COMPULSORY QUESTIONS:

You should take no longer than 3 days to complete the following 4 questions. Do strictly follow the language requirement for each question.

Question 1 - Royal Rumble (Java only)

An ordinal number is a word representing rank or sequential order. The naming convention for royal names is to follow a given name with an ordinal number using a Roman numeral to indicate the birth order of two people of the same name.

The Roman numerals from 1 to 50 are defined as follows: The numbers 1 through 10 are written I, II, III, IV, V, VI, VII, VIII, IX, and X. The Roman numerals corresponding to the numbers 20, 30, 40, and 50 are XX, XXX, XL, and L. For any other two-digit number < 50, its Roman numeral representation is constructed by concatenating the numeral(s) for its multiples of ten with the numeral(s) for its values < 10. For example, 47 is 40 + 7 = "XL" + "VII" = "XLVII".

In this challenge, you will be given a list of royal name strings consisting of a given name followed by an ordinal number. You must sort the list first alphabetically by name, then by ordinal increasing within any given name.

For example, if you are given the royal names [George VI, William II, Elizabeth I, William I] the result of the sort is [Elizabeth I, George VI, William I, William II].

Your task:

Download the project and complete the function `getSortedList` in RoyalRumble.

https://drive.google.com/open?id=16Klg5pqWAPSWAgZi-QzK1RN2d8abvXwd

`getSortedList` takes in a list of royal name strings and must return the list of names sorted first by given name, then by ordinal.

You are to submit only the implemented RoyalRumble.java. Please do not change the basic structure of RoyalRumble.java so it will still work with Main.class and run.sh.

Constraints:

There will be between 1 and 50 names on the list.

Each name is a single string composed of `firstName` and `ordinal`, separated by a space.

`ordinal` is a valid Roman numeral representing a number between 1 and 50, inclusive.

The length of `firstName` will be between 1 and 20.

Each `firstName` comprises only uppercase and lowercase ascii characters [A-Za-z].

Sample Input:

```
Louis IX
Louis VIII
David II
```

Sample Output:

```
David II
Louis VIII
Louis IX
```

Question 2 - TAIGER CAFE (Bash only)

Since Taiger HQ is situated opposite a fruit stall in chinatown, Taiger is thinking of opening Taiger Cafe to earn more profit. The Cafe serves 3 items, with the following ingredients required:

- APPLE PIE - 3x Apple
- PINEAPPLE PIE - 3x Pineapple
- FRUIT PARFAIT - 2x Apple + 2x Pineapple

You are an inventory manager in charge of checking if we can fulfill a customer's orders with the ingredients we have in stock. Luckily, the inventory list is contained in a text file. Write a bash script named `recipeChecker.sh` which will let us know if we can fulfill the order.

Your task: Download the project with the sample `fruitList.txt` and implement the bash script `recipeChecker.sh`
https://drive.google.com/open?id=1mewjHA1Pfn1AJPueOfQX6R_3Dc5dflUw

Your script should take in 2 command line arguments:

- Arg 1: absolute file path of the inventory list (e.g. `fruitList.txt`)
- Arg 2: the customer's order consisting only one of ["APPLE PIE", "PINEAPPLE PIE" or "FRUIT PARFAIT"] in BLOCK letters

Your script should output the following to stdout:

- If there are enough ingredients to create the specified dessert, output **exactly (including all punctuation)** : `You shall have (recipe name in block letters)!`
- Else, output: `You shall not have (recipe name in block letters)`
- If an unrecognised recipe is provided (eg Pineapple Pie), output: `We do not have that on the menu`

Other Specifications

You should consider all fruits to be case insensitive (i.e if a list contains "Apple" and "apple" you can consider them as 2 apples)
Each line of the inventory list provided will contain only one item, and terminates with a ","
Any item which contains "apple", such as "apples", "apple jam" or "rotten apple" should not be considered an "apple" ingredient.
See the following examples of how the inventory list will be formatted

Example

fruitList.txt

```
Pineapples,
Apple,
Pineapple,
Orange,
APPLE,
Rotten apple,
Grape,
apple,
```

Sample 1

Command: `./recipeChecker.sh fruitList.txt "APPLE PIE"`
Output: `You shall have APPLE PIE!`
Explanation: Lines 2, 5, 8 contain 3 apples together

Sample 2

Command: `./recipeChecker.sh fruitList.txt "FRUIT PARFAIT"`
Output: You shall not have FRUIT PARFAIT
Explanation: Lines 2, 5, 8 contains 3 apples, line 3 contains 1 pineapple. We need minimum 2 of each.

