Problem Link:

<https://leetcode.com/problems/design-spreadsheet/?envType=daily-question&envId=2025-09-19>

Solution:

class Spreadsheet {

private:

int rows;

vector<vector<int>> grid;

pair<int, int> parseCell(const string& cell) {

int c = cell[0] - 'A';

int r = stoi(cell.substr(1)) - 1;

return {r, c};

}

int getOperandValue(const string& operand) {

if(isdigit(operand[0]))

{

return stoi(operand);

}

else

{

auto [r, c] = parseCell(operand);

return grid[r][c];

}

}

public:

Spreadsheet(int rows) : rows(rows), grid(rows, vector<int>(26, 0)) {}

void setCell(string cell, int value) {

auto [r, c] = parseCell(cell);

grid[r][c] = value;

}

void resetCell(string cell) {

auto [r, c] = parseCell(cell);

grid[r][c] = 0;

}

int getValue(string formula) {

formula = formula.substr(1);

size\_t plusPos = formula.find('+');

string l = formula.substr(0, plusPos);

string r = formula.substr(plusPos + 1);

int v1 = getOperandValue(l);

int v2 = getOperandValue(r);

return v1 + v2;

}

};

/\*\*

\* Your Spreadsheet object will be instantiated and called as such:

\* Spreadsheet\* obj = new Spreadsheet(rows);

\* obj->setCell(cell,value);

\* obj->resetCell(cell);

\* int param\_3 = obj->getValue(formula);

\*/