Stage 2

The 2 challenges below can be found on Hackerrank. We have simply copied them here for convenience.

To submit your results, make a public repo on Github called "NLP Challenges" where you should upload your code, as an IPython Notebook, in different folders for different challenges. Share the link with us and we would review the code and your approach and get back to you.

If it is found that the code was plagiarised, the applicant would get disqualified immediately.

Please note: If you have already created a repo for an earlier challenge, just add solutions to these as different folders in the same repo.

1. Matching Book Names (Super Relevant)

Our bot has crawled several product pages from the popular Indian e-commerce website, Flipkart.com. All of these pages are specifically about the most popular books being sold on Flipkart (at the time of the crawl).

Each page contained information for exactly one book. We noted down exactly two fields from each of these pages:

```
1. The name of the book.
2. Description fragment: The first few sentences of the description of the book,
```

```
as displayed on the page. In some cases,  this string or text field might be
terminated prematurely (i.e., not exactly at a word or a sentence boundary).
```

Each of these text blocks is split into two parts of roughly equal length.

Set A contains the **names of all the books**. **Set B** contains the **description fragments** for all the books.

Both the Sets A and B are shuffled up, and the ordering of elements is lost.

Your task is to identify, for each name (*a*) in **Set A**, which is the correct corresponding text fragment (*b*) in **Set B**, such that, *b* was the descriptive fragment for the book named *a*.

Hints: Getting started - Think about using TF-IDF Scores (or a modification of it)

For those getting started with this fascinating domain of text classification, here's a wonderful Youtube video of Professor Christopher Manning from Stanford, explaining the TF-IDF , which you could consider using as a starting point.

Input Format

An Integer N on the first line. This is followed by 2N+1 lines.

Text fragments (numbered 1 to N) from Set A, each on a new line (so a total of N lines).

A separator with five asterisk marks "*****" which indicates the end of Set A and the start of Set B.

Text fragments (numbered 1 to N) from Set B, each on a new line (so a total of N lines).

Output Format

N lines, each containing one integer.

The i-th line should contain an integer j such that the j-th element of **Set A** and the i-th element of **Set B** are a pair, i.e., both originally came from the same listing page on Flipkart.

Constraints

1 <= N <= 1000

No text fragment will have more than 10000 characters.

Sample Input

```
5
How to Be a Domestic Goddess : Baking and the Art of Comfort Cooking (Paperback)
Embedded / Real-Time Systems 1st Edition (Paperback)
The Merchant of Venice (Paperback)
Lose a Kilo a Week (Paperback)
Few Things Left Unsaid (Paperback)
*****
Today the embedded systems are ubiquitous in occurrence, most significant in function and project an absolutely promising picture of developments in the near future.
The Merchant Of Venice is one of Shakespeare's best known plays.
How To Be A Domestic Goddess: Baking and the Art of Comfort Cooking is a bestselling cookbook by the famous chef Nigella Lawson who aims to introduce the art of baking through text with an emphasis.
Lose A Kilo A Week is a detailed diet and weight loss plan, and also shows how to maintain the ideal weight after reaching it.
Few Things Left Unsaid is a story of love, romance, and heartbreak.
```

Sample Output

```
2
3
1
4
```

5

Explanation

Explaining the Input

The first line indicates that the test case contains the names and descriptions of five popular books listed on Flipkart.

The next five lines are the names of the books (i.e, **Set A**). After that, we have a separator. That is followed by five lines, each containing description fragments from **Set B**.

Explaining how we arrived at the Output

The first description, is visibly most closely related to the second book (Embedded / Real-Time Systems 1st Edition (Paperback)).

The second description, is clearly about the Merchant of Venice - which is the third book name in Set-A.

The third description is about Baking - and so, it corresponds to the first of the book names, in Set-A. Similarly, the fourth and fifth descriptions match best with the fourth and fifth book names (i.e, it so happens that they are already in order).