Sample 3

Command: `./recipeChecker.sh fruitList.txt "CHICKEN PIE"`
Output: `We do not have that on the menu`
Explanation: Menu only contains PINEAPPLE PIE, APPLE PIE and FRUIT PARFAIT

Question 3 - TAIGER CAFE EXPANSION (Bash only)

As Taiger Cafe expands, they decide to increase the menu items. Unfortunately, there were too many orders that went unfulfilled. Taiger's Data Scientists request for some data so the Taiger Cafe Logistics Team may review how to better manage the inventory based on customer demand.
You decide to write another bash script named `unfulfilledOrders.sh` to help the Data Scientists which will output the 3 most recent unfulfilled orders for the specified date and order, sorted by timestamp from oldest to newest.

Your task: Download the project with the sample `fruitList.txt` and implement the bash script `unfulfilledOrders.sh`
https://drive.google.com/open?id=1dAh17zZRCt_dsTT9Uo5DsampKwiT92AW

Your script should take in 3 command line arguments:

- Arg 1: absolute file path of the order list (eg orderLists.txt)
- Arg 2: the customer's order

- Arg 3: the order date in YYYY-MM-DD format

Other Specifications

There is an infinite list of orders that are not predefined. You can consider the orders to be case insensitive. You may also get orders which contain non-alphabetical characters.
Each line of the order list provided will contain only one order at one timestamp
If there are fewer than 3 orders which fulfill the requirements, you can just display those.
The order list provided contains orders in a random order
See the following examples of how the order list will be formatted

orderList.txt

```
2019-01-18 15:30:00 order="STRAWBERRY SHORTCAKE" fulfilled=FALSE
2019-01-18 16:40:00 order="STRAWBERRY SHORTCAKE" fulfilled=FALSE
2019-01-18 13:15:00 order="STRAWBERRY SHORTCAKE" fulfilled=FALSE
2019-01-18 10:15:00 order="MILLE CREPE" fulfilled=FALSE
2019-01-18 08:30:00 order="MILLE CREPE" fulfilled=TRUE
2019-01-18 13:25:00 order="STRAWBERRY SHORTCAKE" fulfilled=TRUE
2019-01-18 17:00:00 order="STRAWBERRY SHORTCAKE" fulfilled=TRUE
2019-01-17 10:30:00 order="STRAWBERRY SHORTCAKE" fulfilled=FALSE
2019-01-17 18:30:00 order="STRAWBERRY SHORTCAKE" fulfilled=FALSE
2019-01-18 12:15:00 order="STRAWBERRY SHORTCAKE" fulfilled=FALSE
2019-01-18 11:30:00 order="MILLE Crepe" fulfilled=FALSE
```

Sample 1

Command: `./unfulfilledOrders.sh orderList.txt "STRAWBERRY SHORTCAKE" 2019-01-18`

Output:

```
2019-01-18 13:15:00 order="STRAWBERRY SHORTCAKE" fulfilled=FALSE
2019-01-18 15:30:00 order="STRAWBERRY SHORTCAKE" fulfilled=FALSE
2019-01-18 16:40:00 order="STRAWBERRY SHORTCAKE" fulfilled=FALSE
```

Explanation:
Lines 1-3, 6-7 and 10 contain STRAWBERRY SHORTCAKE orders on 2019-01-18
Lines 1-3, and 10 contain unfulfilled (fulfilled=FALSE) orders
Arrange lines 1-3 and 10 in ascending order, and the last 3 timings are 13:15:00, 15:30:00 and 16:40:00

