

CS 1301 Homework 05

Due: Thursday October 8th, before 11:55pm

This can be a pair programming assignment!

You are allowed to work with a student in your assigned recitation, but you are **both** responsible for submitting the **exact same** code to T-Square. Follow good pair-programming practices by working together at a single computer and switching the driver/navigator role frequently.

Your pair may collaborate with other students in this class. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. You **should not** exchange code or write code for other pairs. Collaboration at a reasonable level will not result in substantially similar code.

For pair programming assignments, you and your partner **should** turn in identical assignments.

Feel free to work on this assignment alone, as long as you have the appropriate collaboration statement.

Files to submit: hw5.py

For help:

- TA Helpdesk
- Email TAs
- Piazza

Notes:

- **Don't forget to include the required comments and collaboration statement (as outlined on the course syllabus)**
- **Do not wait until the last minute** to do this assignment in case you run into problems.
- If you find a significant error in the homework assignment, please let a TA know immediately.

Part I - List Manipulation

For this part of the assignment, your group will be creating three functions regarding list manipulation.

- (5pts) Define a function called *onlySixths* that takes in a single parameter: a list of integers. The function should return a new list containing all of the numbers that are evenly divisible by six.

Examples:

```
onlySixths([1,2,3,4]) should return [ ]
```

```
onlySixths([12,37,42,66,72]) should return [12,42,66,72]
```

- (10pts) Define a function called *union* that takes in two lists that contain numbers - and returns the union of the two lists. The function should not change either of the input lists. The elements in the returned list must be in ascending order, and there should be no repeats in the returned list. Note: Numbers includes both floats and integers, although our example below only shows integers.

(Hint: The `sort()` function may help)

Examples:

```
>>> a = [1,3,2]
```

```
>>> b = [4,2,5]
```

```
>>> c = union(a,b)
```

```
>>> print(c)
```

```
[1,2,3,4,5]
```

- (15pts) Define a function called *multiplyNums* that takes in a list and returns the product of all of the numbers from that list. The list may contain elements which are neither integers nor floats. If there is a nested list within any list, multiply all of the numbers in the nested list by your overall product as well. Your returned value can be either a float or an integer depending upon if the input list had floats.

(Hint: Utilize conditionals and iteration to figure out which list items to multiply, and recursion may help you process nested lists.)

Examples:

```
>>> a = ["Goku", 1, "Vegeta", 7, [1,2,3]]
```

```
>>> b = multiplyNums(a)
```

```
>>> print(b)
```

```
42
```

Part II - String Manipulation

For this part of the assignment, your group will be creating two functions regarding string manipulation.

- (10pts) Write a function called *abbreviator* that takes in one string as a parameter. The function should return a new string. The returned string should contain only the capital letters and numbers from the original string. Make sure that your returned string does not contain any special characters such as periods, semicolons, apostrophe's, spaces, etc. (Hint: the `isalpha()` and `isupper()` functions may help)

Examples:

```
>>> a = "If You're Reading This It's Too Late"
>>> b = abbreviator(a)
>>> print(b)
IYRTITL
```

```
>>> c = "Super Saiyan 2 Gohan"
>>> d = abbreviator(c)
>>> print(d)
SS2G
```

- (15pts) Write a function called *parse* that takes in two parameters: a string and a delimiter. Your function should return a list of separated strings depending on the delimiter. You may assume the delimiter will be a single character. Each string in the list should not contain the delimiter itself. You **may not** use any built-in Python functions such as the `split` function, instead, use iteration and conditionals.

Examples:

```
parse("This is a Sentence", " ") should return ['This', 'is', 'a', 'Sentence']
parse("f-r-i-e-n-d-s", "-") should return ['f', 'r', 'i', 'e', 'n', 'd', 's']
parse("f,.5,.5", ",") should return ['f', '.5', '.5']
```

Part III - Robot Manipulation

For this part of the assignment, your group will be writing one function utilizing the robot.

- (20 pts) Write a function called `lightStats` that takes in no parameters. While Use the `getLight()` function to get a list containing the left, right, and center values. Print out all three values as a statement in the following format:
Left: <left value>
Center: <center value>
Right: <right value>

Return a new list with the mean, median, and range (in that order) of the three values you collected above. Note that the median will be one of the three values, and the range is the difference between the largest and smallest value.

(Hint: `getLight()` returns a list of different values depending on your robot's position, so the output should change as you move around.)

Examples:

```
>>> stats1 = lightStats()
Left: 2891
Center: 3500
Right: 868
>>> print(stats1)
[2419.67, 2891, 2632]
```

```
>>> stats2 = lightStats()
Left: 1710
Center: 3031
Right: 2817
>>> print(stats2)
[2519.33, 2817, 1321]
```

Rubric

| | |
|--|-----------|
| onlySixths | 5 |
| Returns an empty list when input has zero multiples of six | 2 |
| Returns a list with numbers of only multiples of six | 3 |
| union | 10 |
| The original two lists have remained the same | 3 |
| Numbers are in ascending order in the returned list | 3 |
| No repeated numbers in the returned list | 4 |
| multiplyNums | 15 |
| Returns any float or integer | 3 |
| Returns the correct value of the input | 12 |
| abbreviator | 10 |
| Returned string contains uppercase alpha characters | 2 |
| String contains numbers | 4 |
| String contains zero special characters | 4 |
| parse | 15 |
| Returns a list | 2 |
| Returns a list of separate strings | 5 |
| Separates the strings correctly by the delimiter | 8 |
| lightStats | 20 |
| Prints out the statement in the correct format | 4 |
| Returns a list of three values | 2 |
| Mean value is calculated correctly | 4 |
| Median value is calculated correctly | 5 |
| Range value is calculated correctly | 5 |