So, the expected output is 2, 3, 1, 4, 5 respectively.

Scoring

The weight for a test case will be proportional to the number of tests (book names) it contains. Two sample tests are available and visible on **Compile & Test**. A training driven approach or solution is not expected in this challenge, which is why no comprehensive training data has been provided.

Score = M * (C)/N Where M is the Maximum Score for the test case.

C = Number of correct answers in your output.

N = Total number of book names in the test set (which were divided into Set A and Set B respectively).

Note: Submissions will be disqualified if it is evident that the code has been written in such a way that the sample test case answers are hard-coded, or similar approaches, where the answer is not computed, but arrived at by trying to ensure the code matches the sample answers.

Timelimits

Timelimits can be seen here.

Libraries

Libraries available in our Machine Learning/Real Data challenges will be enabled for this contest and are listed here. Please note, that occasionally, a few functions or modules might not work in the constraints of our infrastructure. For instance, some modules try to run multiple threads (and fail). So please try importing the library and functions and cross checking if they work in our online editor in case you plan to develop a solution locally, and then upload to our site.

2. Gender Prediction

You will be provided with a large corpus of text which contains one or more stories or snippets, which could either be present in full, summarized or in partial form. This text will contain the thoughts, actions, dialogues and interactions between various

characters. It will be segmented into paragraphs which may or may not have a continuity of thought and logic between them. **Your task is to create an intelligent program that guesses the gender of certain characters in this text, whose first names will be specified to you.**

There might be multiple characters who share the same first-name. If a test contains such a name, all such characters will share the same gender.

Text Corpus

This file corpus.txt needs to be read by your program.

Input Format

The first line is an integer N. This is the number of characters from this text, for whom you need to detect the gender.

This is followed by the first names of N people, from the corpus text, each on a new line.

Constraints and Nature of the Corpus

As mentioned previously, it will be segmented into paragraphs which may or may not have a continuity of thought and logic between them. You do not need to code defensively for obscure or ambiguous cases. Tests will be set in such a way that the genders of the characters will be reasonably direct and simple, for a human reader of the text to infer, based on the few sentences before or after the position where that name occurs in the text. Text has been taken from sources such as project Gutenberg, Wikipedia and publicly available news snippets, biographies and stories. To replicate the difficulty of dealing with much of the text available on the web, parts of it might be somewhat unstructured, and there might even be a small percentage of text which is not in English.

Number of Lines in Corpus = 65443

Number of Tests(N) will be such that 1<=N<=100.

Tests will contain first names, where only the first letter is capitalized. Names will only be from the sections of the corpus which are in English.

Output Format

N lines. Each line contains just one word: Male or Female. The i-th line contains the gender detected for the i-th first-name in the tests.

Corpus

The corpus will be provided to you here. Your program can read in the provided corpus by assuming that the file "corpus.txt" lies in the current folder of your program.

Example Input

```
3
John
Sherlock
Mary
```

Example Output

```
Male
Male
Female
```

Explanation and a few hints

On reading the text, if we need to infer the gender of John, we find several portions of the text which provide us with sufficient hints to figure out the gender with reasonable accuracy.

*"Mr. **John** Turner," ried the hotel waiter, opening the door of our sitting-room, and ushering in a visitor. "My nme," said **he**, "is **John** Openshaw, but my own affairs have, as far as I can understand, little to do with this awful business. It is a hereditary matter; so in order to give you an idea of the facts, I must go back to the commencement of the affair. "There is one thing," sid John Openshaw. **He** rummaged in **his** coat pocket,*

Scoring

The Max score for this problem is 50.

Score = MaxScore for a test case * Max(C-W,0)/N

C = Number of tests for which you guess the gender correctly

N = total number of tests (first-names) in the input file.

W = Number of tests for which you guess the gender incorrectly

ML Libraries are not enabled for this challenge because many of them have the potential to provide a near-direct solution for this challenge.