CPS 2231 2025 Spring
Lab 4: Recursion

Instructor: Dr. Y. Tiffany Tang[1] and Dr. Pinata Winoto[2]
Songlin Shang[3], Cong Ye[4] and Zike Deng[5]

Instruction:

Self-check: In all labs in the future, you can download `check.py` in canvas, which will help you check the output format in labs, to make sure you don't have compile error or logic error before you submit your file. If you didn't finish installing the python environment/JDK installation, you can download the file from canvas named as tutorial.pdf. Please read and follow the tutorial, which teaches you how to install a python environment, for checking your code result. Also, the instructions of all questions also teach you how to use `check.py` to check your code.

Plagiarism: We encourage discussing all labs with your classmate, but **NEVER** copy other's code directly. We will use GradeScope@ platform, to check the similarity probability, 90% similarity reported by GradeScope will be reviewed by TAs, and will report to your instructor and receive some penalty based on the event.

AI-Generate code issue: We encourage you to use AI tools, to help you compile all labs, **BUT NEVER** directly **copy and paste**. To detect that, we will use different tool to detect AI possibility include:
ChatGPT Code Detector:https://chatgpt.com/c/67bdddec-c728-8010-b1b6-abf3e56b3136
GPTSniffer:https://github.com/MDEGroup/GPTSniffer
Penalty will be given for high AI probability.

Grading: Each lab has the same points, and each test case has the same points. In check.py, you might check your code with given test cases in labs to avoid spelling errors and compile errors. We still have invisible test cases, which test for grading. This might test you to consider all possible positions. All test cases have the same points.

[1] Tiffany Ya Tang, Associate Professor, Department of Computer Science, Wenzhou-Kean University; Email: yatang@kean.edu
[2] Pinata Winoto, Assistant Professor, Department of Computer Science, Wenzhou-Kean University; Email: pwinoto@kean.edu
[3] Songlin Shang, Department of Computer Science, Wenzhou-Kean University; Email:shangs@kean.edu
[4] Cong Ye, Department of Computer Science, Wenzhou-Kean University; Email:yecon@kean.edu
[5] Zike Deng, Department of Computer Science, School of Computing and Data Science, University of Hong Kong; Email:baylordeng@connect.hku.hk

Get Started

Please download Lab4_ 1.java, Lab4_2.java on canvas.

Before you start the lab, you can create a project on eclipse called Lab 4. Also, we suggest you download the check.py in canvas and move it under the same folder with all labs.

Before you start coding, please direct you terminal to same folder using following code: `cd path` (`path is the way to the folder of check.py`) here is an example:

Now you can start coding after all these preparation.

Lab4_1
Problem: Count Character Occurrences

Write a recursive method countChar that takes a string str and a character ch as input and returns the number of times the character ch appears in the string. You may assume the input string is not null (but it may be empty). You must use recursion to solve the problem and cannot use loops.

Requirements:

- Method signature: public static int countChar(String str, char ch)
- Solve the problem recursively by considering the first character and the rest of the string.
- **Do not use for or while loops.**

Example1:

Case1: countChar("banana", 'a')
returns 3 ('a' appears 3 times in "banana")

Example2:

Case2: countChar("hello", 'l')
returns 2 ('l' appears 2 times in "hello")

Example3:

Case3:countChar("xyz", 'w')
returns 0 ('w' does not appear in "xyz")

Example4:

Case4:countChar("", 'a')
returns 0 (empty string contains no characters)

Hint:

- Base case: If the string is empty, return 0.
- Recursive case: Check if the first character matches the target character, then recursively process the rest of the string and combine the results.

Self-Check:

When you finish coding, you can run following code: `python check.py --lab Lab4_1`

If everything good, you might see following content:

```
(base) matsumatsu@songprodeMacBook-Pro Lab4 % python check.py --lab Lab4_1
********* Compiling Java file: Lab4_1.java *********
✅ Compilation succeeded!
Running test case: countChar("banana", 'a') should return 3
********* Running Java program: Lab4_1 *********
✅ Execution succeeded!
✅ Output is correct: 3
Running test case: countChar("hello", 'l') should return 2
********* Running Java program: Lab4_1 *********
✅ Execution succeeded!
✅ Output is correct: 2
Running test case: countChar("xyz", 'w') should return 0
********* Running Java program: Lab4_1 *********
✅ Execution succeeded!
✅ Output is correct: 0
Running test case: countChar("", 'a') should return 0
********* Running Java program: Lab4_1 *********
✅ Execution succeeded!
✅ Output is correct: 0
Running test case: countChar("a", 'a') should return 1
********* Running Java program: Lab4_1 *********
✅ Execution succeeded!
✅ Output is correct: 1
Running test case: countChar("aa", 'a') should return 2
********* Running Java program: Lab4_1 *********
✅ Execution succeeded!
✅ Output is correct: 2
Running test case: countChar("abc", 'b') should return 1
********* Running Java program: Lab4_1 *********
✅ Execution succeeded!
✅ Output is correct: 1
Running test case: countChar("abcabc", 'c') should return 2
********* Running Java program: Lab4_1 *********
✅ Execution succeeded!

✅ Output is correct: 2
Running test case: countChar("ABC", 'a') should return 0
********* Running Java program: Lab4_1 *********
✅ Execution succeeded!
✅ Output is correct: 0
Running test case: countChar("Aa", 'a') should return 1
********* Running Java program: Lab4_1 *********
✅ Execution succeeded!
✅ Output is correct: 1
✅ All checks passed for the specified lab!
```

