Question 6

Not complete

Marked out of 0.50

Flag question

Implement the following programme:

- An **airlines** dictionary is given, where the keys are airline names and the values are dictionaries (where the keys are the plane names and the values are lists with the cities these planes fly to). This code is already written.
- The name of the city is read out.
- Determine how many planes fly to the given city and print this number.

For example:

Input	Result
Paris	2
Luxemburg	0

Answer: (penalty regime: 0 %)

Reset answer

```
airlines = {
    'Aero_zoom': {'12iy34': ['Milan', 'Paris', 'Kazan'], '193tyw': ['Rome
    'Aeroflot_express': {'12oa': ['Dublin', 'Tokyo', 'Rome'], 'llwo24': [
    'S7_avia': {'qw124': ['Rome', 'Milan', 'Moscow'], '1987ytr': ['Tokyo'
    'Love_airline': {'12kcsl': ['Paris']}
}

city = input()

# your code below
```

Question 7

Not complete

Marked out of 0.50

Flag question

Define a function that works with palindromes.

Note: palindromes are words that are read the same from left to right and right to left.

- Define a **palindrome** function that takes as input a list of strings (words to be checked).
- The function must check each word (whether it is a palindrome or not) case-insensitive.
- If a word is a palindrome, the function adds it to a new list.
- At the end, the function returns the list of palindromes. The words themselves should not change in any way. They are added to the list in the same order as they were entered.
- If there are no such words, the function returns an empty list.

Important: In this task you only define the function, you don't need to call it and read the values, it will happen automatically during the check.

For example:

Input	Result
`['Deed', 'NuN', 'leVel', 'kitten', 'deed']`	`['Deed', 'NuN', 'leVel', 'deed']`
`['water', 'kitten', 'fox']`	`[]`

Answer: (penalty regime: 0 %)

```
1
```

Question 8

Not complete

Marked out of 0.50

Flag question

At the IT Olympiad for schoolchildren the participants represented different regions. The results of the Olympiad are stored in a csv file. Write a programme that outputs all the lines corresponding to the winners in a certain region.

- The program should open a structured text file **olymp.csv**, in **utf-8** encoding, where each line contains the following information about the participant: his name, region, sum of points, the line **"Winner"** or **"Prizewinner"**, if the participant is a winner or prizewinner respectively. For example, **"Konstantin A., Moscow, 557, Winner"**.
- Commas are used to separate data in the file.
- It is guaranteed that there is at least one line with data in the file.
- A string, the name of the region, is entered from the keyboard.
- It is guaranteed that there is at least one representative of this region in the file.
- The programme should output the lines from the original file that correspond to the winners in the entered region (remember to remove the end of line characters!).

ADDITIONAL INFO

- Your solution will be tested on closed tests. Your code must read the **olymp.csv** file without any additional instructions (e.g. file path on your computer).
- The **olymp.csv** file is used in the open test. You can download it and test your solution locally.
- In the example, the first column shows the test number (0 — open test), you do not need to read this number or use it in your solution.

For example:

Test	Input	Result
0	Saint Petersburg	Alexander M.,Saint Petersburg,681,Winner Andrey Ye.,Saint Petersburg,583,Winner Sergey H.,Saint Petersburg,544,Winner

Answer: (penalty regime: 0 %)

Question 9

Not complete

Marked out of 0.60

Flag question

Petya wants to check which words and letter combinations occur most often in the names of the artists in his playlist. Write a programme to help Petya.

INPUT FORMAT

- The first line is used to input the tracks from Petya's playlist in the format '**<player name> | <song name>**'. The tracks are separated from each other by **a comma and a space**. It is guaranteed that the string consists of at least one track.
- On the second line, you enter the string by which you want to search for tracks.

OUTPUT FORMAT

- An integer — the number of tracks whose **artist names** contain the entered string.

*Note: the search should be case insensitive (for example, the words **"love"** and **"LOVE"** are considered as one word).*

For example:

Input
`Pearl Jam | Garden, Fools Garden | Lemon tree, Secret Garden | Song from a secret garden, S garden`
`ABBA | Dancing queen, Paula Abdul | Cold Hearted, Patti LaBelle | New Attitude, Hooverphoni AB`

Answer: (penalty regime: 0 %)

```
1
```

Question 10

Not complete

Marked out of 0.60

Flag question

Ron wants to find out which subjects he needs to study harder. Write a programme to help Ron.

INPUT FORMAT

- On the first line, enter Ron's grades in the format '**<subject name> : <grade>**' using commas and spaces. It is guaranteed that the line consists of at least one subject and its corresponding grade.
- On the second line, an integer is entered — the grade that Ron's mother is happy with.

OUTPUT FORMAT

- If there are subjects with grades greater than or equal to the grade that would satisfy Ron's mother, these subjects are listed in the alphabetical order on one line, separated by commas and spaces.
- If there are no such items, the line **"Ron, you need to try harder!"** is displayed. The phrase is saved for you in the variable **ron**.

For example:

Input	R
Broom Flights : 10, Defence Against the Dark Arts : 6, Potions : 3, History of Magic : 8 7	B
Broom Flights : 6, Transfiguration : 4 8	R
Transfiguration : 10 10	T

Answer: (penalty regime: 0 %)

Reset answer

```
ron = "Ron, you need to try harder!"

# your cod below
```