SpreadSheet

This exercise aims creating a basic spreadsheet. The goal **is not** to develop the user interface, but the code that implements the data structures and performs the operations. The spreadsheet follows Excel's conventions. The spreadsheet implements three operations: *set()*, *get()* and *evaluate()*. Set and get assign and return cell's values. Evaluate takes the cell's content and returns the result (e.g.: a cell containing "=1+1" evaluates to 2).

1. Setup

There are two basic operations:

- *set(), that assigns a value to a cell*
- *get(), that retrieves the value of a cell (e.g.: for editing)*

Requirement: Define the operations *set()* and *get()*.

Example: if *set("A1", "1")* is run, a subsequent *get("A1")* gives: "1".

2. Number handling

The spreadsheet should handle correctly formatted integer numbers

Requirement: The spreadsheet shall be able to store and return integer numbers, both signed and unsigned. When a cell containing an integer number is evaluated, the result of the evaluation shall be the number itself.

Example: if *set("A1", "-1")* is run, a subsequent call to *evaluate("A1")* returns the same number: "-1".

3. Wrongly formatted integers

The spreadsheet detects incorrectly formatted integers

Requirement: When a number does not follow the integer format (e.g.: contains a decimal point, special symbols, characters, etc.), the evaluation shall return the error message #Error.

Example: if *set("A1", "5A")* is run, a subsequent call to *evaluate("A1")* returns "#Error".

4. String handling

The spreadsheet should handle arbitrary strings

Requirement: The spreadsheet shall be able to store and return strings. All strings are entered between single quotes. When a cell containing a string is evaluated, the result of the evaluation shall be the same string without quotes. Removing the quotes eases displaying strings in the screen correctly, as well as concatenating with other strings (see below).

Example: if *set("A1", "'a string'")* is run, a subsequent call to *evaluate("A1")* returns the same string without quotes: "a string".

5. Unquoted strings

The spreadsheet detects incorrectly formatted strings

Requirement: When a string does not have heading or trailing quotes, the evaluation shall return the error message #Error.

Example: if *set("A1", "'a string")* is run, a subsequent call to *evaluate("A1")* returns "#Error".

6. Simple formulas

The spreadsheet evaluates simple formulas (without operators or cell references)

Requirement: When a cell contains a "=" sign, followed by a string or integer number, the evaluation of that cell shall return the corresponding string or integer (see Req. 2, 4).

Example: if *set("A1", "='a string'")* is run, a subsequent call to *evaluate("A1")* returns "a string".

7. Simple formulas with errors

The spreadsheet detects errors in simple formulas

Requirement: When a cell contains a "=" sign, followed by a wrong string or integer number (see Req. 3, 5), the evaluation of that cell shall return the error message #Error.

Example: if *set("A1", "='a string")* is run, a subsequent call to *evaluate("A1")* returns "#Error".

8. Cell references

The spreadsheet handles cell references in formulas

Requirement: When a formula contains a reference to a cell (following Excel convention, e.g.: A5), the evaluation shall be recursive, i.e.: the referenced cell is evaluated, and the result is returned by the formula.

Example: if A5 contains the integer "5" and *set("A1", "=A5")* is run, a subsequent call to *evaluate("A1")* returns "5".

9. Errors in cell references

The spreadsheet detects errors when cell references in formulas are used

Requirement: When the value contained in a cell referenced by a formula is incorrect, the evaluation shall return the error message #Error.

Example: if A5 contains the wrong integer "5A" and *set("A1", "=A5")* is run, a subsequent call to *evaluate("A1")* returns "#Error".

10. Circular references

The spreadsheet detects circular references in formulas

Requirement: When a formula contains circular references, the evaluation shall

return the error message #Circular.

Example: if A5 contains the formula "=A1" and *set("A1", "=A5")* is run, a subsequent call to *evaluate("A1")* returns "#Circular".

11. Integer operations

The spreadsheet performs the integer addition, subtraction, multiplication, division, and module.

Requirement: When the operators + - * / % are used in formulas, the evaluation shall return the results of the addition, subtraction, multiplication, division, and module operations, respectively. Usual precedence does not apply, i.e.: formulas are always evaluated from left to right.

Example: if *set("A1", "=1+1*2")* is run, a subsequent call to *evaluate("A1")* returns "4".

12. Errors in integer operations

The spreadsheet identifies errors in integer operations

Requirement: When an operation cannot be performed because (1) the operators are incorrect integer numbers or (2) there is a division by zero, the evaluation shall return the error message #Error.

Example: if *set("A1", "=1+1A")* is run, a subsequent call to *evaluate("A1")* returns "#Error".

13. String operations

The spreadsheet performs the concatenation of strings

Requirement: When the operator & is used in formulas, the evaluation shall return the concatenation of the respective string operators.

Example: if *set("A1", "='a'&'string'")* is run, a subsequent call to *evaluate("A1")* returns "astring".

14. Errors in string operations

The spreadsheet identifies errors in string concatenation

Requirement: When a concatenation cannot be performed because the strings are wrongly formatted (see Req. 5) the evaluation shall return the error message #Error.

Example: if *set("A1", "='a&'string'")* is run, a subsequent call to *evaluate("A1")* returns "#Error".

15. Parentheses

The spreadsheet handles parentheses in formulas

Requirement: When a formula contains parentheses, the internal part shall be evaluated first.

Example: if *set("A1", "=1+(1*2)")* is run, a subsequent call to *evaluate("A1")*

returns “3”.

16. Errors using parentheses

The spreadsheet detects unbalanced parentheses

Requirement: When a formula contains unbalanced parentheses, the evaluation shall return the error message #Error.

Example: if *set(“A1”, “=1+(1*2”)* is run, a subsequent call to *evaluate(“A1”)* returns “#Error”.

17. Spaces

Formulas can contain arbitrary spaces

Requirement: Unnecessary spaces in formulas are ignored.

Example: if *set(“A1”, “ =1+ (1 * 2)”)* is run, a subsequent call to *evaluate(“A1”)* returns “3”.

Congratulations, you are done!