# CSE 4251: Lab 3

This Lab requires writing Bash scripts interacting with a user. Please do this Lab independently.   
The Bash manual can be helpful <https://www.gnu.org/software/bash/manual/bash.html>

# Exercise 1 Description

For this exercise we will create a shell “contacts” program which is used for recording contact information.

Write a Bash script titled ***createcontacts.sh*** that will create and/or modify a text file called ***mycontacts.txt*** in the current directory, where each line of ***mycontacts.txt*** consists of four fields separated by comma. The four fields are full name, birthday (dd-mm), phone number (zzz-yyy-yyyy), and email (name@domain.com). The script will do the following:

1. After running the script, it will show the user a menu with at three options and wait for the user to choose one of them:  
   **1. Enter a new contact**  
   **2. Show contacts**  
   **3. Exit**
2. If the user chooses “Enter a new contact”, the script should prompt instruction information and ask the user for the data for each field as follows:
   * full name
   * birthday (dd-mm)
   * phone number (zzz-yyy-yyyy)
   * email (name@domain.com)

For each field, the script should first prompt the field name and then get the input of the user. Assume that user will never type any content with invalid format or any comma in this step b. After the user enters all four fields. Save the record to the ***mycontacts.txt*** and print out that a new contact is added.

1. If the user chooses “Show contacts”, the script should show all contents based on the ***mycontacts.txt*** file. The output should be formatted to be read easily and contain a header describing all four fields, please refer to the example shown later.
2. After step b and c, the script should go back to step a and show the menu again and wait for the user’s choice.
3. If the user chooses “Exit”, the script exits. If the user chooses an invalid option in the menu of step a, print out that the option is invalid and let user re-choose a valid option. Repeat showing options and waiting the user’s input as in step a until the user chooses a valid option. The script will then operate based on the user’s choice.
4. Make the script file executable and execute it to test each one of the options in the menu of step a.

The script should be able to be executed multiple times. If the ***mycontacts.txt*** does not exist in the current directory when the script is executed for the first time, the script should create a new ***mycontacts.txt*** file. If a valid ***mycontacts.tx***t file already exists in the current directory, the script should append content to the existing ***mycontacts.txt*** file no matter if the script is executed for the first time or not. The script should not accidentally delete the contact information in an existing ***mycontacts.txt*** file in the current directory.  
  
An example of the interaction between the script and the user can be as follows. Here the blue words are typed by a user. Assume the script is executed for the first time and ***mycontacts.txt*** does not exist before this execution:  
  
**$ ./createcontacts.sh**  
**1. Enter a new contact**  
**2. Show contacts**  
**3. Exit**  
**Please choose one of the options: 4**  
**The input option is invalid. Please enter a valid option.**  
**1. Enter a new contact**  
**2. Show contacts**  
**3. Exit**  
**Please choose one of the option: 1**  
**$ Please enter the full name: John Smith**  
**Please enter the birth date (dd-mm): 02-10**

**Please enter the phone number (zzz-yyy-yyyy): 123-123-12345**  
**Please enter the email (name@domain.com): example@abc.com**  
**Successfully add a new contact!**

**1. Enter a new contact**

**2. Show contacts**

**3. Exit**

**Please choose one of the option: 2**  
**Name Birthday Phone Email**

**John Smith 02-10 123-123-12345 example@abc.com**  
**1. Enter a new contact**

**2. Show contacts**

**3. Exit**

**Please choose one of the option: 3**  
**$**  
  
After entering one new contact as shown in the example, the ***mycontacts.txt*** should look like the following:  
  
**John Smith,02-10,123-123-12345,example@abc.com**

The ***mycontacts.txt*** should not have the header and each line is just four fields separated by comma. Each line represents the information of a contact. Keep in mind that the prompt of your ***createcontacts.sh*** script and how the script prints the output does not need to be the same as the example. But the format of the ***mycontacts.tx***t should be the same as the example.

# Exercise 2 Description

For this exercise write a Bash script titled ***searchcontacts.sh*** that manipulates the text file named ***mycontacts.txt*** in the current directory. The ***mycontacts.txt*** uses the same format as defined in Exercise 1. The bash script ***searchcontacts.sh*** does the following:

1. Show the user the day of the week, and the current time in the beginning.
2. Show the full names of all contacts in the ***mycontacts.txt*** whose birthdays are today (which is given by the current system time). If no contacts’ birthdays are today, prints out “no contact’s birthday is today”.
3. Ask the user to enter a full name to be searched in the ***mycontacts.txt***.
4. Search in the file ***mycontacts.txt*** for that name and print out the contact whose name is the same as the full name given by the user. If there are multiple contacts with the same name as the name given by the user, print out all of them. Exit the script after printing out the content.
5. If the ***mycontacts.txt*** does not contain any contact whose name is the same as the name given by the user, prompt out that “no matched contact found” and exit the script.

An example of the interaction between the script and the user can be as follows. Assume today is Oct 2 and the ***mycontacts.txt*** in the current directory is created in the example of Exercise 1. Blue words are typed by a user:  
  
**$ ./searchcontacts.sh**   
**Today is Sunday and the current time is 22:00:10.**  
**The following contacts’ birthdays are today**  
**John Smith**  
**Please type the name to be searched: John Smith**  
**Name Birthday Phone Email**

**John Smith 02-10 123-123-12345 example@abc.com**  
**$**

Similar to Exercise 1, the print output of your ***searchcontacts.sh*** does not need to be exactly the same as the example. But the script should assume that the format of the ***mycontacts.txt*** strictly follows the format defined in Exercise 1.

# Submission Instructions

To submit, create a single zip filethat contains the ***createcontacts.sh*** and ***searchcontacts.sh.***