Brackets

Given a bracket sequence of length **L**, you are to determine if it is valid!

A valid bracket sequence is defined recursively as:

- "" (empty sequence) or
- (**x**) where **x** is a valid bracket sequence, or
- **xy**, where **x** and **y** are valid bracket sequences.

However, two other pairs of brackets can be found in the bracket sequence, '[' which is to be matched with ']' and '{' which is to be matched with '}'.

Input
The first line will be a single integer **N**. There will be a total of **N** testcases per input file. Each testcase will contain 2 consecutive lines, in the below format:
The first line of each testcase will contain a single integer **L**.
The second line of each testcase will contain one string, the bracket sequence.

Output
On the **i**-th line, output "Valid" if the bracket sequence in the **i**-th testcase is valid or "Invalid" if it is not valid.

Limits

- 0 < **N** ≤ 50
- 0 < **L** ≤ 100,000
- Sum of **L** in each input file will not exceed 2,000,000.
- It is guaranteed that the string will be exactly length **L** and contain only '(', ')', '[' , ']', '{' and '}' characters.
- **Please be reminded that strings are immutable in Java.**

Sample Testcase

Sample Input (**brackets1.in**)	Sample Output (**brackets1.out**)
8 6 ([]{}) 8 (())(((() 6 ([}{]) 6 (){}[] 1 [10 {{{{{}}}}} 2)(2 (}	Valid Invalid Invalid Valid Invalid Valid Invalid Invalid

Notes:

1. You should develop your program in the subdirectory **ex1** and use the skeleton java file provided. You should not create a new file or rename the file provided.
2. You are free to define your own helper methods and classes (or remove existing ones).
3. Please be reminded that the marking scheme is:
   a. Public Test Cases (1%) - 1% for passing **all** test cases, 0% otherwise
   b. Hidden Test Cases (1%) - Partial scoring depending on test cases passed
   c. Manual Grading (1%)
      i. Overall Correctness (correctness of algorithm, severity of bugs)
      ii. Coding Style (meaningful comments, modularity, proper indentation, meaningful method and variable names)
4. Your program will be tested with a time limit of not less than **2 sec** on Codecrunch.

Skeleton File – Brackets.java

You are given the skeleton file `Brackets.java`. You should see a non-empty file when you open the skeleton file. Otherwise, you might be in the wrong working directory.

You should see the following contents when you open the skeleton file:

```
import java.util.*;
public class Brackets {
    private void run() {
        //implement your "main" method here
    }
    public static void main(String[] args) {
        Brackets newBrackets = new Brackets();
        newBrackets.run();
    }
}
```