Java Development Homework 3&4

Due before 2025 April 9 11:59pm

Notes

1. Before answering, make sure you are familiar with the operation process of the online judge (automated grading system).

2. The online judge will evaluate the code submitted to Moodle after the deadline. Once you have completed your code, be sure to submit it to Moodle.

3. The online judge has a code similarity detection system, and both the person who plagiarizes and the one whose code is plagiarized will receive a score of 0.

4. If you have any questions about the problem, please contact the teaching assistant.

Submission

Please archive your source code to STUDENT_ID.zip (download the example zip file from Moodle) and **upload to Moodle Homework 1** before deadline.

Your zip file should follow the following format.

```
STUDENT_ID.zip

|- src

|- META-INF

| |- MANIFEST.MF
```

All the source files (*.java) are put in the src directory.

The entry point (i.e. main class) of the program is specified in the MANIFEST.MF file.

No late submission is accepted.

Homework 3

Problem Description

Problem: Palindromic Substrings

Description:

Given a string s, return the number of palindromic substrings in it.

Where:

- A string is a palindrome when it reads the same backward as forward.

- A substring is a contiguous sequence of characters within the string.

Sample Input and Output

Keyboard Input	abc
Output	3

Keyboard Input	aaa
Output	6

Keyboard Input	cbazyxwvutsrqponmlkjihgfedcbazabcdefghijklmtuvwxy
Output	62

Input:

- Each case inputs one line of string s.

Output:

- the number of palindromic substrings.

Constraints:

- 1 <= s.length <= 1000
- s consists of lowercase English letters.

Homework 4

Problem Description

Problem: Linked List Cycle Detection

Description:

You are given a linked list with n pairs of id and next values. Each pair corresponds to a node in the linked list.

The first pair represents the head of the linked list. Some of the nodes might form a cycle, and you are required to detect it.

Your task is to check if the linked list has a cycle and, if it does, return the index of the node with the smallest id in the cycle. If no cycle exists, return -1.

Definition for singly-linked list :

```
class ListNode {
    int val;
    ListNode next;
    ListNode(int x) {
        val = x;
        next = null;
    }
}
```

Sample Input and Output

Keyboard Input	4
	0 1
	1 3
	3 2
	2 1

Output	1

Keyboard Input	2
	0 1
	1 2

Output	-1

Input Format:

- **The first input line contains a single integer n, representing the number of (id, next) pairs that describe the nodes in the linked list.**
- **The following n lines each contain two integers: id and next.**
  - **id is the index of the current node.**
  - **next is the index of the node that the current node points to**
  - **The first pair (id, next) corresponds to the head of the linked list.**

Output Format:

- **If a cycle exists, return the index of the node with the smallest id in the cycle. Otherwise, return -1.**

Constraints:

- **1 <= n <= 1000**
- **The node ids are distinct and range from 0 to n.**
- **The next pointer is either a valid node index (from 0 to n)**