**Naming Conventions**

1. All the names are meaningful.  
2. Only a single-character variable appears, and it’s used as index in a loop.  
3. Every class in the method is correctly-written.  
4. Every interface recalled in the method follows the class-naming convention.  
5. Every method’s name begins with a verb in lower-case characters.  
6. Every attribute follows the attribute’s naming conventions.  
7. No constants are used.

**Indention**

8. Every indentation is made of four spaces.  
9. Automatic tabulation is used two times at line 938 (last two indentations).

**Braces**

10. The chosen bracing convention is the Kernighan and Ritchie one.  
11. No single-line bocks not contained between curly braces are present.

**File Organization**

12. The following blank lines could be removed, because they do not separate any section:

* 935
* 939
* 943
* 948
* 951
* 953
* 956
* 960
* 967
* 969

13. No line exceeds 80 characters.  
14. No line needs to exceed 80 characters.

**Wrapping Lines**

15. In the following lines the line break occurs before the operator, but should appear after it:

* 937

16. In the following lines, an high level break would be preferred, but in this way the lines will exceed 80 characters: an evaluation should be made in order to decide which of the two problems solve.

* 937-938

17. Every new statement is aligned with the beginning of the expression of the same level as the previous line.

**Comments**

18. There are no comments in the method.  
19. The method does not contains commented out code.

**Java Source Files**

20. The file containing the method provides only one public class; no private classes are present.  
21. The public class is the first class of the file.  
22. *Common part*  
23. Javadoc is written for this method and explains correctly all the information needed.

**Package and Import Statements**

24. *Common part*

**Class and Interface Declarations**

25. *Common part*  
26. *Common part*  
27. There is no cloned or replicated code in this method.

**Initialization and Declarations**

28. Variable “equal” (line 932) should be declared with private visibility, due to the fact that it’s used only in this method.  
29. Every variable is declared in the proper scope.  
30. Call to constructors is not necessary.  
31. Every object is initialized before the use.  
32. Every variable is initialized where is declared.  
33. Every declaration appears at the beginning of the block in which it appears.

**Method Calls**

34. Every method is invoked with the correct order of arguments.  
35. Every invoked method is the right one to invoke.  
36. Every returned value is used properly.

**Arrays**

37. No off-by-one errors are present.  
38. All array indexes have been prevented from going out-of-bounds.  
39. Arrays are given as arguments, and they don’t need new items creation.

**Object Comparison**

40. In the comparison statements is always used “equals”, except for “int” variables comparisons, that can be compared with “==” due to the fact that they are of primitive types.

**Output Format**

41. No output is present.  
42. No output is present.  
43. No output is present.

**Computation, Comparisons and Assignments**

44. The method implementation avoids brutish programming.  
45. All the parenthesizing and operators precedence are in the correct order.  
46. Parenthesis are correctly used to avoid operator precedence problems.  
47. There are no denominators to check.  
48. All the integer arithmetic expressions are used appropriately.  
49. Comparisons and Boolean operators are correct.  
50. The possible exception are thrown in the invoked methods. The “finally” block ensures that in case of error the computation continues.  
51. There aren’t implicit type conversions: all the type conversion are made explicitly using a cast.

**Exceptions**

52. No exception is caught.  
53. There are no catch blocks.

**Flow Control**

54. There are no switch statements.  
55. There are no switch statements.  
56. There is only one loop that is correctly initialized and has defined bounds; furthermore the bounding variables are not changed inside the loop.

**Files**

57. No file is used.  
58. No file is used.  
59. No file is used.  
60. No file is used.