CS251: Final Exam

April 26, 2017, 9:15am

Total Marks: 275

The Game

The Devas (gods) are playing a game to compete among themselves to see whose value is the greatest. There are 33 Devas (note that the word "koti" in the often quoted "33 koti" gods and goddesses do not stand for crore, but for *type*) listed in the file `devas.txt`. According to Indian science, there are 27 Nakshatras (listed in the file `nakshatras.txt`) and 12 Rashis (zodiac signs listed in the file `rashis.txt`). The Rashis divide the Nakshatras equally among themselves. Each Nakshatra has a weight. Each Deva plays a 12-faced die twice and chooses two Rashis. The corresponding weights of the Nakshatras add up to constitute the value of the Deva. The Devas are ultimately plotted in a descending order of their values.

General Instructions

There are *eleven (11)* phases in the exam which together complete the work-flow for the above game. The *twelfth (12th)* phase (a bash script) runs them all together. Each phase produces an output and may require an input which is produced by one or more of the previous phases. *No* manual editing of any files is allowed. For any of the phases, you can create and use temporary files.

The file `flow.pdf` summarizes the entire workflow.

If you cannot complete a phase, you can ask the TA for a solution to that phase. However, your output from the previous phases must comply strictly to the format required by this phase; otherwise, the solution provided by the TA will not work. You will, of course, lose all the marks for that phase.

You can, however, return to it later to salvage up to 50% of the marks for that phase.

You are allowed to use the Internet for only two purposes: *canvas* and searching. If, at any time, you are found using it for any other purpose (including ssh and scp), it will be treated as a case of copying, and will be dealt accordingly.

A *bash* script can contain any of the standard `bash` commands including *grep*, *sed* and *awk*.

Submission Instructions

Create a folder with your **roll number**. *Ensure* that everything that you work on is *inside* this folder.

Zip the folder named after your roll number as `rollnumber.zip`. Ensure that the zipping is done correctly, i.e., it contains all the files, sub-folders, etc.

Upload this zip file to canvas **in front of** a TA **only after** the scheduled time ends. Remember that *only* the contents of this zip file will be graded.

Phase 1: Bash (15)

Write a **bash** script, `phase1.sh`, that repeats each line 4 times of the file `nakshatras.txt` to output `phase1.txt`. For example, the first two lines of output should be

```
ashwini,7
ashwini,7
```

Phase 2: Bash (15)

Write a **bash** script, `phase2.sh`, that splits `phase1.txt` into 12 files each containing 9 lines. The files must have the prefix `phase2-` and should have a numeric suffix starting from `00`. Thus, the 12 files produced will be `phase2-00`, ..., `phase2-11`. For example, the first two lines of output of *second* file should be

```
krittika,6
krittika,6
```

Phase 3: Bash (40)

Write a **bash** script, `phase3.sh`, that reads `rashis.txt` and the 12 files produced in phase 2 and for each file produces the following. For each rashi, it reads the corresponding phase2- output file, and writes how many times each nakshatra is there in each rashi along with its count and weight. The outputs are collected in `phase3.txt`. For example, the first line of output should be

```
mesha :  4 ashwini 7 4 bharani 20 1 krittika 6
```

Phase 4: R (40)

Write an **R** script, `phase4.r`, that reads `phase3.txt` and outputs the combined sum of weights for each rashi in `phase4.txt`. For example, the first line of output should be

```
mesha 114
```

The weight of *mesha* is $4 \times 7 + 4 \times 20 + 1 \times 6 = 114$.

You may need to run the script using the `Rscript` command.

Phase 5: Octave (15)

Write an **octave** script, `phase5.m`, to output two random rashi numbers for each deva. The output file `phase5.txt` contains *only* the two rashi numbers. For example, the first line of output may be

```
3 8
```

This phase does not read any input.

Phase 6: Bash (20)

Write a **bash** script, `phase6.sh`, that outputs *only* the names of the Devas from `devas.txt` and not their supertypes. The output should be sorted by the names and the names should start from the first character in each line. For example, the first line of output should be

```
aghora
```

Phase 7: Bash (30)

Write a **bash** script, `phase7.sh`, that does the two following jobs.

The first job reads `phase4.txt` and outputs it in a *csv* format with a sequence number prepended. The output file is `phase7-1.txt`. For example, the first line of output should be

```
1,mesha,114
```

The second job reads `phase6.txt` and `phase5.txt` and combines them in a line by line manner to the file `phase7-2.txt`. For example, the first line of output should be

```
aghora,3,8
```

Phase 8: Sqlite3 (40)

Write a **sqlite3** script, `phase8.sql`, that creates a database `phase8.db`. (If the database file exists, delete it first. The deletion can be done in the overall bash file and not here.) As part of the database, create two tables. The first table, `rashis`, contains the fields id, name and weight and reads from `phase7-1.txt`. The second table, `devas`, contains the fields name, first rashi number and second rashi number and reads from `phase7-2.txt`.

Write a query that does the following. For each deva, it reads the corresponding two rashi numbers and outputs their weights. For example, the first line of output should be

```
aghora,134,160
```

The output is collected in `phase8.txt`.

Phase 9: Python3 (15)

Write a **python3** script, `phase9.py`, that reads `phase8.txt`, and replaces the two fields by their sum. The output is in `phase9.txt`. For example, the first line of output should be

```
aghora,294
```

Phase 10: Bash (10)

Write a **bash** script, `phase10.sh`, that sorts `phase9.txt` in a descending order according to the weight to output `phase10.txt`. For example, the first line of output should be

```
savitr,294
```

Phase 11: Gnuplot (20)

Write a **gnuplot** script, `phase11.gnu`, that plots the data in `phase10.txt` in a histogram format. The title, axes and legends should be properly marked. The output figure must be in color and directly produced to `phase11.pdf`.

Phase 12: Bash (15)

Write a **bash** script, `phase12.sh`, that automates the running of the 11 previous phases. It should take care of the dependencies correctly. It should also include the proper commands to run each of the phases.