Sample 2

Command: `./unfulfilledOrders.sh orderList.txt "MILLE CREPE" 2019-01-18`

Output:

```
2019-01-18 10:15:00 order="MILLE CREPE" fulfilled=FALSE
2019-01-18 11:30:00 order="MILLE Crepe" fulfilled=FALSE
```

Explanation:
Lines 4-5 and 11 contain MILLE CREPE orders on 2019-01-18
Lines 4 and 10 contain unfulfilled (fulfilled=FALSE) orders
Arrange lines 4 and 10 in ascending order, and the last up to 3 timings are 10:15:00 and 11:30:00

Question 4 - Defender Arcade (Java only)

TAIGER employees love playing video games, so they have Defender Arcade within the company (Work hard-Play hard). Since everyone is quite busy with work, everyone has provided the time (start time and finish time) when he or she wants to play games. If play time of two employees overlap then they start fighting and stop working. The boss got to know about this situation and asked you to help him by calculating the minimum number of Defender Arcades needed so that every employee can play during their specified time.

Note: If one employee is leaving and at the same time another employee is starting then only one Arcade is needed.

Your task: Download the project and complete the function `countArcades` in DefenderArcade.

https://drive.google.com/open?id=1WFR6e2wN4WCNe9ienpB4tt4fu3qp7xlo

`countArcades` takes in a list of employees' play times and returns the number of arcade machines needed.

You are to submit only the implemented DefenderArcade.java. Please do not change the basic structure of RoyalRumble.java so it will still work with Main.class and run.sh.

Constraints:

There will be between 1 and 100 time periods in the list

Each item in the list is a string composed of start time and end time, separated by a space.

Times are denoted in 24-hour format. For example 915 means 9:15AM, 2145 means 9:45PM, etc.

Output:

Single integer denoting minimum number of Defender Arcade needed.

Sample Input:

```
900 910
940 1200
950 1120
1100 1130
1300 1400
1350 1420
```

Sample Output:

```
3
```

Explanation:

Between 1100 and 1120, there are 3 employees who wants to play, so a minimum of 3 Defender Arcades are needed.

OPTIONAL: NLP ASSESSMENT

Please answer the following questions if you are keen to be considered for a role in the NLP function / Software NLP position. This is not compulsory, but will serve as an advantage in your application.

Question 5: Basic NLP

Given Natural Language paragraphs, come up with NER module to identify Person Name, Location and Organization. Assume that there is no training data available and need to use linguistic features to tag the entities.
In case any assumptions are made, please write them clearly in Readme file.
Corpus: news.txt
Example input: Jim bought 300 shares of Acme Corp. in 2006.
Output: <Person>Jim</Person> bought 300 shares of <Organization>Acme Corp.</Organization> in 2006.

Question 6: Machine Learning

The Enron e-mail corpus (https://en.wikipedia.org/wiki/Enron_Corpus) is a publicly available archive of Enron corp e-mails. The goal of this challenge is to cluster e-mails from different employees based on their content.

The dataset can be downloaded from here. Each link leads to the contents of the inbox of an individual:

Assumptions: Make any number of simplifying assumptions so long as they are clearly stated. For instance - 'attachments will be ignored' OR 'only inbox and sent_items will be considered' etc.
**Things we look for : Data wrangling code should be clean and modularized*
Step 1: Text pre-processing and feature extraction and selection
Step 2: Clustering model selection : What clustering algorithms did you use and why?
Step 3: Visualization and/or evaluation: In an unsupervised setting how do we evaluate cluster cohesiveness?
Step 4: Generate top keywords in each cluster to get a sense of the underlying structure
There's no right answer. We are just looking for a good workflow for natural language processing tasks.
Deliverables: Code and/or Jupyter notebooks and a write-up of your approach consisting of the entire workflow reasoning about the choice of your algorithm, key assumptions made
Corpus: maildir.zip

Submission

Write documentation in code or in readme file as necessary, you will not be given a chance to explain your assignment of any ambiguity in any of the next phases. **Please upload your answer as 1 zip file** containing a separate folder for each question, including instructions on how to run the program. If you did not manage to complete any questions, let us know why. (Max 100mb). Attach the zip to the email or upload the zip in an online storage and share with us the link. We will not process your submission if you did not do it exactly this way.