3484. Design Spreadsheet

Medium | Topics | Companies | Hint

A spreadsheet is a grid with 26 columns (labeled from `'A'` to `'Z'`) and a given number of `rows`. Each cell in the spreadsheet can hold an integer value between 0 and 10^5.

Implement the `Spreadsheet` class:

- `Spreadsheet(int rows)` Initializes a spreadsheet with 26 columns (labeled `'A'` to `'Z'`) and the specified number of rows. All cells are initially set to 0.
- `void setCell(String cell, int value)` Sets the value of the specified `cell`. The cell reference is provided in the format `"AX"` (e.g., `"A1"`, `"B10"`), where the letter represents the column (from `'A'` to `'Z'`) and the number represents a **1-indexed** row.
- `void resetCell(String cell)` Resets the specified cell to 0.
- `int getValue(String formula)` Evaluates a formula of the form `"=X+Y"`, where `X` and `Y` are **either** cell references or non-negative integers, and returns the computed sum.

Note: If `getValue` references a cell that has not been explicitly set using `setCell`, its value is considered 0.

```python
class Spreadsheet:

    def __init__(self, rows: int):
        self.ss = [ 0 for _ in range(26) ] for _ in range(rows)
        self.rows = rows

    def parsecell(self, cell: str):
        col = ord(cell[0]) - ord('A')
        row = int(cell[1:]) -1
        return col, row

    def setCell(self, cell: str, value: int) -> None:
        r, c = self.parsecell(cell)
        self.ss[r][c] = value


    def resetCell(self, cell: str) -> None:
        r, c = self.parsecell(cell)
        self.ss[r][c] = 0

```

correct

```python
    def getValue(self, formula: str) -> int:
        parts = re.split('()[]+-*/', formula[1:])
        res = 0

        if len(parts) == 3:
            a = parts[0].strip() # A1
            b = parts[1].strip() # operand
            c = parts[2].strip() # B2

            c1, r1, c2, r2 = int(a[0] - ord('A')), int(a[1:]) -1, int(c[0] - ord('A')), int(c
[1:]) -1
            x, y = min(0, ss[r1][c1]), min(0, ss[r2][c2])

            match b:
                case '+':
                    res = x + y
                case '-':
                    res = x - y
                case '*':
                    res = x * y
                case '/':
                    res = x / y

        return res
```

need modification:
- always sum, no need to specify op

updated:

```python
class Spreadsheet:

    def __init__(self, rows: int):
        self.ss = [[ 0 for _ in range(26) ] for _ in range(rows)]
        self.rows = rows

    def parsecell(self, cell: str):
        if cell[0].isalpha() and cell[1:].isdigit():
            col = ord(cell[0]) - ord('A')
            row = int(cell[1:]) - 1
            if 0 <= col <= 26 and 0 <= row <= self.rows:
                return row, col
        return None

    def setCell(self, cell: str, value: int) -> None:
        pos = self.parsecell(cell)
        print(pos, self.rows)
        if pos:
            self.ss[pos[0]][pos[1]] = value

    def resetCell(self, cell: str) -> None:
        pos = self.parsecell(cell)
        if pos:
            self.ss[pos[0]][pos[1]] = 0

```

parsecell():

helper func, return the position of the cell

```python
    def getValue(self, formula: str) -> int:
        assert formula.startswith('=')
        parts = formula[1:].split('+')
        res = 0


        if len(parts) == 2:
            a = parts[0].strip() # eg. A10 / 10
            c = parts[1].strip() # eg. B12 /12

            def resolve(x: str) -> int:
                x = x.strip()
                pos = self.parsecell(x)
                if pos:  # it's a cell ref
                    row, col = pos
                    if 0 <= row < self.rows and 0 <= col < 26:
                        return self.ss[row][col]
                    else:
                        # Out of spreadsheet range → treat as 0
                        return 0
                else:    # it's a number
                    return int(x)

            return resolve(a) + resolve(c)
```

resolve():

- check the input is a location or string
- if it's a position, find the value of the cell; else return the int value