CSC 230: Elementary Data Structures and Algorithms

Fall 2022

Assignment 6

- General programming guidelines:
  - Create a separate NetBeans project with the name `QuestionXX`, where `XX` is the question number.
  - Do not forget necessary javadoc comments before classes and methods.
  - You should use *single line* or *multiline* comments, **if it is required**. Do not put unnecessary comments (Do not state the obvious!!!).
  - Use meaningful identifier.
  - Don't forget to check the parameters of your methods and throw appropriate exceptions as necessary.
  - **Creating correct NetBeans projects, zipping your final assignment folder, and testing it before uploading are your responsibilities. If any of these steps fails, you will receive a grade of zero.**
- Academic integrity policy
  - You are **not** allowed to use any online resources EXCEPT class lecture notes and Java documentation.
  - All programs/ code must be your own work.
  - You should be able to clearly explain every line of your code, if instructor requests you to do so.
  - Any violation of these policies will be considered as plagiarism and dealt accordingly.

Question 1 (25 points): Write a recursive method to find the length of a string. You should not use `length` method. Obviously, your method should take a string as a parameter and returns an integer. Make sure to provide test cases. Show your method works by using a few test cases.

Note: Do not worry about the use of `substring`, though we know it is not ideal in this situation.

Question 2 (25 points): Write a recursive method to reverse a non negative integer. Your method should take a non negative integer as a parameter and return the reverse of the number as an integer. e.g. if you pass 12345, your method should return 54321. Show your method works by using a few test cases.

Note: Do not use number to string and string to number conversions inside your method. In other words, this problem can be solved using simple integer arithmetic. No points will be awarded, if you use strings or arrays.

Question 3 Methods `reverses` shown below return the reverse of a `String`.

1. Implement `reverse1` recursively. Do not worry about the inefficiency of string concatenation.
   `private static String reverse(String s, int position, String result)`
   `public static String reverse1(String s)`
2. Implement the same algorithm, but this time use a `StringBuffer` instead of a `String` to improve the efficiency.
   `private static StringBuffer reverse(String s, int position, StringBuffer sb)`
   `public static String reverse2(String s)`

Show your methods, `reverse1` and `reverse2`, work by using a few test cases.

Question 4 (25 points): Write a recursive method (say `fib_bad`) to calculate the n-th Fibonacci number using our naive approach discussed in the class. Now, write another recursive method (say `fib_good`) that utilizes an array to store previously calculated Fibonacci numbers for future use as recursion progresses. Try both of these algorithms with relatively large n and observe the performance difference.