CSC1003 Assignment 3

Important Notes:

1. The assignment is an individual project, to be finished on one's own effort.
2. The work must be submitted before 6pm Dec. 2, 2023 (Monday), Beijing Time. This is a firm deadline. No late submissions are accepted.
3. Plagiarism is strictly forbidden, regardless of the role in the process. Notably, ten consecutive lines of identical codes are treated as plagiarism. Depending on the seriousness of the plagiarism, 30%-100% marks will be deducted.

Marking Criteria:

1. The maximum score of the assignment is 100 marks (i.e. more than 100 marks will be counted as 100 marks).
2. Each program will be evaluated with several unseen test cases. A program obtains the full score if and only if it passes all test cases.
3. According to the school policy, using AI will be regarded as academic dishonesty. If TA suspects that your code was generated by AI, you will be invited to an offline meeting and express your understanding of the assignment. Otherwise, you may receive a minimum grade.

Running Environment:

1. The submissions will be evaluated in the course's OJ system running Java SDK. It is the students' responsibility to make sure that his/her submissions are compatible with the OJ system.
2. The submission is only allowed to import four packages of (java.lang.*; java.util.*; java.math.*; java.io.*) included in Java SDK. No other packages are allowed.
3. All students will have an opportunity to test their programs on the OJ platform prior to the official submission.

Submission Guidelines:

1. The score is based on your submission on OJ, but you also need to submit your code (that is, **directly** submit *the screenshots, RegionFill.java and TestMathExpr2.java*) on blackboard. Otherwise, you will get 0 for this assignment.
2. Inconsistency with or violation from the guideline leads to marks deduction.
3. All students are reminded to read this assignment document carefully and in detail. No argument will be accepted on issues that have been specified in this document.

Exercise One (15 marks):

Finish at least five problems (https://oj.cuhk.edu.cn/d/csc1003_2024_fall/p) from P1015 to P1025 online, and upload screenshot(s) to show that you finished the exercise. You can earn 15 marks without requiring "AC" by attempting to complete 5 questions.

Exercise Two (50 marks):

Write a Java program (RegionFill.java) to simulate the process of filling color to a closed region. The region is formed by a closed curve and/or the boundary. The program starts from a point with given coordinates and fills in the "color" for all pixels inside the closed region.

Example input	Expected output
**********	**********
_____**	**_____****
*_*****_****	*_______****
*_******_****	*________****
_****_****	**________****
_***_****	***________****
_****_****	***_________****
_*****_****	***__________****
****_*******_****	****_________****
*****_******_****	*****________****
******_*****_****	******_______****
*******_*****_****	*******_______****
********_****_****	********_____****
*********__**_****	*********____****
**********_*__****	**********___****
**********_*******	**********_*******
**********	**********
**********	**********
**********	**********
**********	**********
9, 11	
**********	__________
_____**	__________
*_*****_****	__*****_____
*_******_****	__******_____
_****_****	___******_____
_***_****	____******_____
_****_****	____*******_____
_*****_****	____********_____
****_*******_****	_____*******_____
*****_******_****	______******_____

******_*****_******* *******_*****_****** ********_****_****** *********__**_****** **********_*_****** ***********_******* ******************** ******************** ******************** ******************** 18, 0	_______*****_______ ________*****_______ _________****_______ __________**_______ __________*_______ ____________________ ____________________ ____________________ ____________________ ____________________

The console input is **always** given by **20 * 20** characters of either "*" or "-". A closed region can be formed by the "-" characters, or by the "-" characters and the "*" characters in the boundary. Following the input of the characters, **two numbers** of the (x, y) coordinates of the starting point are given, separated by a comma and a space.

In the first example above, the starting point is (9, 11), which is the 12th character of line 10. The corresponding output of the result is shown in the right.

In the second example above, the starting point is (18, 0), which is the 1st character of line 19. For this starting point, the corresponding output of the result is shown in the right.

Note: the whole region could possibly be divided into more than 2 parts.

Exercise Three (50 marks):

Write a Java Program (TestMathExpr2.java) with the following requirement.

1. It evaluates the value of math expressions, and outputs an integer value.
2. Each math expression includes (see the example below):
   - 2.1 numbers (integers and doubles);
   - 2.2 (no more than five) operators of "+" (addition), "-" (subtraction), "*" (multiplication) and "/" (division);
   - 2.3 (no more than three) functions including "sin" (sine function), "cos" (cosine function), "tan" (tangent function) and "sqrt" (square root function).
   - 2.4 "(" and ")" (brackets);
   - 2.5 possibly blank space, but there's no blank space inside a function (e.g. "sq rt") or a number (e.g. "12 3.4").
3. All expressions are valid. The output is an **integer value after rounding**.
4. The number of math expressions T follows $1 \leq T \leq 20$.

Example input	Expected output
1+2.0*sin(37+(25*3))	-1
(2+ 3.50)*4*sqrt(sin(1.5))	22
-3+4/ (2.5+3.7)	-2
(-3+4)/2.5+3.9	4
1.2-3.5*5.2- 13.2	-30

1.2-3.5*5.2-13.7	-31
2.3*5*7 - 12*9/8	67
-sin(3.5-sqrt(4)) + cos(tan(2.5))	0

Note: Each submission is expected to **strictly follow** the following template to implement the required function by modifying the **parse()** function/method.

```
import java.util.*;

public class TestMathExpr {

    public static double parse(String str) {
        // implement the code here
        // evaluate a math expression, and return the value
        return 0.0d;
    }

    public static void main(String[] args) throws Exception {
        Scanner input = new Scanner(System.in);
        while (input.hasNextLine()) {
            double result = parse(input.nextLine());
            System.out.println(String.valueOf(Math.round(result)));
        }
    }
}
```