1) Car Selling Platform

Problem Statement:

We need to build a selling and buying platform for used cars and bikes. A seller can list his vehicle by giving the vehicle details and price. Buyer / viewer can search for the cars via multiple filters like city, model, year, price. Viewers can also see all the vehicles listed by a single seller.
An additional feature would be a comment section attached to each vehicle where users (seller or buyers) can comment. These comments will be visible to all the users.

You need to design classes and their correlations and come up with functions for

search vehicles that must take multiple search parameters.
vehicle details that will return details and comments attached to that vehicle
No need to do the login / signup part.
Input:

This is an open-ended question that will evaluate your OOP, data structures and general design skills.
You can write your assumptions in comments
There are no test cases for this program. So, no need to read anything from the file.
Output:

The running code with classes and functions
Some test code to mimic different scenarios defined above.
Code MUST be executable, otherwise it won't be submitted.
No need to save any data to the database or file system. You can hardcode some test data if needed.
Final code should be properly commented and readable.
BONUS PART

Q: Let's assume that your platform has become popular and its listing has crossed 500,000 active vehicles and increasing on a daily basis. A problem that comes with it is slower search results. What measures can you take to make your search faster or even avoid this situation from happening?

Note: Add your answer in comments at the end of the code. No need to do the changes.

2) Google assistant needs your help

It's April 2016, one month before the "Google Assistant" launch. Google developers have been excessively testing the assistant and everything looks great. However, the assistant is having a hard time reading some phone numbers in documents. Good news is that Google has figured out the bug and it is because every country has unique ways to write phone numbers i.e. In Pakistan, the phone numbers are 11 digits like : 03364646555 or 12 digits like 923364646555. Some people divide the numbers into 4 - 7 format like 0336 4646555 or 5 - 7 format: 92336 4646555. While some divide the numbers into 3 - 2 - 7 format: 923 36 4646555. Different formats lead to different ways to read these numbers.

+ 923 36 4646555 reads nine two three three six four six four six triple five

92336 4646555 reads nine two double three six four six four six triple five

Google has devised some rules for its assistant and trained her on these rules:

Read single numbers separately
For 2 successive numbers use double
For 3 successive numbers use triple
For 4 successive numbers use quadruple
For 5 successive numbers use quintuple
For 6 successive numbers use sextuple
For 7 successive numbers use septuple
For 8 successive numbers use octuple
For 9 successive numbers use nonuple
For 10 successive numbers use decuple
More than 10 successive numbers read them all separately
Now that the code has been updated, Google has asked for Arbisoft's expertise to test the assistant. Your task is to generate an output containing the expected output in English after reading a file of phone numbers and provide those files to Google so that they can match with Google assistant's output. Google has provided you with some instructions for those files you will send them. They want you to group the same number written in different formats together in the output file.

Input
First line will contain the number of phone numbers, N in the file followed by the list of phone numbers from different countries. Phone numbers can contain different delimiters to separate the grouping of countries. Phone 3 - 3 - 4 format number can be written as (425) 555 - 1212, (425) 555 1212, 425 555 1212 etc.

Output
Output will contain the phone number followed by the unique ways it can be read. Phone numbers must be displayed in the same order in which they appear in the input file.

Same phone number can have different ways it can be written in the input file i.e., 0333 4507281, 033 3450 7281, but it must appear only once in the output file and without any format 03334507281.

Plus, sign (+) can be a prefix in some cases but it must not be part of both English translation and phone number in output.

Assumption / Restrains
A delimiter can be a space ' ', hyphen ' - ' or round brackets '()'

Although these two numbers 923364646555 and 03364646555 are the same, for sake of simplicity assume these are different numbers.
Sample input and output files are given below

Sample Input:
6

923 33 3303600

(425) 555 - 1212

425 - 555 - 1212

923 3333 03600

0333 303600

+ 1 - 650 - 513 - 0514

Sample Output:
923333303600

nine two three double three double three zero three six double zero

nine two three quadruple three zero three six double zero

4255551212

four two five triple five one two one two

0333303600

zero triple three three zero three six double zero

16505130514

one six five zero five one three zero five one four

3) Text Encode

Problem Statement
The year is 2035, humans have finally found sentient existence in space on a far-off planet and the species existing there seems to be peaceful and as much as in search of us as we were in search of them.

Now after finally finding them, the one problem that remains is communicating with them. Linguists from both civilizations cross their heads, try to research other civilizations' words, letters and come up with some solutions.

After days of brainstorming, they come across a draft which they think will at least break the ice between the two civilizations.

The aliens come up with this bizarre program after studying English letters and ask humans to code them in the language of their choice to be able to run on their earthly computers.

There’s a 6x5 matrix of all the capital letters, and the program should mention the instructions to type the word. For each word, the cursor always starts with the starting 0, 0 position which is A.

Whenever one goes down, they print ‘d’, ‘u’ for upend ‘l’ (small L), ‘r’ for left and right respectively.

When cursor is exactly at the letter you need to be, program prints ‘#’.

Since this program needs to print instructions of big words, it has to be very fast in execution.

