Project 10-1: HTML Converter

Create a program that reads an HTML file and converts it to plain text.

Console

```
HTML Converter

Grocery List
* Eggs
* Milk
* Butter
```

Specifications

- Store the following data in a file named groceries.html:

```
<h1>Grocery List</h1>
<ul>
    <li>Eggs</li>
    <li>Milk</li>
    <li>Butter</li>
</ul>
```

- When the program starts, it should read the contents of the file, remove the HTML tags, remove any spaces to the left of the tags, add asterisks (*) before the list items, and display the content and the HTML tags on the console as shown above.

Project 10-2: Email Creator

Create a program that reads a file and creates a series of emails.

Console

```
Email Creator

======================================================================
To:      jbutler@gmail.com
From:    noreply@deals.com
Subject: Deals!

Hi James,

We've got some great deals for you. Check our website!
======================================================================
To:      josephine_darakjy@darakjy.org
From:    noreply@deals.com
Subject: Deals!

Hi Josephine,

We've got some great deals for you. Check our website!
======================================================================
To:      art@venere.org
From:    noreply@deals.com
Subject: Deals!

Hi Art,

We've got some great deals for you. Check our website!
```

Specifications

- Store a list of email addresses in a file using this format:

```
james,butler,jbutler@gmail.com
josephine,darakjy,josephine_darakjy@darakjy.org
art,venere,art@venere.org
```

- Store a template for a mass email in a file like this:

```
To:      {email}
From:    noreply@deals.com
Subject: Deals!

Hi {first_name},

We've got some great deals for you. Check our website!
```

- When the program starts, it should read the email addresses and first names from the file, merge them into the mass email template, and display the results on the console.
- All email addresses should be converted to lowercase.
- All first names should be converted to title case.
- If you add names to the list of email addresses, the program should create more emails.
- If you modify the template, the program should change the content of the email that's created.

Project 10-4: Pig Latin Translator

Create a program that translates English to Pig Latin.

Console

```
Pig Latin Translator

Enter text: Tis but a scratch.
English:    tis but a scratch
Pig Latin:  istay utbay away atchscray

Continue? (y/n): y

Enter text: We are the knights who say nee!
English:    we are the knights who say nee
Pig Latin:  eway areway ethay ightsknay owhay aysay eenay

Continue? (y/n): n

Bye!
```

Specifications

- Convert the English to lowercase before translating.
- Remove any punctuation characters before translating.
- Assume that words are separated from each other by a single space.
- If the word starts with a vowel, just add *way* to the end of the word.
- If the word starts with a consonant, move all of the consonants that appear before the first vowel to the end of the word, then add *ay* to the end of the word.
- If a word starts with the letter *y*, the *y* should be treated as a consonant. If the *y* appears anywhere else in the word, it should be treated as a vowel.

Note

- There are no official rules for Pig Latin. Most people agree on how words that begin with consonants are translated, but there are many different ways to handle words that begin with vowels.

Project 12-1: Game Stats

Create a program that allows you to view the statistics for a player of a game.

Console

```
Game Stats program

ALL PLAYERS:
Elizabeth
Joel
Mike

Enter a player name: elizabeth
Wins:   41
Losses: 3
Ties:   22

Continue? (y/n): y

Enter a player name: john
There is no player named John.

Continue? (y/n): y

Enter a player name: joel
Wins:   32
Losses: 14
Ties:   17

Continue? (y/n): y

Enter a player name: mike
Wins:   8
Losses: 19
Ties:   11

Continue? (y/n): n

Bye!
```

Specifications

- The program should use a dictionary of dictionaries to store the stats (wins, losses, and ties) for each player. You can code this dictionary of dictionaries at the beginning of the program using any names and statistics that you want. Make sure to provide stats for at least three players.
- The program should begin by displaying an alphabetical list of the names of the players.
- The program should allow the user to view the stats for the specified player.

Project 12-3: Champion Counter

Create a program that reads a text file that contains a list of FIFA World Cup champions and determines the country that has won the most championships.

Console

```
FIFA World Cup Winners

Country         Wins  Years
=======         ====  =====
Argentina       2     1978, 1986
Brazil          5     1958, 1962, 1970, 1994, 2002
England         1     1966
France          1     1998
Germany         4     1954, 1974, 1990, 2014
Italy           4     1934, 1938, 1982, 2006
Spain           1     2010
Uruguay         2     1930, 1950
```

Specifications

- Your instructor should provide a text file named world_cup_champions.txt that contains data like this:

```
Year,Country,Coach,Captain
1930,Uruguay,Alberto Suppici,José Nasazzi
1934,Italy,Vittorio Pozzo,Gianpiero Combi
1938,Italy,Vittorio Pozzo,Giuseppe Meazza
...
...
2002,Brazil,Luiz Felipe Scolari,Cafu
2006,Italy,Marcello Lippi,Fabio Cannavaro
2010,Spain,Vicente del Bosque,Iker Casillas
2014,Germany,Joachim Löw,Philipp Lahm
```

- When the program starts, it should read the text file and use a dictionary to store the required data using the name of each country that has won the World Cup as the key.
- The program should compile the data shown above and display the countries alphabetically.

Project 12-4: Monthly Sales

Create a program that allows you to view and edit the sales amounts for each month of the current year.

Console

```
Monthly Sales program

COMMAND MENU
view   - View sales for specified month
edit   - Edit sales for specified month
totals - View sales summary for year
exit   - Exit program

Command: view
Three-letter Month: jan
Sales amount for Jan is 14,317.00.

Command: edit
Three-letter Month: jan
Sales Amount: 15293
Sales amount for Jan is 15,293.00.

Command: totals
Yearly total:       67,855.00
Monthly average:     5,654.58

Command: view
Three-letter Month: july
Invalid three-letter month.

Command: exit
Bye!
```

Specifications

- Your instructor should provide a text file named monthly_sales.txt that consists of rows that contain three-letter abbreviations for the month and the monthly sales.
- The program should read the file and store the sales data for each month in a dictionary with the month abbreviation as the key for each item.
- Whenever the sales data is edited, the program should write the changed data to the text file.