Project 7-1: Pig Dice Rules

Create a program that reads a list of rules from a file and displays them.

Console

```
Pig Dice Rules:
* See how many turns it takes you to get to 20.
* Turn ends when player rolls a 1 or chooses to hold.
* If you roll a 1, you lose all points earned during the turn.
* If you hold, you save all points earned during the turn.
```

Specifications

- Your instructor should provide a text file named rules.txt.
- Your program should read the text file and display it on the console.

Project 7-2: Wizard Inventory

Create a program that keeps track of the items that a wizard is carrying.

Console

```
The Wizard Inventory program

COMMAND MENU
walk - Walk down the path
show - Show all items
drop - Drop an item
exit - Exit program

Command: walk
While walking down a path, you see a scroll of uncursing.
Do you want to grab it? (y/n): y
You picked up a scroll of uncursing.

Command: walk
While walking down a path, you see an unknown potion.
Do you want to grab it? (y/n): y
You can't carry any more items. Drop something first.

Command: show
1. a wooden staff
2. a scroll of invisibility
3. a crossbow
4. a scroll of uncursing

Command: drop
Number: 3
You dropped a crossbow.

Command: exit
Bye!
```

Specifications

- Your instructor should provide a text file named wizard_all_items.txt that contains a list of all the items that a wizard can carry.
- You should create another text file named wizard_inventory.txt to store the current items that the wizard is carrying.
- A wizard can only carry four items at a time, and those four items must be different.
- When the user selects the walk command, the program should read the items from the file, create a list of the items that aren't already in the wizard's inventory, randomly pick one of those items, and give the user the option to grab it. To create a list of the items that aren't already in the wizard's inventory, you can use a list comprehension as described in chapter 6.
- Make sure to update the inventory text file you created every time the user grabs or drops an item.
- For the drop command, display an error message if the user enters an invalid number for the item.

Project 7-3: Contact Manager

Create a program that a user can use to manage the primary email address and phone number for a contact.

Console

```
Contact Manager

COMMAND MENU
list - Display all contacts
view - View a contact
add  - Add a contact
del  - Delete a contact
exit - Exit program

Command: list
1. Guido van Rossum
2. Eric Idle

Command: view
Number: 2
Name: Eric Idle
Email: eric@ericidle.com
Phone: +44 20 7946 0958

Command: add
Name: Mike Murach
Email: mike@murach.com
Phone: 559-123-4567
Mike Murach was added.

Command: list
1. Guido van Rossum
2. Eric Idle
3. Mike Murach

Command: exit
Bye!
```

Specifications

- Your instructor should provide a CSV file named contacts.csv.
- When the program starts, it should read the contacts from the CSV file.
- For the view and del commands, display an error message if the user enters an invalid contact number.
- When you add or delete a contact, the change should be saved to the CSV file immediately. That way, no changes are lost, even if the program crashes later.

Project 7-4: Monthly Sales

Create a program that reads the sales for 12 months from a file and calculates the total yearly sales as well as the average monthly sales. In addition, this program should let the user edit the sales for any month.

Console

```
Monthly Sales program

COMMAND MENU
monthly - View monthly sales
yearly  - View yearly summary
edit    - Edit sales for a month
exit    - Exit program

Command: monthly
Jan - 14317
Feb - 3903
Mar - 1073
Apr - 3463
May - 2429
Jun - 4324
Jul - 9762
Aug - 15578
Sep - 2437
Oct - 6735
Nov - 88
Dec - 2497

Command: yearly
Yearly total:     66606
Monthly average:  5550.5

Command: edit
Three-letter Month: Nov
Sales Amount: 8854
Sales amount for Nov was modified.

Command: exit
Bye!
```

Specifications

- Your instructor should provide a CSV file named monthly_sales.csv that contains the month and sales data shown above.
- For the edit command, display an error message if the user doesn't enter a valid three-letter abbreviation for the month.
- When the user edits the sales amount for a month, the data should be saved to the CSV file immediately. That way, no data is lost, even if the program crashes later.
- Round the results of the monthly average to a maximum of 2 decimal digits.

Project 7-5: Email List Cleaner

Create a program that reads a CSV file that contains a list of prospects for an email list, reformats the data, and writes the cleaned list to another file.

Console

```
Welcome to the Email List Cleaner

Source list:  prospects.csv
Cleaned list: prospects_clean.csv

Congratulations! Your list has been cleaned!
```

The prospect.csv file

```
FIRST_NAME,LAST_NAME,EMAIL
james,butler,jbutler@gmail.com
Josephine  ,Darakjy,josephine_darakjy@darakjy.org
ART,VENERE,ART@VENERE.ORG
...
```

The prospect_clean.csv file

```
First_Name,Last_Name,email
James,Butler,jbutler@gmail.com
Josephine,Darakjy,josephine_darakjy@darakjy.org
Art,Venere,art@venere.org
...
```

Specifications

- Your instructor should provide a CSV file named prospects.csv that contains a list of prospects.
- Your program should fix the formatting problems and write a file named prospects_clean.csv.
- All names should use title case. To convert a string to title case, you can call the title() method from the string.
- All email addresses should use lowercase. To convert a string to lowercase, you can call the lower() method from the string.
- All extra spaces at the start or end of a string should be removed. To do that, you can call the strip() method from the string.