Testing Strategy

The first thing I did was to generally map out the structure of each function, and what I needed to do. I find it easier to write in plain C than in pseudocode, so I had basically done a rough bad C sketch of the code. Then I wrote one iteration of the program (I didn't have access to a compiler, so I did it all at once) trying to do things the hard way, and then ended up scrapping it. A rough draft of sort. Then after I felt like the code was getting too awry, I decided to start from scratch and coded almost all of it in one sitting at the library without a compiler, shoved it into VS Code, and within 15 minutes cleared the errors. Then it was a matter of fixing the memory leak which took another 15 minutes or so. At that point I quickly came up with some code that generates test cases by drawing random numbers within the specified intervals and printing them to a file, with the smallest ingredient being Ingredient1 and the largest ingredient being Ingredient0 and going up to Ingredient9999, all under 20 characters. I started with the randomizers being small so I could visually read the test cases to make sure they were formatted correctly and tested the code with a small format test case. Verifying this was correct, I generated a large format test case. Mimir was of no help and was utterly useless with its debugger while down to be honest.

Test Case Generator

```c
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
int main(){
    FILE *fp;
    fp = fopen("testcase.txt", "w");
    srand(time(NULL));
    int numIngredients = 1+rand()%10000; // n <= 10^5;
    fprintf(fp,"%d\n", numIngredients);
    for(int i = 0; i < numIngredients; i++){
        fprintf(fp, "Ingredient%d\n", i);
    }
    int numRecipes = 1+rand()%10000; // s <= 10^5
    fprintf(fp, "%d\n", numRecipes);
    for(int i = 0; i < numRecipes; i++){
        int numItems = 1+rand()%100;  //i <= 100
        fprintf(fp, "%d ", numItems);
        for(int j = 0; j < numItems; j++){
            fprintf(fp, "%d %d ", rand()%numIngredients, 1+rand()%1000); // Idx of Ingredient, parts up to 1000.
        }
        fprintf(fp, "\n");
    }
    int numStores = 1+rand()%100; // k <= 100 stores
    fprintf(fp, "%d\n", numStores);
    for(int i = 0; i < numStores; i++){
        int numRecs = 1+rand()%100;  //s <= 10^5
        fprintf(fp, "%d ", numRecs);
        for(int j = 0; j < numRecs; j++){
            fprintf(fp, "%d %d ", rand()%numRecipes, 1+rand()%1000); // Idx of Smoothie, weight up to 1000.
        }
        fprintf(fp,"\n");
    }
    fclose(fp);
    return 0;
}
```