# CSE 160 Spring 2025 Practice Quiz 2 Key

1. You are given the following function definitions. Answer the questions below based on what the functions do.

```
def my_func1(x):
    return x % 2 == 0

def my_func2(x, y):
    if x > y:
        return x - y
    else:
        return y - x

def my_func3(word):
    return word[::-1]
```

a. What will the function call `my_func1(7)` return?

False

b. What about `my_func1(10)`?

True

c. What will `my_func2(3, 10)` return?

7

d. What about `my_func2(8, 2)`?

6

e. What does `my_func3("banana")` return? Explain what this function is doing in one sentence.

“ananab”
The function my_func3 returns its input reversed.

2. You are given a spreadsheet about grocery stores that looks like the following:

```
Target cheap groceries home clothing
Trader_Joe's medium groceries
Whole_Foods expensive groceries
Goodwill cheap home clothing
```

There are more lines that are not shown. You can assume that all lines follow the same format (`store_name price item1 item2 item3 . . .`); that is, every line has at least store_name, price, and *at least* one item, with no limit on the maximum number of items.

Write a function called `find_store()` that

- Takes in a filename as a parameter
- Opens and reads the file line by line
- Finds all of the *cheap* stores that sells *home* goods
- Returns the names of the stores as a list

```
def find_store(stores_file):
      matching_stores = []
      opened_file = open(stores_file)
      for line in opened_file:
            parts = line.split(" ")
            # parts becomes a list of strings where each word (separated
            # by " ") is one element in the list
            store_name = parts[0]
            price = parts[1]
            items = parts[2:]
            if price == "cheap" and "home" in items:
                  matching_stores.append(store_name)
      opened_file.close()
      return matching_stores
```

Note that this exact style of problem will not be on the actual quiz since we weren't able to cover file i/o thoroughly in class.

3. You are given a nested list representing groups of test scores:

```
scores = [[70, 80], [95], [], [100, 85, 90]]
```

a. Calculate the average score for all scores across all groups. Your code should print a single number representing the average.

```
all_scores_total = 0
num_scores = 0
for group in scores:
      num_scores += len(group)
      for score in group:
            all_scores_total += score
print(all_scores_total / num_scores)
```

b. Modify the original scores list so that any inner list with a total score less than 160 is replaced by the string `"Needs Improvement"`.

```
for i in range(len(scores)):
      inner_list = scores[i]
      scores_total = 0
      for score in inner_list:
            scores_total += score
      if scores_total < 160:
            scores[i] = "Needs Improvement"
```