ABCDE

FGHIJ

KLMNO

PQRST

UVWXY

Z

Like for example, to print ‘UP’ the instructions would be ‘dddd#u#’.

At the start the cursor will be at A position, we will need to do the “d” operation 4 times to come down to 5th row, we will be at position “U” then where we need to be, so we print “#”, then to go to “P”, we need to go up again from that position so we print “u”and then we are at “P” so we just print “#”.

The aliens don’t need space to separate words or punctuation characters for now to get the gist of the language, you can print instructions of words only ignoring the space between them.

Now, to make the encodings consistent for each word, we have set up some rules that:

We will ALWAYS do the vertical operation first. Like if we want to go from A to G, we will always do the “d” operation first& then the “r” operation.
After we are in our relevant row, then we will perform the “l” or “r” operation.
The program has to be efficient, we want to reach our appropriate letter in the shortest way possible.
Write a program which takes a word as input and outputs the instructions to print it.

Input
The input will be a sentence consisting of words (in all capital letters) separated by spaces. The input might have non alphabetical characters which need to be ignored.

If n is the total number of characters in the input sentence,

1 n 5000

Output
Output will be the print instructions of each word.

Example
Test Case 1:
Input: UP

Output: dddd#u#

Test Case 2:
Input: UP YOU GO

Output: dddd#u#ddddrrrr#uu#ddllll#dr#drrr#

4) API MODIFICATION

Lost World
Lost World

You have been traveling back and forth in time with a group of millionaires and scientists. A time machine accident has left your group stranded in the ancient past. Your only hope of being rescued is to screw up the timeline so much that the time police will notice. Messing with timelines is almost an impossible task. It requires executive access to the timeline data and understanding of complex operations. Your fellow fresh graduates that are now working in arbisoft have noticed that you have been missing from the office for more than 5 years now. They have decided to open up a secret api portal that will allow you to receive data that can possibly help you disrupt the timeline.

Though most of the data retrieved from the secret portal makes no sense. Your only hope is to interpret the data from the secret portal and respond back to the time police helpline with a secret key number that will disrupt the timeline.

Scientists have suggested, following random data manipulation might help us generate a final number required to disrupt the timeline. The only reasonable input from millionaires is an influencing quote “Hustle but strictly stick to guidelines and remember your aim, that is to get us back”.

Here are the list of steps your group has decided to perform:

Read the trip id from the time machine logs e.g., input file
Input File

b/ITZH

Use trip id from input file to retrieve timelines, their masks labels and action plan using timelines portal api:
Api-URL:

https://jsonkeeper.com/b/ITZH

Sample Response:

```
{"status": "True",
    "data": {
        "timelines": [ 1, 2, 44, 555, 0, 0],
        "masks": ["Are", "Are", "you", "HIGH", "you", "HIGH”],
        "action_plan": {
            "operation": "FOO",
            "sub_operations": {
                "Are": "BAR",
                "you": "FOX",
                "HIGH": "BAR"
            }
        }
    }
}
```

Timelines are numeric values. Each index of the timeline array has a corresponding mask value stored in the masks array. For example index 0 of

timeline array has value 1 with corresponding mask value of “Are”. Index 5 of the timeline array has value 0 with corresponding mask value of “HIGH” and so on.

“Action_plan” variable consists of operation and sub_operations. Sub operations are defined as key value pairs. You need to perform a “BAR” operation on the data that belongs to the “Are” mask label.

Operation and sub-operations will be performed following the steps mentioned below:
Group the values from timelines array that belong to the same mask labels.
Use translation table below and information retrieved from timeline api to perform sub operations on the data grouped in step 1 for each respective mask.
This shall output a single value for each mask label.
Perform final operation using translation table below on the resulting data set from different masks sub-operations.
Feed the final result to the time police helpline e.g., output file in our case.

Translation Table

Operation Keyword

Actual Operation

FOO

SUM

FOX

MAX

BAR

MIN

```
Example Scheme Dry Run



Sample Data retrieved from timelines api
 "timelines": [ 1, 2, 44, 555, 0, 0],

 "masks": ["Are", "Are", "you", "HIGH", "you", “HIGH"]



"operation": "FOO"

"sub_operations": {“Are": "BAR”, “you": "FOX", "HIGH": "BAR"}



Timelines grouped by mask labels
“Are”: [1, 2]
“you”: [44, 0]

“HIGH”: [555, 0]



Resultant data after performing sub-operations on masked labeled data as per
translation table above


 “Are”:  Min ([ 1, 2])  = 1

 “you”:  Max ([44, 0] )    = 44

“HIGH”: Min ([ 555, 0 ])      = 0



“data” : [1, 44, 0]



Final Step
Operation: “FOO” translates to SUM operation

1+44+0 = 45
```

```
Output: 45



Limitations & Assumptions

For all test case this condition will holdSize of Timeline Array == Size of Masks
Array
0  Size of Timelines /Masks Array  10^6
0  Number of SubOperations  Size of Mask Array
Number of Operation = 1
For most cases there will be sub operation for any given key that is part of the
masks array. In case a sub operation is missing for any mask label, the
calculated value for that mask can be assumed to be 0.
Lookout for api portal timeouts !
```