Contents

1.Python– First Problem Statement.. 2
1.1 Question.. 2
1.2 Solution.. 3
1.3 Test cases.. 3
2.Python– Second Problem Statement.. 4
2.1 Question.. 4
2.2 Solution.. 7
2.3 Test cases.. 10

Solutions for TCS Xplore iPA held on 25th-JAN-23

1 Simple Python Program

1.1 Problem Statement

Write a python program to remove the suffix "ly" in words of a string ,if present .
The program will take an input from user in the form of a string. It then has to look for words within the string who have the suffix "ly" and remove the suffix .For other words i.e. The ones who do not have the suffix "ly", must be retained as it is. The output will be a string of words without having the suffix “ly”.Any two words in a string is separated by a single space. Considering the above scenario into account, build the logic to print the string having words with no suffix "ly"..
Refer to below instructions and sample input-output for more clarity on the requirement.

Instructions to write main section of the code:

a. You would require to write the main section completely, hence please follow the below instructions for the same.
b. You would require to write the main program which is inline to the "sample input description section" mentioned below and to read the data in the same sequence.

Steps inside main section:

1. First read an input string from the user
2.For each word in the string, if the word contains the suffix "ly", remove it
3. Print the output string i.e.string containing words without having the suffix "ly".

You can use/refer the below given sample input and output to verify your solution.

Sample Input (below) description:

The line of input taken in the main section is a string with words having /not having suffix "ly"

For example :
Input-The patient simply waited patiently in the clinic
Output-The patient simp waited patient in the clinic

1.2 Solution

```
str=input()
suffix="ly"
str=str.split()
str1=""
for i in str:
    if i.endswith(suffix):
        i=i[:-len(suffix)]
        str1=str1+" "+i
    else:
        str1=str1+" "+i
str1=str1.lstrip()
print(str1)
```

1.3 Test Cases

TESTCASE 1

Sample input 1

```
The patient simply waited patiently in the clinic
```

Sample output 1

```
The patient simp waited patient in the clinic
```

TESTCASE 2

Sample input 2

```
The employee was a highly appreciated highly acceptable highly paid but had a highly controversial
past
```

Sample output 2

```
The employee was a high appreciated high acceptable high paid but had a high controversial past
```

TESTCASE 3

Sample input 3

```
I have a simple request
```

Sample output 3

```
I have a simple request
```

TESTCASE 4

Sample input 4

```
Taupin seems more lyrically astute than he has for years
```

Sample output 4

```
Taupin seems more lyrical astute than he has for years
```

2 Complex Python Program

2.1 Problem Statement

Define a class to create objects of type MovieInfo that has the following attributes:
Title-Title should be of the type String; representing title of the movie.Eg: Coda
Year-Year should be of the type Integer;representing the year movie was released. Eg:2021
Awards-Awards should be of the type Integer;representing the number of awards the movie has won.Eg:3
Nomination-Nominations should be of the type Integer;representing the number of nominations the movie was nominated for.Eg:3
Country-Country should be of the type String; representing the movie origin country .Eg:Italy

Define the __init__ method to initialize the attributes of the objects to be created.

Create another class Award with following attributes:
a. List of MovieInfo Objects. Eg: For example if lst is a list containing 2 objects i.e.lst=[m1,m2] where m1 and m2 are 2 different MovieInfo Objects.
Create below functions inside the Award class:

1. The first method takes in year (Type-Integer) and returns details of all non-American movies that have won awards in that year specified.
   Note: All string comparisons should be case insensitive.
2. The second method uses the information from MovieInfo class and creates a dictionary that keeps movies titles as keys and success rate as values. It then sorts the dictionary in descending order based on values. This method returns a sorted dictionary.

For example: Success Rate calculation for the above data:
Number of nominations=3
Number of awards=3
Success rate=(Number of awards/Number of nominations)*100=(3/3)*100=100
Note: Success rate must be an integer value
Instructions to write main section of the code:
a. You would require to write the main section completely, hence please follow the below instructions for the same.
b. You would require to write the main program which is inline to the ""sample input description section"" mentioned below and to read the data in the same sequence.
c. Create the respective objects (MovieInfo, Award)with the given sequence of arguments to fulfill the __init__ method requirement defined in the respective classes referring to the below instructions.
1. First create a list of MovieInfo objects,to do so:
   i. Read a number for the count of MovieInfo objects to be created.

   ii. Create a MovieInfo object after reading the data related to it and add the object to the list of MovieInfo objects. This point repeats for the number of MovieInfo objects (considered in the first line of input) .

2. Create an object of the class Award using the list of MovieInfo.

d. Read the year (Type-Integer).

e. Call the methods described above from the main section with appropriate values as arguments from the main section.

f. Display MovieInfo details that matches the input read. You may refer to the sample output for the display format. If method returns None, then display the message 'No Movie found' (excluding the quotes).

g. Display the sorted dictionary (sorted in descending order based on Values) consisting of movie titles as key and the success rate as values. You may refer to the sample output for the display format. If method returns None, then display the message 'Unable to create dictionary' (excluding the quotes).

