

CSE 333 - OPERATING SYSTEMS

Programming Assignment # 1

DUE DATE: 15/11/2021 - 23:59

1. (20 pts) Write a shell script that takes a single command line argument which is a file containing one integer per line as the following:

```
20
18
5
32
4
```

From this data, your program should print out a row of stars of the given length for each integer. For the input above, your program should print the following:

Ex:

```
$ ./myprog1.sh filename
```

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

2. (20 pts) Write a shell script that takes directory name as an argument, and it deletes every file in the directory that is **not**:
 - i. a c file (i.e., a file matching *.c) or
 - ii. a header file (i.e., a file matching *.h) or
 - iii. a file named "Makefile" or "makefile"

This command should not attempt to remove subdirectories of the given directory.

Ex:

```
$ ls dir1
Makefile bar foo junk myprog.c mydir
$ ./myprog2.sh dir1
$ ls dir1
Makefile myprog.c mydir
```

3. (25 pts) Write a shell script that takes a filename and two words as arguments. Then it searches for the first word in that file and substitute all occurrences of it with the second word and print how many substitutions was made (In numbers (i.e., 3), **not** in text(i.e., three)). For example, if the program is executed with a file including the word "apple" 3 times and the second word is "orange", then it should change all 3 "apple" to "orange". The program has to be case sensitive.

Ex:

```
$ ./myprog3.sh file apple orange
All 3 occurrences of "apple" in "file" has changed with "orange"
```

4. (20 pts) Write a shell script to organize a directory by first creating two sub directories named smallest and largest. Then the script should move the largest file in the current directory to largest directory and the smallest file to smallest directory.

Ex:

```

$ ls -l
-rw----- 1 std std 152144 Jun 20 2005 alice-in-wonderland.txt
-rw----- 1 std std 82140 Jun 20 2005 barleby-scrivener.txt
-rw----- 1 std std 13421 Jun 20 2005 calaveras-county.txt
-rw----- 1 std std 13107 Jun 20 2005 cask-of-amontillado.txt
-rw----- 1 std std 635 Jun 20 2005 french.txt
-rw----- 1 std std 496769 Jun 20 2005 hawthorne.txt
-rw----- 1 std std 172541 Jun 20 2005 looking-glass.txt
drwx----- 14 std std 476 May 25 2007 shakespeare
-rw----- 1 std std 192710 Jun 20 2005 song-of-hiawatha.txt
-rw----- 1 std std 35238 Jun 20 2005 trees-and-other-poems.txt
$ ./myprogr4.sh
$ ls -l
-rw----- 1 std std 152144 Jun 20 2005 alice-in-wonderland.txt
-rw----- 1 std std 82140 Jun 20 2005 barleby-scrivener.txt
-rw----- 1 std std 13421 Jun 20 2005 calaveras-county.txt
-rw----- 1 std std 13107 Jun 20 2005 cask-of-amontillado.txt
drwx----- 2 std std 496769 Oct 25 2021 largest
-rw----- 1 std std 172541 Jun 20 2005 looking-glass.txt
drwx----- 14 std std 476 May 25 2007 shakespeare
drwx----- 2 std std 635 Oct 25 2021 smallest
-rw----- 1 std std 192710 Jun 20 2005 song-of-hiawatha.txt
-rw----- 1 std std 35238 Jun 20 2005 trees-and-other-poems.txt
$ ls -l largest
-rw----- 1 std std 496769 Jun 20 2005 hawthorne.txt
$ ls -l smallest
-rw----- 1 std std 635 Jun 20 2005 french.txt

```

5. (15 pts) Write a shell script that accepts a number as an argument. Then it should find the sum of numbers formed by exchanging consecutive digits. For input 5614, the output should be 65+16+41=122.

```

Ex:
$ ./myprog5.sh 5614
122

```

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 or two. Your partner(s) will not be changed throughout the semester.
- You are required to submit commented code (via e-mail: cse333.projects@gmail.com).
- Please put your COMMENTED source codes and project report in a zip file and make sure that your zip file name contains your student IDs!
Ex: 150713852_150713853_Project1.zip

IMPORTANT

- Your project will be graded with a auto grader. A small example shell script that executes programs is given to you with this project. Your programs should be able to executed with this script. Here are the rules you should follow for the auto grader to correctly grade your programs:
- You have to write your programs in separate shell files. The names of your shell programs should follow the examples above. (myprog1.sh, myprog2.sh, etc.)
- You should not put any spaces for the name of your programs and zip file.
- You should not use Turkish characters anywhere. (In the programs or as filenames.)
- When extracted, make sure that your 5 programs are directly under the extracted directory (the extracted directory should not contain any subdirectory).
- The arguments to the programs should be given exactly as in the examples above.
- Your programs should do all changes in their current working directory only. Do not make any changes in any other places!
- Your programs should make only the requested changes and nothing more.
- If your program does not follow these rules, then you will not get a grade for violating questions.