------------------------
1. Add a contact
2. Remove a contact
3. Find a contact
4. List all contacts
5. Exit the program

Enter a choice (1-5): 1
Enter a name: Nicholas Coleman
Enter an email address: colemann@apsu.edu

Select one of the following options:
-----------------------------------
1. Add a contact
2. Remove a contact
3. Find a contact
4. List all contacts
5. Exit the program

Enter a choice (1-5): 1
Enter a name: John Nicholson
Enter an email address: nicholsonj@apsu.edu

Select one of the following options:
-----------------------------------
1. Add a contact
2. Remove a contact
3. Find a contact
4. List all contacts
5. Exit the program

Enter a choice (1-5): 4
Nicholas Coleman, colemann@apsu.edu
John Nicholson, nicholsonj@apsu.edu

Select one of the following options:
-----------------------------------
1. Add a contact
2. Remove a contact
3. Find a contact
4. List all contacts
5. Exit the program

Enter a choice (1-5): 1
Enter a name: Nicholas Coleman
There is already an entry for Nicholas Coleman
```

```
Select one of the following options:
-----------------------------------
1. Add a contact
2. Remove a contact
3. Find a contact
4. List all contacts
5. Exit the program

Enter a choice (1-5): 3
Enter a name: Nicholas Coleman
Nicholas Coleman, colemann@apsu.edu

Select one of the following options:
-----------------------------------
1. Add a contact
2. Remove a contact
3. Find a contact
4. List all contacts
5. Exit the program

Enter a choice (1-5): 2
Enter a name: Nicholas Coleman

Select one of the following options:
-----------------------------------
1. Add a contact
2. Remove a contact
3. Find a contact
4. List all contacts
5. Exit the program

Enter a choice (1-5): 4
John Nicholson, nicholsonj@apsu.edu

Select one of the following options:
-----------------------------------
1. Add a contact
2. Remove a contact
3. Find a contact
4. List all contacts
5. Exit the program

Enter a choice (1-5): 5
Thanks for using this program!
```

What to Hand In

Download the source file `yourlastnameAssign3.sh` to your local machine, then upload it to D2L in the dropbox called Assignment 3.