

Q1)

Balloon Decoration

Arbisoft is opening its new office in Germany and preparing to decorate the new office in a fun and colorful way. They used a repeating pattern of colorful balloons to make the office look amazing. These balloons came in three colors: orange, blue, and white, represented by the letters 'O', 'B', and 'W' respectively. The balloon pattern is like a magical melody that will play on a loop, giving the office a lively and enchanting aura.

Decorating a large office space was no small task. To expedite the process, Arbisoft divided the office into sections, and different teams were responsible for decorating each section. Each team had a starting point and an ending point within the office. Now, the challenge was to distribute the right number of balloons to each team based on their assigned area and the balloon pattern.

To solve this problem, Arbisoft sought the help of coding wizards like you. You will be provided with a pattern of balloons as a string like "obbwwbbo" and starting and ending index parameters specify the beginning and end points of the area each team is responsible for. Your job is to write a code that should tell us the number of balloons of each color needed for decoration within a specific range.

Input:

The input will be read from a file. The first line of the file will be the pattern. The second and third lines will be the starting and ending indexes respectively.

Output:

The output will be a lowercase string and the color order must be Blue, Orange, and then White along the number of balloons i.e. "b12o5w7".

NOTE:

The index starts from 0.

Starting & ending indexes are included.

Constraints:

The starting and ending indexes may either be equal to or fall within the range of 0 to 100,000,000.

Ending index > starting index

4 <= The length of the balloon pattern <= 15

Sample 1

Input:

bowbo

7

12

Output:

b2o2w2

Sample 2

Input:

owbowbob

17

48

Output:

b12o12w8

Sample 3

Input:

obbo

0

6478

Output:

b3240o3239w0

Q2)

ICC Champion trophy 2025

Pakistan is hosting the ICC champion trophy 2025. Top eight ICC ranked teams will participate in the contest. In total fifteen matches would be played in different venues of the country. Tens of millions of people will follow it online.

To make it interesting for the fans around the globe, Pakistan cricket board has come up with different ideas. One of them is to release the win predictability of each team against all other participating teams a week in advance of the tournament. They have score cards for all the matches played among the participating teams in history. All that data is available through the [API](#). It returns a list of matches, each match contains the score cards of both of the teams. Here is the structure of the response

```
[
  {
    "date": "02/02/2022",
    "team1": "Pakistan",
    "team2": "India",
    "venue": "MCG",
    "scoreCardTeam1": [
      {
        "name": "Fakhar Zaman",
        "score": 53
      },
      . . . .
    ]
  }
]
```

```

]

"scoreCardTeam2": [

    {

        "name": "Virat",

        "score": 10

    },

    . . . .

],

},

. . . .

]

```

You have been hired by the board to build the very first version of this predictor. Your task is to write a program that will take two countries as input, say A & B and will predict the winning chances of each of these countries out of hundred by following the given criteria

1. Percentage of number of times team A has scored more than team B will represent the winning chances of A, rounded by two decimal places
2. Percentage of number of times team B has scored more than team A will represent the winning chances of B, rounded by two decimal places
3. Each match has a weightage based upon how far in the past it was played
 1. Matches played in the current year has a weightage of 10

2. Weightage reduces by 1 with every year in the past
3. Weightage for all the matches played before 10 or more years would be 0.5

General Example:

Let's say Pakistan and Australia have played 10 matches in the history ever, five of them were played in 2024 and 5 were played in 2022. Pakistan scored more than Australia in two matches of 2024 and three matches of 2022. The weighted percentage of the number of times Pakistan won, will represent the winning chances of Pakistan, that is 48.89% in this example. Similarly the weighted percentage of the number of times Australia won, will represent the winning chances of Australia.

Input

The input will be read from a file, it will contain names of two teams, separated by comma for which prediction is required to be made.

Output

The output would be the win prediction for the given teams in same order, separated by comma

NOTES:

1. Current year is 2024
2. Each of the teams has played minimum of 10 matches with every other team
3. Team's score is equivalent to the sum of the each player's score available in the scorecard
4. Skip the draws(matches where scores of both teams were equal)
5. Country spellings in the input would exactly match with that in response of API

Sample 1

Input:

NewZealand,Pakistan

Output:

34.72%,65.28%

Sample 2

Input:

SriLanka,Netherlands

Output:

54.95%,45.05%

Sample 3

Input:

Pakistan,Netherlands

Output:

50.61%,49.39%

Input file reading Instructions:

The input is read from a file. The filename/path of the file is passed to your program as the first command-line argument. There is no fixed name for the input file. Do not hardcode the input file name.

