

Lab 6

This lab provides additional exercises on iteration over lists.

Download lab6 files , place it in your `cpe101` directory.

Fold Pattern

Develop these functions in the `fold` directory in files `fold.py` and `fold_tests.py`.

Each of the functions for this part of the lab is to be implemented using the fold pattern. In this pattern, the values in the input list are combined in some manner (e.g., an arithmetic or relational operation) to compute a single result value. Each of these functions will use a local variable (or local variables) to hold the computed value as the list is traversed. This variable is often updated at each step of the iteration.

`sum`

The `sum` function returns the sum of all values in the input list.

`index_of_smallest`

The `index_of_smallest` function returns the index of the smallest value in the input list. If the list is empty (i.e., the length is ≤ 0), then the function returns -1.

`nearest_origin`

The `nearest_origin` function returns the index of the point nearest the origin. If the list is empty, then the function returns -1.

Grouping

For this part of the lab you will develop a function that will be of use in the last assignment of the quarter. In `groups.py`, define the following function.

`groups_of_3`

The `groups_of_3` function takes a list of values as its only argument. This function will group the elements of the input list into groups of three (i.e., the first three consecutive values form a group, the second three form a group, etc.). This function must return a list of lists where each sublist (excluding, perhaps, the last) is a group of three values. If there are not enough values to fill the last grouping, then it will contain fewer than three values.

As examples, consider the following.

```
>>> groups_of_3([1,2,3,4,5,6,7,8,9])
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
>>> groups_of_3([1,2,3,4,5,6,7,8])
[[1, 2, 3], [4, 5, 6], [7, 8]]
```

While

The `while` statement is another way to specify iteration, though it is typically used for a different type of iteration than that over lists.

For this part of the lab, edit the `loops.py` file in the `loops` directory. Complete the `while_version` function by translating the `for` statement from the `for_version` function into an equivalent `while` statement. Think carefully about the different steps in the `for` statement.

Demonstration

Demonstrate the test cases from each part of the lab to your instructor to have this lab recorded as completed. In addition, be prepared to show the source code to your instructor.