**Problem 1**

A Multi city Clothing store has been running for quite sometime now. But they were not able to make as what their customers wanted to say to them, after all to run a successful business you need to hear the voice of your customers.

So recently they started to collect their customers voice in the form of surveys, and they have collected a huge data set of around 15,00,000 customers, they asked around 13 questions to their customers and now they are unable to understand the data set.

To help them out it is your responsibility to show them the summary of this data set in the form of distribution analysis which a very basic form of analysis. For example, in your school you must have obtained marks in your exams in the form of percentage so the same needs to be done here.

**Objective**

You need to show the distribution for the following categories.

1. Overall Rating of the stores.
2. City wise Shopping Experience
3. City with respect to what customers didn’t like in the store

For example, if we want to see the **Gender wise Age group distribution** then the output will look something like this.

|  |  |  |  |
| --- | --- | --- | --- |
| **Gender/Age Group** | **13-18** | **19-27** | **28-35** |
| Male | 45% | 15% | 31% |
| Female | 55% | 85% | 69% |

**Given**

You are provided with the following items:

1. A data set of 15,00,000 customers in the form of CSV file
2. A Map file to refer for all the questions and variables in the previous file.
3. A JSON file which will help you to cross check the type of the question and its valid options

**Each question is structured as follows in the JSON**

{

"name": "Gender",

"options": [

{

"code": 1,

"label": "Male"

},

{

"code": 2,

"label": "Female"

}

],

"type": 1,

"variable": "v1"

}

**Where:**

**Name**: Name of the question

**Options**: Available options of the question with their code and label

**Type**: Type of the question such as single choice (denoted by 1) or multiple choice (denoted by 2)

**Variable**: variable code used in the data file.

**A sample row from the data file will look like this:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| v1 | v2 | v3 | v4 | v5 | v6 | v7 | v8 | v9 | v10 | v11 | v12 | v13 |
| 2 | 1 | 3 | 3|6|1|2|5 | 5 | 1 | 1 | 4 | 4|1|3|2|5 | 2 | 3|5|4|2 | 1 | 2 |

**Please note:** The top row is the Header for the variables used for each question and as mentioned earlier few questions are multiple choice question so for them the response is captured in the form of pipe separated codes (‘|’) in the above row [variables v4, v9 and v11 are multiple choice variables].

***Bonus points will be given to those whose execution time is less than equal to 5 seconds.***

**Problem 2**

The **first part** of the problem is to write a function which accepts two number parameters and multiple them. But the only thing you are restricted is to use the multiplication operator (\*) however you are free to use + or – operators.

The **second part** of the problem is to identify the greater parameter of the two and then find the factorial value of it without using any loop method.

Once the first and second part are solved all you need to do is to multiple the results with the help of the first part of this problem.

**For example:** if the function is input with parameters (4, 3) then the first part should result 12 and then the second part should result in 24. The result should be 12\*24= 288.

**Note**: Both second and first part require you to use recursion and no loop methods. The final result should be calculated using the first part of this problem.

**Problem 3**

**Convert 7 segment display to decimal digits**

We have recently decided to digitise our old invoice archives. Since finding a volunteer for such an arduous task was impossible, an employee was selected at random and instructed to type in all invoice numbers into a text file. Little did we know that the employee we picked is an aspiring ASCII artist. Instead of handing us a file containing a set of numbers, we ended up with 7-segment display representations of the invoice numbers. This is where you come in. Write an application (choice of your language - preferable Python, TS, JS) that allows the user to upload a text file of 7-segment invoice numbers, and outputs a list with the parsed invoice numbers.

**Input**:

* A text file containing several hundreds of invoice numbers in the following form:

\_ \_ \_ \_ \_ \_

|\_ | || | | |\_| \_| ||\_ |\_

|\_||\_||\_| | | \_| | \_| \_|

Invoice number format:

**Note:** You can use input\_user\_story\_1.txt to feed your program.

* Each invoice number is constructed of 9 digits [0..9]
* Invoice number is written using \_ and | characters.
* Invoice number input takes 4 lines
* The first 3 lines contain 27 characters.
* The fourth line is blank.

**Output:**

* A text file with the parsed invoice numbers. One number per row.

Example:

123456789   
490867715

Note: You can use output\_user\_story\_1.txt to test that the output file generated by your program is correct.