Due : Oct. 10 (10/10), 13:00
Late submission due: Oct. 11 (10/11), 13:00

Overview

This assignment consists of 2 parts.

General Notes

- *Read this homework guideline carefully.* If you do not follow the guidelines, you may receive a 0 regardless of whether your code works or not.
- Do not use any IDEs (Eclipse, IntelliJ IDEA, etc.)
  - We recommend Sublime Text (Linux/Mac/Windows), Atom (Linux /Mac/Windows), Notepad++ (Windows), or TextWrangler (Mac).
  - IDEs often create a "package" of your code, which breaks the autograder.
  - **If you know how to fix the package problem**, you can use any IDE you want. However, we will not answer any questions related to this problem since we have already recommended a solution.
- Do not change any method or class signatures. If your code changes any class or method names or signatures, you will receive an automatic 0.
- Make sure your code compiles. Non-compiling code will automatically receive a 0. If you have a problem that is causing you to not be able to compile, it may be better to just comment out the incorrect code and return a dummy value (something like null or -1) so the rest can compile.
- To ensure that your code will be accepted by the autograder, you should submit your code on LearnUS, download it again, recompile it and check the provided test suite. This way, you know that the file you are submitting is the correct one.
- You can use any course materials. However, if you do not cite the source you referred to, it might be checked as copied code. Please write the material (and page) you referred in comments.

1 String

A string is a sequence of symbols ('a', 'b', ... ,'z'). You will implement the string data structure based on doubly linked list. You should use the ListNode class we provided.

You must implement the following methods as well as its constructor in MyString.java file:

`MyString`: Constructor of the MyString. Create an empty string.

`append(x)`: Insert the given symbol x as the last element of the list.

`prepend(x)`: Insert the given symbol x as the first element of the list.

`findFirst(x)`: For a given symbol x, return the smallest index of a node with x. We consider that the index starts with x.

For example, the result of find first 'a' in 'cbaba' is 2. If there is no such node in the string, return -1.

`findLast(x)`: For a given symbol x, return the largest index of a node with x. For example, the result of searching 'a' in 'cbaba' is 4. If there is no such node in the string, return -1.

`lessOrEqual(str2)`: Compare the ternary string itself and the given string $str2$. For example, '$str1$.lessOrEqual($str2$)' compares '$str1$' and '$str2$' in the lexicographical order and returns whether or not $str1 \leq str2$. Note that the empty string '' is the smallest.

For example, 'ab' $\leq$ 'ab', 'a' $\leq$ 'aac', 'ab' $\leq$ 'c' and 'baa' $\leq$ 'bac'.

`print()`: Return the whole string composed by the characters from head to tail.

`size()`: Return the length of the string.

2 Nonogram

A nonogram is a picture logic puzzles in which cells in a grid must be filled or left empty according to numbers at the side of the grid to reveal a hidden picture.

A number according to a row or column indicates the number of continuous filled cells in that row or column. If there are multiple numbers according to a row or column, there is at least one empty cell between the filled cells. For example, “10 4” means that there are 10 cells and 4 cells filled continuously, seperated by at least one cell in between.

10	4		🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥			🟥	🟥	🟥	🟥		

You must implement the following methods as well as its constructor in BoardSolver.java file:

> `solve(board)`: Given a board with at least one cell (1x1), return a completed board satisfying the conditions. If the conditions cannot be matched, return *null*.

General Directions

- Write your name and student ID number in the comment at the top of the files you submit.
- Implement all of the required methods.
- You should not import anything that is not already included in the file.
- Pay careful attention to the required return types and edge cases.
- All the codes we provide can be found in src/base directory. If you are unsure what a class/method exactly does, please refer to the code.
- You are free to implement any algorithm that you wish, but you must code it yourself. If you referred to any course materials, you must write the name of the material and page in comments. We will only be testing that your code produces a correct result and terminates in a reasonable amount of time.

Submission Procedure

You *must* make a zip file for submission using Gradle build tool (refer to Compiling section). For this assignment, the zip file will contain only the following three files:

- MyString.java
- BoardSolver.java
- your_student_id_number.txt

You must rename 2021xxxxxx.txt to your actual student ID number. Inside of that text file, you must include the following text. Please be sure to write all the following text including the last period.

In completing this assignment, I pledge that I have not given nor received any unauthorized assistance.

If this file is missing, you will get 0 on the assignment. It should be named *exactly* your student id, with no other text. For example, *2021123456.txt* is correct while something like *2021123456_pa1.txt* will receive 0.

Compiling

This assignment uses Gradle build tool to automate compiling and testing procedure. The following command will test your Java code against the provided test suite:

```
% ./gradlew -q runTestRunner
```

The following command will zip the files for your submission. The zip file will be named with your student id (the name of .txt file) and will lie in "build" directory. Be aware that the command will be interrupted if your pledge does not comply the guideline.

```
% ./gradlew -q zipSubmission
```

Since the testrunner blocks the standard output from printing, it is hard to test your code fragment while writing the code. For this purpose, we also provide an empty Main class. As this file is not for submission, you may use any features Java provides. The following command will run the Main class instead of the test suite.

```
% ./gradlew -q --console=plain runMain
```

On Windows, try `gradlew.bat` instead of `./gradlew` if you met an error. Moreover, you may omit the '`-q`' option to review the compile log.

Testing

We have provided a small test suite (src/test) for you to check your code. You can test your code by the means mentioned above.

Note that the test suite we will use to grade your code is much more rigorous than the one provided here (and not necessarily a superset of the provided tests). You should consider making your own test suites to check your code more thoroughly.