

Scripting and Computer Environments - CSE 505
IIIT Hyderabad - Monsoon 2016
ASSIGNMENT - 2

ASSIGNMENT DATE : 6 Oct 2016
SUBMISSION DATE : 14 Oct 2016, 9PM

SUBMISSION FORMAT :

1. Create a folder with your roll number <rollno>_assignment2
2. Under the folder create <section_no>_<question_no>_<subpart>.sh for each question
3. Compress the file as <rollno>_assignment2.tar.gz and then upload it to your moodle account.

How to get zero in the Assignment?

1. DEVIATING FROM THE UPLOAD FORMAT will get you a straight zero.
 2. ANY KIND OF PLAGIARISM will also lead to a straight zero.
-

SECTION -1 Solve using single Unix Commands ONLY.

Q 1.

Consider the file **q1.txt**

- a) Print all the employees whose ids are between 350 and 750
- b) You will have many departments in that given file, print the number of people in each department.

Eg: Sample o/p: Technology => 2

Bleh => 3

Q 2.

You are given a file called **sort.lst**, the record format in that is

name|empID|designation|department|dob|salary

Sort the records based on the employeeID. And print the whole record.

Meaning: The output should contain the sorted order of records, the sorting criteria being empID.

Go through the **sort.lst** records to understand this question better/

Q 3.

Given **a number as command line input** , your task generate n prime numbers (starting from 2) using awk.

Q 4.

You are given a file called “**numbers.txt**” in this file the following format is followed.

Format (all space separated) :

number1 word1 word2 word3 Wordn

Number2 word1 word2 ... wordk

.
. .
.

numberX word1 word2.....wordm

i.e, each line will mandatorily contain the first field as a number, and then variable number of fields of words.

Now, your job is to print '*' in the beginning, the number of stars should be equal to the number which is present in the first field, this followed by, the fields of the row, where the first field and last fields are swapped.

A sample i/p and output is given, go through it carefully:

Eg: if input is => **3 shweta is from pune**

Output should be => ***** pune is from shweta**

Q 5.

Sunil has recently joined the SCE TAs team. As he is a fresher he was given the basic task of creating a file containing the list of all the students and assign the students as evenly as possible to all the 7 TAs for the purpose of evaluation. As Sunil is too lazy to do this work, he plans to write a script/command to perform this task. You being the Scripting guru, decided to help him in writing the script/command. As you have decided to help him. You can find the list of all the assignment submissions in SubmittedFiles.txt. (Do it using only **awk**) **Note:** For each instance of execution, the generated output should vary.

Note: For each instance of execution, the generated output should vary.

Sample Output:

```
1. 201350839 - Prabhakar
2. 201350914 - Sheshadri
3. 201532656 - Viplav
4. 201550848 - Akanksha
5. 201560516 - Deeksha
6. 201560522 - Rambabu
7. 201560543 - Harika
-----ROUND: 1
8. 201560557 - Prabhakar
9. 201560558 - Sheshadri
10. 201560562 - Viplav
11. 201560573 - Akanksha
12. 201560592 - Deeksha
13. 201560597 - Rambabu
14. 201560609 - Harika
-----ROUND: 2
15. 201560617 - Prabhakar
16. 201560619 - Sheshadri
-----ROUND: 3
```

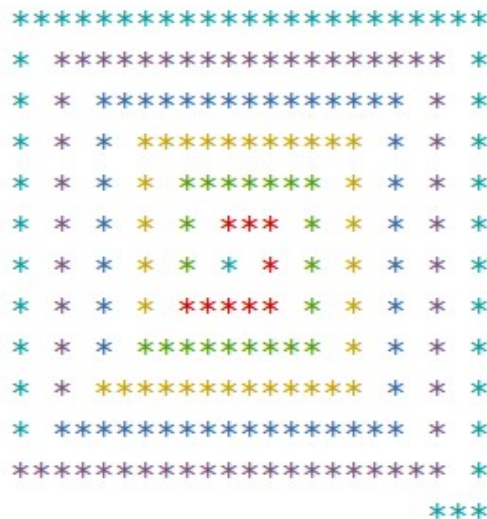
SECTION -2 YOU CAN ATTEMPT ANY 6 OUT OF 7 (if all attempted, best 6 will be considered)

Q 1.

Utsav wants to design a new screensaver using bash.

