**Task 1: Read in (and validate) a Boolean function given as:**

a. A SoP (Sum of Products) expression, for example: abc + a'b'c

b. A PoS (Product of Sums) expression, for example: (a + c)(d' + e)

**Checkers for Validating SoP Expressions:**

1. Variable Format Checker: This checker ensures that variables are represented using uppercase and lowercase letters (A-Z, a-z). If the character doesn't fall within this range, it's considered invalid.
2. Checker: if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z'))

Single Quotes Checker: This checker verifies that single quotes are used only to indicate complemented variables and that they follow a valid variable. If a single quote is encountered and the previous character was not a variable, it's considered invalid.

1. Checker: if (c == '\'' && !prevWasVariable)

Plus Operator Checker: This checker ensures the plus operator + is used correctly within the expression. It should be used between variables or complemented variables.

1. Checker: if (c == '+' && !prevWasVariable)

Open and Close Bracket Checker: These checkers ensure that brackets ( and ) are balanced and used correctly in the expression. Open brackets should be followed by valid expressions, and close brackets should have corresponding open brackets.

1. Checker for open bracket: if (c == '(')

Checker for close bracket: if (c == ')' && openBrackets > 0)

Invalid Character Checker: This checker detects and handles any characters that are not part of the valid expression. It ensures that only spaces, variables, single quotes, plus operators, and brackets are allowed.

1. Checker: if (c != ' ')

Invalid Combination Checker: This is a more general checker that can be used to detect invalid combinations of characters, such as two consecutive variables without an operator in between. The specific implementation for this checker would vary depending on the exact requirements of your validation logic.

**Checkers for Validating PoS Expressions:**

1. Variable Format Checker: Similar to the SoP expression, this checker ensures that variables are represented using uppercase and lowercase letters (A-Z, a-z) and follow the correct format. It should also check that a variable follows a valid operator.
2. Checker: if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') && !prevWasVariable)

Plus Operator Checker: This checker ensures that the plus operator + is used correctly within the expression. It should be used between valid expressions, which include variables and complemented variables.

1. Checker: if (c == '+' && prevWasVariable)

Open and Close Bracket Checker: These checkers verify that brackets ( and ) are balanced and used correctly in the expression. Open brackets should be followed by valid expressions, and close brackets should have corresponding open brackets. Additionally, open brackets should not follow a variable directly.

1. Checker for open bracket: if (c == '(' && !prevWasVariable)

Checker for close bracket: if (c == ')' && prevWasVariable)

Single Quotes Checker: This checker ensures that single quotes are used to indicate complemented variables and that they follow a valid variable. The single quote should not appear after an operator.

1. Checker: if (c == '\'' && prevWasVariable)

Invalid Character Checker: Like in the SoP expression, this checker detects and handles any characters that are not part of the valid expression. It ensures that only spaces, variables, single quotes, plus operators, and brackets are allowed.

1. Checker: if (c != ' ')

Invalid Combination Checker: This is a more general checker that can be used to detect invalid combinations of characters, such as two consecutive variables without an operator in between. The specific implementation for this checker would vary depending on the exact requirements of your validation logic.