Input Format for Custom Testing

a. The 1st input taken in the main section is the number of MovieInfo objects to be added to the list of MovieInfo, say x.

b. The next line of input is the title of the movie.

c. Next input denotes the year the movie was released.

d. The next input denotes the number of Awards won by the movie.

e. Next line of input is the number of nominations the movie received.

f. Next line of input is the country of origin the movie was made.

Point b to f is repeated for x times (where x is the number of MovieInfo objects given in the first line of input).

g. The next line of input refers to the year which is to be passed as argument to the first method.

You can consider below sample input and output to verify your implementation before submission.

Sample Input:

```
8
Coda
2021
3
3
Italy
Drive my car
2021
1
4
Japan
Tenet
2020
1
2
America
```

```
Soul
2020
2
3
America
Mank
2020
2
10
America
Marriage Story
2019
1
6
America
Birdman
2014
4
9
America
Belfast
2021
1
7
Britain
2021
```

Sample Output:

```
Coda
2021
3
3
Italy
Drive my car
2021
1
4
Japan
Belfast
2021
1
7
Britain
```

```
Coda 100
Soul 66
Tenet 50
Birdman 44
Drive my car 25
Mank 20
Marriage Story 16
Belfast 14
```

2.2 Solution

```
class MovieInfo:

    def __init__(self,title,year,awards,nomination,country):

        self.title=title

        self.year=year

        self.awards=awards

        self.nomination=nomination

        self.country=country

class Award:

    def __init__(self,lstAward):

        self.lstAward=lstAward

    def Awardcountry(self,cyear):

        if len(self.lstAward)==0:

            return None

        else:

            lst=[]

            for i in self.lstAward:

                if i.year==cyear:
```

```
                if i.country!='America':
                    lst.append(i)
            return lst
    def dict_Award(self):
        if len(self.lstAward)==0:
            return None
        else:
            d=dict()
            for i in self.lstAward:
                srate=(i.awards/i.nomination)*100
                srate=int(srate)
                d[i.title]=srate
            srted=dict(sorted(d.items(), key=lambda item: item[1],reverse=True))
            return srted
n=int(input())
mlist=[]
for i in range(n):
    title=input()
    year=int(input())
    awards=int(input())
    nomination=int(input())
    country=input()
    mlist.append(MovieInfo(title,year,awards,nomination,country))

obj=Award(mlist)

Awardyear=int(input())

res1=obj.Awardcountry(Awardyear)

if res1:

    for i in res1:

        print(i.title)

        print(i.year)

        print(i.awards)

        print(i.nomination)

        print(i.country)

else:

    print("No Movie found")

res2=obj. dict_Award ()

if res2:

    for k,v in res2.items():

        print(k,v)

else:

    print("Unable to create dictionary")
```

2.3 Test Cases

TESTCASE 1

Sample Input 1

```
8
Coda
2021
3
3
Italy
Drive my car
2021
1
4
Japan
Tenet
2020
1
2
America
Soul
2020
2
3
America
Mank
2020
2
10
America
Marriage Story
2019
1
6
America
Birdman
2014
4
9
America
Belfast
2021
```

```
1
7
Britain
2021
```

Sample Output 1

```
Coda
2021
3
3
Italy
Drive my car
2021
1
4
Japan
Belfast
2021
1
7
Britain
Coda 100
Soul 66
Tenet 50
Birdman 44
Drive my car 25
Mank 20
Marriage Story 16
Belfast 14
```

TESTCASE 2

Sample Input 2

```
8
Coda
2021
3
3
Italy
Drive my car
2021
1
4
Japan
Tenet
```

```
2020
1
2
America
Soul
2020
2
3
America
Mank
2020
2
10
America
Marriage Story
2019
1
6
America
Birdman
2014
4
9
America
Belfast
2021
1
7
Britain
2020
```

Sample Output 2

```
No Movie found
Coda 100
Soul 66
Tenet 50
Birdman 44
Drive my car 25
Mank 20
Marriage Story 16
Belfast 14
```

TESTCASE 3
Sample Input 3

```
8
Coda
2021
3
3
Italy
Drive my car
2021
1
4
Japan
Tenet
2020
1
2
America
Soul
2020
2
3
America
Mank
2020
2
10
America
Marriage Story
2019
1
6
America
Birdman
2014
4
9
America
Belfast
2021
1
7
```

```
Britain
2000
```

Sample output 3

```
No Movie found
Coda 100
Soul 66
Tenet 50
Birdman 44
Drive my car 25
Mank 20
Marriage Story 16
Belfast 14
```

TESTCASE 4

Sample Input 4

```
0
2021
```

Sample Output 4

```
No Movie found
Unable to create dictionary
```