He builds a spiral starting from middle of the screen growing upto limit of your terminal and back again to the centre.

Limit of terminal is defined by $\min(\text{rows}, \text{columns})$, where rows is rows of terminal and columns is columns in terminal. For demo, you can view spiral.gif in resources.



Hint : use **tput** command

Catch: Your script should work for all terminal sizes. even if we minimise or maximize terminal size in runtime, screen saver must keep running with new configurations.

Q 2.

As the moodle was down during the peak hours of submission, Akanksha has received many assignment submissions through mail. Few of the mails were sent before the deadline while few of them arrived after the deadline. Now, Akanksha needs your help to perform the following tasks

1. **7_a.sh:** Check if the copy of assignment that she received is indeed completed before the deadline or not. Take deadline as command line argument(Hint: use the timestamp of the files present inside the tar.gz). If any of the .sh or the script file is edited after the deadline, print *Reject* else print *Accept*. If there are no non-empty .sh files or no .script file, print *Reject*.

Command Format: 7_a.sh "Deadline as YYYY-MM-dd HH:mm"

Sample command: 7_a.sh "2016-10-03 00:00"

2. **7_b.sh:** Check If the mailed copy of assignment is adhering to the pre-announced upload-format or not. If the upload format is followed then print *Accept* else print the name of the .sh file for which the naming-format has been violated. If there are no .sh files or .script file print *Reject*.

```
Rollno_assign2.tar.gz/  
|-- 1_1_a.sh  
|-- 1_2_a.sh  
|-- 1_2_b.sh  
|-- 1_3_a.sh  
|-- 1_3_b.sh  
|-- 1_3_c.sh  
|-- 1_4_a.sh  
|-- 2_1_a.sh  
|-- 2_2_a.sh  
|-- 2_2_b.sh  
|-- 2_2_c.sh  
|-- 2_3_a.sh  
|-- commads.lst  
|-- Rollno_assign2.script  
  
0 directories, 14 files
```

Valid upload format:

1. The archive should be named in the format *Rollno_assign2.tar.gz*
2. Each file present in the root of *Rollno_assign2.tar.gz* should be named in the following format *SectionNo_QuestionNo_SubQuestionNo.sh*

SectionNo → Integer

QuestionNo → Integer

SubQuestionNo → lower-case alphabet

e.g., 1_1_a.sh

In the .tar,gz has any file without the extension of .sh or .script, ignore them.

In the image, you should not validate the fileName-format for the files *commads.lst* and *Rollno_assign2.script*.

Q 3.

Write a script that recursively lists all files i.e is equivalent to 'ls -R'.

Requirements:

***You must use the recursive function to traverse**

*Can take any number of commandline arguments. For each of them, call this script

*If no command line argument is provided, work on current directory (just like ls!).

-Usage/Presentation Requirements:

* Name the script 3.sh * Usage: bash 3.sh <any number of files/directories>

- * Support any number of command line arguments, including zero.
- * Validate incoming inputs. Do not stop on Errors.

-Errors to handle and message Format :

- * 'Error: Invalid File/Directory!' - If a non-existent file/directory is passed
- * 'Warning: No permission' - If a directory has no permission, skip this and continue.

- Outputs to handle and message Format :

- * Output will be a directory along with the files in it. If you encounter a directory in here, call your script to traverse this directory. Here is the format to be followed:

<PathName of Directory>:

<file1>

<file2>

Q 4.

Write a recursive copy shell script to copy source folder to destination folder without using 'cp -r'. (Hint: Use export and readlink to store absolute path of the script). Input is given in format

4.sh <source> <destination>

where source is a folder name and destination is a folder name.

Requirements:

- **You cannot use recursive functions or cp -r command.**
- If destination folder does not exist, create one.
- Call the same script to traverse down the directory.
- Use **cp** command to copy single file, **mkdir** command to create directory with same name.
- **Directory names with spaces must be considered.**

Errors to handle and message Format:

1. 'Error: Invalid number of arguments!' - If number of arguments is != 2
2. 'Error: Invalid input!' – If source is not directory.
3. 'Error: Access denied!' - if source directory does not have access permission.
4. 'Warning: Dir <dir> has no permission. Skipping' - If any of the subdirectory doesn't have execute permission. (special case to handle: Think of how to copy linked files.)

Q 5.

