

CSE 333 - OPERATING SYSTEMS

Programming Assignment # 1

DUE DATE: 31/10/2016 - 23:00 (No extension)

1. (15 pts) Write a shell script that asks a user for a password before allowing them to continue. Set up a password to be “cse333” in the code then get and verify the password from the user. Display “Access granted” if the input is “cse333”. Display “ACCESS DENIED” if not. A user has a maximum number of 3 wrong trials. If it passes this value, your program should exit with a message.

Ex:

```
$ ./myprog1.sh
Please enter the password: wrongPassword
ACCESS DENIED!
Please enter the password: wrongPassword2
ACCESS DENIED!
Please enter the password: wrongPassword3
ACCESS DENIED!
You have reached the maximum number of trials.
```

Ex:

```
$ ./myprog1.sh
Please enter the password: cse333
Access granted
```

2. (20 pts) Write a shell script that prints the name of the oldest file (or directory) in the current directory. Your program should report if it is a directory, regular file or symbolic link and also report the group permissions (read, write and execute) on the file or directory. Please note that you should consider all possible error checking and related warnings.

Ex:

```
$ ./myprog2.sh
The oldest file is File1 which is modified on 01/09/2016 at 02.53.
File1 is a file and has read and write permissions for the group!

$ ./myprog2.sh
The oldest directory is Dir1 which is modified on 11/08/2016 at 19.20.
Dir1 is a directory and has read and write permissions for the group!
```

3. (15 pts) Write a shell script that reads integers (one-per-line) from the user until it reads the word “**finish**”. Afterward, it will report the *sum* and the *average* of numbers given by the user. Please note that you should consider all possible error checking and related warnings.

Ex:

```
$ ./myprog3.sh
Enter a sequence of numbers followed by "finish"
7
18
42
123
54
```

```
92
finish
Sum : 336
Average : 56
```

4. (20 pts) Write a shell script which takes a single command-line argument, a number between 5 and 79 (inclusive). If the given number is out of this range, your program should print out an error message and then exit.

If your program is given legal input, it should print a square shape like the following (which would be the output with an argument of 7). The printed square is always outlined with two rows/columns of asterisks; the command-line argument determines the height of the square.

Ex: `$./myprog4.sh 7`

```
*****
*****
**    **
**    **
**    **
*****
*****
```

5. (20 pts) Suppose that you want to write the same letter to many people except that you want each letter addressed to the senders personally. This mail merge facility can be created using a shell script. In this question, your task is to write a letter template that has the word NAME where a name should go. Your program should ask the user for the name for the letter.

Ex:

Your template file might hold: (letter.txt)

```
Hi NAME
I want to tell you about a new mobile application we are offering.
It will make run a little robot that will make you coffee in the
morning and deliver it to you in bed.
Please buy this.
Regards,
Professor Moriarty
```

You run your script:

```
./myprog5.sh
```

Enter the name of the recipient: Sherlock Holmes

Expected result:

```
Hi Sherlock Holmes
I want to tell you about a new mobile application we are offering.
It will make run a little robot that will make you coffee in the
morning and deliver it to you in bed.
Please buy this.
Regards,
Professor Moriarty
```

- **Bonus:** You will get 10% extra credit if your program support a *Menu* including all questions above. Example:

```
./myprog.sh
```

```
Please select an option:
```

1. Password Check
2. Find the Oldest File
3. Find sum and average
4. Square of asterisks
5. Rename the Letter
6. Exit

The user should not select other options without Password Check (Question 1). If the user provides the option 3 without password check, you will give an appropriate warning message to select option 1 firstly. These options must be printed inside a loop until “Exit” option is selected.

Notes:

- You are required to consider all necessary error checking for the programs.
- No late homework will be accepted.
- In case of any form of **copying and cheating** on solutions, all parties will get **ZERO** grade. You should submit your own work. In case of any forms of cheating or copying, both giver and receiver are equally culpable and suffer equal penalties.
- You have to work with a partner. Your partner will not be changed throughout the semester.
- You are required to submit a minimum 2-pages report and commented code (via e-mail: cse333.projects@gmail.com).
- Your report should include explanations about implementations. Your implementation detail should be provided in the source code comment.
- Please put your COMMENTED source codes and project report in a zip file and make sure that your zip file contains your names and student IDs!