Lab4_2
Problem: Count Palindromic Substrings

Write a recursive method countPalindromicSubstrings that takes a string as input and returns the number of palindromic substrings in the string. A palindromic substring is a contiguous sequence of characters that reads the same forward and backward (e.g., "aba", "rr", "a"). You may assume the input string is not null, but it may be empty. Do not use loops; use recursion to solve this problem.

Requirements:

- Method signature: public static int countPalindromicSubstrings(String str)
- Solve the problem recursively by considering substrings starting from the beginning of the string.
- Do not use for or while loops.

Example1:

Case1: countPalindromicSubstrings("aba")
returns 4 (palindromes: "a", "b", "a", "aba")

Example2:

Case2: countPalindromicSubstrings("aaa")
returns 6 (palindromes: "a", "a", "a", "aa", "aa", "aaa")

Example3:

Case3: countPalindromicSubstrings("xy")
returns 2 (palindromes: "x", "y")

Example4:

Case4: countPalindromicSubstrings("")
returns 0 (empty string has no substrings)

Hint:

- Base case: If the string is empty, return 0.
- Recursive case: Count all palindromic substrings starting at the first character, then recursively count palindromic substrings in the rest of the string (excluding the first character).
- Use a helper method to check if a substring is palindromic by comparing the first and last characters and recursing on the inner substring.

Self-Check:

When you finish coding, you can run following code: `python check.py --lab Lab4_2`

If everything good, you might see following content:

```
(base) matsumatsu@songprodeMacBook-Pro Lab4 % python check.py --lab Lab4_2
********* Compiling Java file: Lab4_2.java *********
✅ Compilation succeeded!
Running test case: countPalindromicSubstrings("aba") should return 4
(palindromes: "a", "b", "a", "aba")
********* Running Java program: Lab4_2 *********
✅ Execution succeeded!
✅ Output is correct: 4
Running test case: countPalindromicSubstrings("aaa") should return 6
(palindromes: "a", "a", "a", "aa", "aa", "aaa")
********* Running Java program: Lab4_2 *********
✅ Execution succeeded!
✅ Output is correct: 6
Running test case: countPalindromicSubstrings("xy") should return 2
(palindromes: "x", "y")
********* Running Java program: Lab4_2 *********
✅ Execution succeeded!
✅ Output is correct: 2
Running test case: countPalindromicSubstrings("") should return 0 (empty string
has no substrings)
********* Running Java program: Lab4_2 *********
✅ Execution succeeded!
✅ Output is correct: 0
Running test case: countPalindromicSubstrings("a") should return 1 (palindrome:
"a")
********* Running Java program: Lab4_2 *********
✅ Execution succeeded!
✅ Output is correct: 1
Running test case: countPalindromicSubstrings("aa") should return 3
(palindromes: "a", "a", "aa")
********* Running Java program: Lab4_2 *********
✅ Execution succeeded!
✅ Output is correct: 3
Running test case: countPalindromicSubstrings("abc") should return 3
(palindromes: "a", "b", "c")
********* Running Java program: Lab4_2 *********
✅ Execution succeeded!

✅ Output is correct: 3
Running test case: countPalindromicSubstrings("abba") should return 6
(palindromes: "a", "b", "b", "a", "bb", "abba")
********* Running Java program: Lab4_2 *********
✅ Execution succeeded!
✅ Output is correct: 6
Running test case: countPalindromicSubstrings("racecar") should return 10
(palindromes: "r", "a", "c", "e", "c", "a", "r", "cec", "aceca", "racecar")
********* Running Java program: Lab4_2 *********
✅ Execution succeeded!
✅ Output is correct: 10
Running test case: countPalindromicSubstrings("abcd") should return 4
(palindromes: "a", "b", "c", "d")
********* Running Java program: Lab4_2 *********
✅ Execution succeeded!
✅ Output is correct: 4
✅ All checks passed for the specified lab!
```

Last Step

Congratulations! You have finished all Labs in Lab 4.

Now please run the following code in the end:

```
python check.py --uid 123456
(replace 123456 as you student number)
```

```
Your Student ID is 1308184. Is this correct? (y/n): y
```

Input "y",then press "enter", you will see following information.

```
********* Zipping Files *********
Zipping Lab4_1.java... Done.
Zipping Lab4_2.java... Done.
Zipping answer_sheet.pickle... Done.

🎉 Successfully created submission package:
Lab4_1308184_0422_2139.zip
```

You will see there is a new file under the folder like: