fullName:________________________ andrewID:_________________ section:___

15-112 N25
Quiz5 Part 2 of 2

This is the paper version of **Part 2 of Quiz5**. You will receive this paper version only if requested (or if taking the quiz with the testing center). You **must write your name on this paper and hand this back** in immediately after the assessment, even if you do not use it to write your answers. If we do not receive it immediately, you will receive a zero on the assessment.

Quiz5 Part 2 is located in section 8.12 of CS Academy (the end of Unit 8), and contains only the FRs. If you cannot see the submit button, call over a TA or Mike and let us know your andrewID.

We will not grade anything you write on these pages unless you initial the box below:

________ **Write your initials here if and only if you wish for us to grade your FRs on this paper *instead of* any answers submitted in CS Academy.** Leave the space blank if you wish for us to grade what is written in CS Academy (the default, recommended option).

During the quiz, as always, you may not view any other notes, prior work, websites or resources, including any form of AI. You may not communicate with anyone else except for TAs or faculty during the assessment. All syllabus policies apply.

Some students will take this at a different time with testing accommodations. As such, you may not discuss this test with anyone else, even briefly, in any form, until we have released grades. Failure to abide by these rules may result in an academic integrity violation.

Do not use any concepts not taught in lecture or CS Academy. Also, read the restrictions on FR1 carefully; you may not use loops on that problem.

Do not open this or look inside (even briefly) before we instruct you to begin. Close it once you are done.

Part 2[80pts total]: Free Response

Your functions should work generally for the kinds of inputs specified in the problem statement, and we may test your code using additional test cases. We will manually grade both of these problems for partial credit if you do not pass all the test cases.

FR1[35pts]: gapSum(n)

Write the recursive function **gapSum(n)** that takes a (possibly negative) integer n and returns its "gap sum" (a made-up term). The gap sum is found by finding the absolute difference between each neighboring pair of digits and adding those together.
For example:
`assert(gapSum(564) == 3)`
Because `abs(5-6) + abs(6-4) == 1 + 2 == 3`

Likewise:
`assert(gapSum(-47325) == 11)`
Because `abs(4-7) + abs(7-3) + abs(3-2) + abs(2-5) == 3 + 4 + 1 + 3 == 11`

See the test cases for more examples.

You must solve this function recursively in order to receive any points. You may not use loops. Also, you may not use strings, lists, tuples, sets, or dictionaries.

```
    assert(gapSum(564) == 3)
    assert(gapSum(-47325) == 11)
    assert(gapSum(1246) == 5)
    assert(gapSum(-1246) == 5)
    assert(gapSum(5555) == 0)
    assert(gapSum(5050) == 15)
    assert(gapSum(0) == 0)
    assert(gapSum(9) == 0)
    assert(gapSum(-6) == 0)
```

Write your answer on the next page

```
#Write your answer here
```

FR2[45 pts]: Book and Chapter classes

In this exercise, you will write the Chapter and Book classes so that the following test cases pass. You may not hard-code any of the test cases in your solution.

Do not hardcode the test cases, but you may assume that the parameters are always legal (so, for example, chapter indexes are always in bounds). You must use OOP properly. Do not add any unneccessary methods or inheritance to either class.

With that, here are the test cases:

```
chapterA = Chapter('I love CS!', 30) # chapter title, # of pages
assert(chapterA.getTitle() == 'I love CS!')
chapterB = Chapter('So do I!', 15)
book1 = Book('CS is Fun!', [chapterA, chapterB]) # book title, chapters
book2 = Book('The Short Book', [ Chapter('Quick Read!', 5) ])
assert(book1.getChapterCount() == 2)
assert(book1.getPageCount() == 45)
assert(book2.getChapterCount() == 1)
assert(book2.getPageCount() == 5)
assert(book1.getChapter(0).getTitle() == 'I love CS!')
assert(book1.getChapter(1).getTitle() == 'So do I!')
assert(book2.getChapter(0).getTitle() == 'Quick Read!')

# Move chapter 0 from book1 to the end of book2
# so moveChapter always moves to the end of the target book.
book1.moveChapter(0, book2)
assert(book1.getChapterCount() == 1)
assert(book1.getPageCount() == 15)
assert(book1.getChapter(0).getTitle() == 'So do I!')
assert(book2.getChapterCount() == 2)
assert(book2.getPageCount() == 35)
assert(book2.getChapter(0).getTitle() == 'Quick Read!')
assert(book2.getChapter(1).getTitle() == 'I love CS!')
```

Write your answer on the following page

```
# Write your answer here
```

# You may continue your answer here if you wish