API: l0l6pp2i0k.execute-api.eu-north-1.amazonaws.com/default/icc_matches

Q3)

Yet Another Chat App

We need your help in building a chat application for a new type of device called Com-link. Com-link connects to other similar devices with a limited cutting-edge network where we can send digits from 0 to 9 rather than binary 0 and 1. So we have come up with a newer model of encoding as described below.

Example 1:

Given the key:

“r”, “sw”, “x”, “c”, “nm”, “q”, “yt”, “z”, “ab”

If I want to send the message “bat” then the encoding result will be “99090770”

Two 9’s represent the second character in the ninth string, which happens to be “b”. 0 is a delimiter. One 9 represents “a” as the first character in the 9th string is “a”, then 0 as a delimiter. For “t” we had to use two 7’s.

If I want to send the message “goat” then encoding is not possible. As “g” is not available in the key.

If we want to send a message “BAT” then the result will be “09900900770”. Adding an extra “0” at the start of the character’s encoding turns it into uppercase.

Example 2:

Given the key:

“1abc”, “2def”, “3ghi”, “4jkl”, “5mno”, “6pqr”, “7stu”, “8vwx”, “90yz”

If I want to send the message “5 Roses” then the encoding result will be “5090066660555507702220770”

One 5 represents the first character in the fifth string, which happens to be “5”. 0 is a delimiter. One 9 represents the space as the first character in the ninth string is a space, then a delimiter. For “R” we had to use 0 for capitalization and four 6’s then a 0 delimiter. Four 5’s represent ‘o’ and so on.

On the receiving end, we have to transfer the encoded result back to the actual string using the same key by reversing the process.

Input:

The input will be read from a file. The first line of the file will be the key. The second line will describe the operation, 1 for encode and 2 for decode. The third line will contain the string to encode or decode.

Output:

The output will be a single line of encoded or decoded results or "Error" if decoding or encoding is not possible.

Constraints:

- The key will have a max of 9 strings.
- Encoding and decoding of a UTF-8 character set is expected.
- $0 \leq \text{Length of each string in the key} \leq 100$
- 0 will always be the delimiter.
- 0 at the start will indicate conversion to uppercase.
- $0 < \text{Length of string to encode or decode} < 1000000$

Sample 1

Input:

"r", "sw", "x", "c", "nm", "q", "yt", "z", "ab"

1

bat

Output:

99090770

Sample 2

Input:

"r", "sw", "x", "c", "nm", "q", "yt", "z", "ab"

2

099090770

Output:

Bat

Sample 3

Input:

"r", "sw", "x", "g", "nm", "q", "yt", "z", "ab"

1

Cat

Output:

Error

Q4)

eCommerce Store Design Problem

Owing to your impressive technical skills you are hired as the founding CTO of the next big ecommerce startup. Your first job as the CTO is to follow the system design requirements, and build an initial version of the system's functionality based on those requirements, using Object Oriented Design patterns.

System Design Requirements:

The system should allow:

- Having items from many different vendors.
- Having many different items from each vendor and thousands of different items in total.
- Having multiple types of users with the ability to perform different tasks as follows:
 - normal_user: These users can view and buy items.
 - store_manager: These users can do everything a normal_user can do, and can also add new items to an existing vendor.
 - admin: These users can do everything a store manager can do, and can also create new vendors and new users in the system.
- Creating new users.
- Logging in existing users.
- Creating new vendors.
- Adding new items to vendors.
- Viewing items (all items, all items from a vendor, or details of a specific item using a unique identifier).
- Buying items, while keeping track of the available stock for each item.

Following OOD and keeping the above requirements in mind, you should create classes and make use of object-oriented principles as you see fit. Feel free to make assumptions (writing them down as a part of your code), and extend the requirements if you want.

Your program must have some testing code that shows that the above requirements are met. There is no need to save any data to the database or file system, you can hard code test data if needed. Your final code is expected to be well-commented and readable.