**Existing script:**

We have a script "nanp-script.js", which returns the region (state) corresponding to a phone number. To use this you need to call "compareNumber" function of this script and pass the array of numbers. It will return you the array of regions for those numbers.

To initialize this script we need to call "readFile" of this script. Pre-requisite for this script to run is nanp.xlsx, which is used for the logic of calculation of numbers and regions.

**Task:**

1) Make a front-end web UI, through which user can upload an xlsx file. This file will have one column which has phone numbers. We can assume that the first column of the file will have the phone numbers, or make an input box in the front end web form where user can enter the number of the column which has the phone numbers. Sample file is included input.xlsx.

2) Once the file is uploaded, the node application will read all the phone numbers from the file.

3) Now use the external script "nanp-script" as a module in your node app, and evaluate the region of all the phone numbers in the xlsx file. Make a new column (region) in the file and add corresponding region of the phone numbers in the file.

4) Once the processing is complete, send the file back to the user.

**Note:**

1) The file can contain lakhs of rows, so design should be optimized to handle such large file.

2) The generated file should be sent over web sockets, since for larger file the processing might take time and the http session might time out

3) There can be multiple users uploading the file at the same time.

4) It depends on you whether you do parallel processing of the phone numbers in the input file/serial, but optimization should be taken care.

5) You are free to make changes in nanp-script.js file according to the need.