Junior Knights Week #8 Python Homework: Lists

Part A: Movie Cast (cast.py)

Pick your favorite movie and create a program that forms a list that stores each major actor/actress in the movie. Then, ask the user for a name of an actor/actress to search for. If the person is in your movie, print out a message indicating that their actor/actress is in the movie. Otherwise, print out a message stating that their actor actress isn't in the movie. After this, print out the whole cast of the movie.

Sample Run #1

```
Which actor/actress do you want to search for?
Tom Cruise
Sorry, Tom Cruise is NOT in Star Wars.
```

Sample Run #2

```
Which actor/actress do you want to search for?
Harrison Ford
Yes, Harrison Ford is in Star Wars!
```

Part B: Movie list Edit (movies2.py)

Edit the program movies.py, shown in class to include a menu option that allows the user to search for a particular person. The edited program should allow the user this option (as option #3) and if the user chooses this option, prompt them to enter a person for which to search. Your program should then print out an appropriate message indicating whether or not that person is currently in the list of people to go see the movie.

Sample Run

```
Please select one of the following menu options.
1. Add a person to the movie list.
2. Delete a person from the movie list.
3. Search for a person in the movie list.
4. Print out all people currently going to the movie.
5. Close the list.
1
Who should we add to the list?
John

Please select one of the following menu options.
1. Add a person to the movie list.
2. Delete a person from the movie list.
3. Search for a person in the movie list.
4. Print out all people currently going to the movie.
5. Close the list.
1
Who should we add to the list?
Sarah
```

```
Please select one of the following menu options.
1. Add a person to the movie list.
2. Delete a person from the movie list.
3. Search for a person in the movie list.
4. Print out all people currently going to the movie.
5. Close the list.
3
Who do you want to check to see is in the list?
Jimmy
Sorry, Jimmy is NOT on the movie list right now.

Please select one of the following menu options.
1. Add a person to the movie list.
2. Delete a person from the movie list.
3. Search for a person in the movie list.
4. Print out all people currently going to the movie.
5. Close the list.
3
Who do you want to check to see is in the list?
Sarah
Great, Sarah is going to the movie so far.

Please select one of the following menu options.
1. Add a person to the movie list.
2. Delete a person from the movie list.
3. Search for a person in the movie list.
4. Print out all people currently going to the movie.
5. Close the list.
5

Here is the final list of people going to the movie:
['John', 'Sarah']
```

Part C: Sort Students (sort.py)

Write a program that prompts the user to enter students’ names and then sorts them in alphabetical order and prints out the sorted list. Follow the sample run below.

Sample Run

```
How many students are in your class?
3
Please enter their names, one per line.
Tinisha
Caroline
Raymond
Here is your class sorted:
[‘Caroline’, ‘Raymond’, ‘Tinisha’]
```

Part D: Parentheses Edit (parens2.py)

Some expressions have BOTH nested sets of parentheses AND curly braces. Edit the parentheses program so that it takes one of these expressions in as input and prints out whether or not the parentheses and curly braces are well-formed.

Here is an example of a well-formed set of parentheses and curly braces:

```
{ () () ({ ( () ) } { } () ) }
```

Here is an example that is NOT well-formed:

```
{ () () ({ ( () ) } { } () } )
```

This example is NOT well-formed because when you attempt to match the second to last symbol, a close brace, the top of the stack stores an open parenthesis, which is NOT a match. Thus, even if the number of open symbols and close symbols is equal AND their relative ordering is okay, the expression may NOT be well-formed. In particular, if the closing symbol isn't the same type as the corresponding opening symbol, the expression is not well-formed even if everything else matches.

Note that our algorithm changes as follows:

When reaching a close parenthesis OR close brace, when we consult the stack, we MUST pop off the correct corresponding open symbol. For example, if we are processing a close brace, then the item at the top of the stack must be an open brace. If it's an open parenthesis, the expression is NOT well-formed.

Sample Run

```
Please enter your set of parentheses.
{ () () ({ ( () ) } { } () ) }
Your string is a valid set of parens.
```

Sample Run

```
Please enter your set of parentheses.
{ () () ({ ( () ) } { } () } }
Sorry, your parentheses don't match.
```