**Word Task Server**

Each BSE 2017 student is required to perform the following tasks on words, including the following commands and functions

1. Doubling the word eg. double sara gives sarasara
2. Reverse the word eg. rev sar gives ras
3. Delete characters, specified by a comma separated list of numbers. E.g.chair 1,2 results in the word air
4. Replace characters, specified by a comma separated list of numbers. E.g. middle 1-h, 2-u, 3-s, 4-t generates a word hustle
5. Encrypt –eg encrypt zoo
6. Decrypt –eg.

In order to encrypt a string, perform the following steps

* Assign numbers starting from 1 in ascending order to the letters of the alphabet. Eg. a=1, b= c,…. A= 27, B=28 etc
* Start replacing the first character of the string you want to encrypt
  1. If the character is greater than 9, insert the two numbers that represent the encrypted string separated by space and follow the steps below to continue until you get a single digit number e. g. if z then 2 6, otherwise, replace it with the capital letter corresponding to that number.
  2. Subtract 1 from 26, and that is the character in small letters to insert between the string e.g. 2y6
  3. Subtract 9 from the previous letter holder e.g. 26-9 = 17
  4. Put the above result into in between the numbers but after the character in the second sub section. Eg 2y17 6
  5. Subtract 1 from 17 and identify the character at that position eg. 2y1p76
  6. Subtract 9 from 17 = 8. If it is a single digit number, pick the corresponding character and insert the character in capital letters in the middle of the string eg. 2y1Hp76
  7. Move to the next character in the string you want to convert and perform the steps from a to f, until the string is complete

Decrypting is the opposite of encrypting and ensure that you get the original string. Many jobs can be sent at once by one person, separated by semi colonies. In that case, the server applies the priorities, while processing one by one.

All jobs are submitted via commandline, on a client window, which is the interface for submitting the students’ jobs.

The tasks are submitted to a server, which receives all the tasks and processes them. Each task is identified by the ID of the person who sent it, time and date job was submitted, Job ID, processing duration and type of task. All these logs are stored in the database table. The server can process up to 1 task and processed results are stored in a file known as ready\_jobs.The jobs are read from the file by a PHP script every 30 seconds and inserted into a database table and thereafter the record cleared from the file. Tasks that arrive during the time the server is busy, are saved to a file called busy\_list and priority given to them as follows. A given job receives a priority number based on the sum of the following conditions:

1. Priority 1, which is the highest if job contains the least number of characters
2. Priority 2, if owner has the highest number of waiting jobs
3. Priority 3, if replace function

Jobs with the least number of priority are scheduled first while Jobs with the same priority are serviced on first-come-first-serve basis. Jobs that contain more than 50 characters are blacklisted and the reason is indicated on the web interface on request.

Design a web interface, which assists server administrators to perform the following:-

1. Calculate the average rate at which a server processes the jobs, within a specified period of time
2. Display which student’s jobs have the highest success rate and which percentage
3. Display which student’s jobs have the highest failure rate and which percentage
4. Which jobs are waiting for processing
5. List of waiting jobs with priorities
6. List of ready jobs with priorities

A person can also use commands to see the status of their jobs. For the pending jobs, the priorities shall be indicated plus the estimated time before the jobs can be executed. All results are returned to the client and a copy stored in a database

**Deliverables**:

Design document – Deadline 13th June 2018

Implementation: Deadline: 31st July 2018

Presentation Deadline: Deadline 7th August 2018

Note:

1. You will show your work to your supervisor every week and ensure that you sign
2. Design documents shall be submitted to the coordinator
3. Each group shall submit its own design document
4. You will present to a person, other than your supervisor (as shall be indicated by the coordinator)
5. There will be individual marks during the presentation and the supervisor shall submit an assessment for each of the members they will supervise
6. Use C programming and web technologies including PHP, CSS, Javascript etc