After releasing the SCE assignment-2, Viplav(TA) went to his native place for spending the Dashara vacation with his family. Viplav got to know that every now and then there are queries popping up on the moodle discussion thread. For answering the queries from outside the campus, Viplav has to configure the VPN, start the VPN and then login to the moodle for answering the queries. Now Viplav realised that whenever he visits home, he'll have to repeat the same process of configuring the VPN and start it in order to access the moodle. So, he thought he could get some help from you in writing a bash script(start_vpn.sh) that could download the certificates, extract them to the relevant directories and then start the VPN. There should be another bash script file(stop_vpn.sh) for stopping the VPN. Help Viplav by providing the bash script that could serve his need.

Requirements:

The bash script, start_vpn.sh should download the certificates from the <https://vpn.iiit.ac.in/> and place the certificates in the appropriate directories and then start the VPN. Once the VPN is started, the script should print the message “VPN Started” on to the terminal. Later when stop_vpn.sh is executed the VPN should be stopped.

Q 6.

An online evaluation engine was developed by students for evaluating C/C++ codes. The students are expected to code the solution in C/C++ and submit their code file to the evaluation engine. Each student can make multiple submissions, until their code gets accepted completely by the engine. A folder named **Submissions** is created in the file-system of the judge, for storing all the codes submitted by the students. Whenever the student submits a code for any problem to the online judge(Eg : Mooshak, HackerRank etc), it will create a folder for that code in the below format,

XXXXXXXX_PY_ROLLNUMBER_BRANCH_NAME

Here **XXXXXXXX** is the 8 digit unique id (**greater id for the latest submission**) and “**Y**” in **PY** defines the problem number depending on that lab / assignment / lab test.

The **Submissions.tar.gz**, has the following folder structure

Main Folder : Submissions

Sub Folders : XXXXXXXX_PY_ROLLNUMBER_BRANCH_NAME

Finally Folder structure should be like :

```
Latest Files/
|-- P1
|   |-- 00003668_P1_20162083_CSE_NEETHU
|   |   |-- 1.html
|   |   |-- a1.c
|   |-- 00010243_P1_201550873_PGSSP_SIVARANJANI
|   |   |-- 1.html
|   |   |-- l2_p1.c
|-- P2
|   |-- 00004416_P2_201550873_PGSSP_SIVARANJANI
|   |   |-- 1.html
|   |   |-- l2_p2.c
|   |-- 00006918_P2_20162083_CSE_NEETHU
|   |   |-- 1.html
|   |   |-- b.c
|-- P3
|   |-- 00003627_P3_201550873_PGSSP_SIVARANJANI
|   |   |-- 1.html
|   |   |-- l2_p2.c
|   |-- 00020202_P3_20162083_CSE_NEETHU
|   |   |-- 1.html
|   |   |-- c2.c
```

NOTE: Few subfolders may not be in proper naming format, you have to ignore that.

Input files: <https://drive.google.com/drive/folders/0By4W33gcbxlfRnV3bllhNkxfMkE?usp=sharing>

We may run your program with different dump which has varying number of questions, and it should give proper output.

Q 7.

The online judge(OJ) mentioned in the previous question is used by the students for getting their code evaluated. This is how the OJ grades the students,

Code submitted IN lab \Rightarrow 1 point for each correct code

Code submitted AFTER lab \Rightarrow 0.5 points for each correct code

Statistics of code submitted by a student BEFORE leaving the Lab:

5	20162032_CSE_SANDIP	STUDENTS	20162032 CSE SANDIP	2:42:21 (1)	----- (1)	1:33:36 (0)	----- (3)	2:4:35:57
---	---------------------	----------	---------------------	-------------	-----------	-------------	-----------	-----------

Statistics of code submitted by a student AFTER leaving the Lab:

1	20162032_CSE_SANDIP	STUDENTS	20162032 CSE SANDIP	2:42:21 (1)	6:53:28 (1)	1:33:36 (0)	7:28:54 (5)	4:20:58:19
---	---------------------	----------	---------------------	-------------	-------------	-------------	-------------	------------

In the above mentioned statistics, the 2nd last column denotes the number of questions that has been submitted successfully by that student so for.

Now, your job is to write a code to calculate the total points obtained by each student.

E.g., In the above statistics, the total number of points secured by the student is given as $(2 \times 1) + ((4-2) * 0.5)$