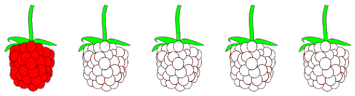
Program: Squad Names.



Your task in this assignment is to write a program that calculates the frequency of names (or substrings) in a provided squad list file. You are provided with a text file that contains the names of all the squad players in the 2022 Fifa World Cup tournament. Your code should prompt the user for a name (or substring), and then print out some statistics that include how many names contain that substring. Below is a sample run with user input in **red**:

```
What file do you want to open? input.txt
the file has 829 names in it
What name (or substring) are you interested in searching for? ad
--------------------------------------------------
2 names start with this string
3 names end with this string
39 names contain this string
--------------------------------------------------
```

It is worth mentioning that your code should recognize when the file name provided does not exist and provide the user with an opportunity to type in the correct file name. Below is a sample run with that functionality demonstrated.

```
What file do you want to open? intput.txt
The file name you specified does not exist.
Enter file name again: tinput.text
The file name you specified does not exist.
Enter file name again: someotherfile.txt
The file name you specified does not exist.
Enter file name again: input.txt
the file has 829 names in it
What name (or substring) are you interested in searching for? pulisic
--------------------------------------------------
0 names start with this string
1 names end with this string
1 names contain this string
--------------------------------------------------
```

To help clarify, here are some specifics and/or constraints:

(1) You are provided with an input file called **`input.txt`** that contains the names you will be parsing. Note that your code should be able to parse any filename that the user provides and not just files named **`input.txt`**

(2) Your program will require at least five user-defined functions. Feel free to add any extra functions you think are necessary.

- A function that reads the contents of the file. This function receives a single argument that represents the filename. It then returns a list containing all the names in the given filename. This function is also in charge of making sure that the filename provided actually exists and if not, provides the user with an opportunity to type in the correct filename. This is a perfect

opportunity to use a **`try except`** construct which you might have seen in previous lessons and assignments.
    - A function that receives two string arguments, and returns a boolean that represents whether the first string argument begins with the second string argument.
    - A function that receives two string arguments, and returns a boolean that represents whether the first string argument ends with the second string argument.
    - A function that receives two string arguments and returns a boolean that represents whether the first string argument contains the second string argument.
    - A function that receives two arguments i.e. a list of names, and a substring. It then creates and returns a short 3 element list that contains how many names in the list contain the substring. The first element contains the number of names that begin with the substring, the second element contains the number of names that end with the substring, and the third element contains the number of names that contain the substring.
(3) The main part of your program should make use of the functions (outlined above) in such a way as to satisfy the problem statement i.e. get the filename, the substring, and print out the appropriate information.
(4) A lot of the functionality that is required in this assignment can easily be implemented by taking advantage of some of python's inbuilt string functions. You are encouraged to look up python string functions to get an understanding of what functionality you have available to you when dealing with strings.
(5) Your output should be **exactly** like the sample runs shown above (of course, actual results will vary depending on the provided input text file and substrings/names);
(6) You must include a meaningful header, use good coding style, use meaningful variable names, and comment your source code where appropriate;
(7) You must use the provided source code template (it contains helpful comments that can help you structure your code); and
(8) You must submit your source code as a single .py file.