**Assumptions:**

1. Ingredients will only have decimal values
2. An Ingredient should appear at least one time in a sentence
3. A expression with one action followed by ingredient or vice versa is valid
4. All the ingredients will start with a decimal digit as its first character
5. Operator precedence is not considered while evaluating a sentence
6. Left to right precedence rule was followed while evaluation of a expression
7. An ingredient might appear multiple times in a sentence
8. The [very last] line next to last ingredient will form the core of expression and all the lines next to it are discarded.
9. The portion next to ingredient is trimmed and inserted in to the map, no matter what ever the string is.

**Time Complexity:**

1. Construction map of ‘n’ Ingredients will take **O(n log n)**, as STL map internally uses red black tree.
2. For evaluating a sentence with ‘m’ number of ingredients **O(m log n)**
3. If we assume each ingredient will appear only once in the sentence the n the time complexity is **O(n log n)**

**[no need to consider complexity of action map as its size is constant]**

**Approaches:**

1. By using Expression trees ( by doing this we can preserve Action precedence too..)
2. By using (head first)recursion while evaluating sentence

I followed the last approach, where I parsed all the ingredients and constructed map. The last line which might have have multiple sentences will be evaluated from left to right.

**Improvisations to be done:**

1. The solution works better with whole numbers need to handle underflow and overflow conditions for **multiplication and division**
2. The exception messages should be more precise and neat
3. Better usage of pointers
4. This solution is not thread safe