LAB 3 – Functions

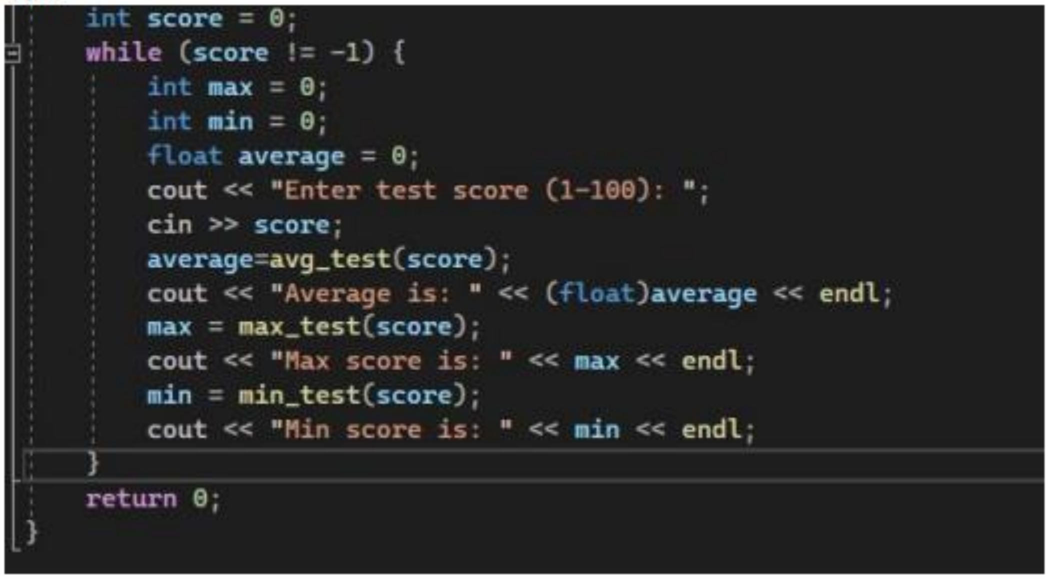
ELEC 3150 – Object Oriented Programming (Fall 2023)

Nick Cebula

HWQ1:

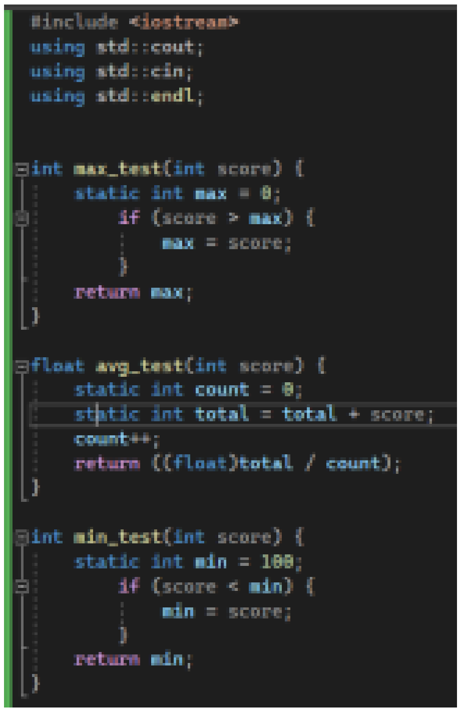
Question 1:

Main:



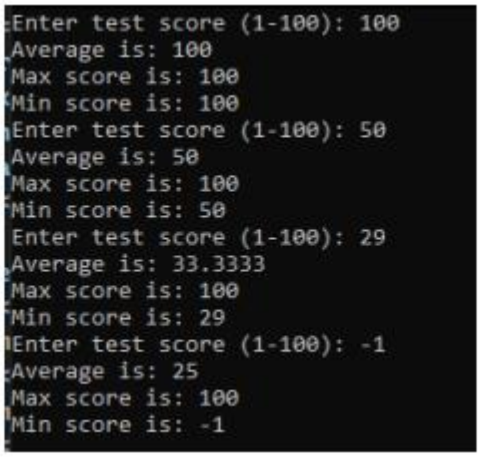
```
    int score = 0;
    while (score != -1) {
        int max = 0;
        int min = 0;
        float average = 0;
        cout << "Enter test score (1-100): ";
        cin >> score;
        average=avg_test(score);
        cout << "Average is: " << (float)average << endl;
        max = max_test(score);
        cout << "Max score is: " << max << endl;
        min = min_test(score);
        cout << "Min score is: " << min << endl;
    }
    return 0;
}
```

Mycode.cpp:



```
#include <iostream>
using std::cout;
using std::cin;
using std::endl;


int max_test(int score) {
    static int max = 0;
        if (score > max) {
            max = score;
        }
    return max;
}

float avg_test(int score) {
    static int count = 0;
    static int total = total + score;
    count++;
    return ((float)total / count);
}

int min_test(int score) {
    static int min = 100;
        if (score < min) {
            min = score;
        }
    return min;
}
```

Q1 Results:

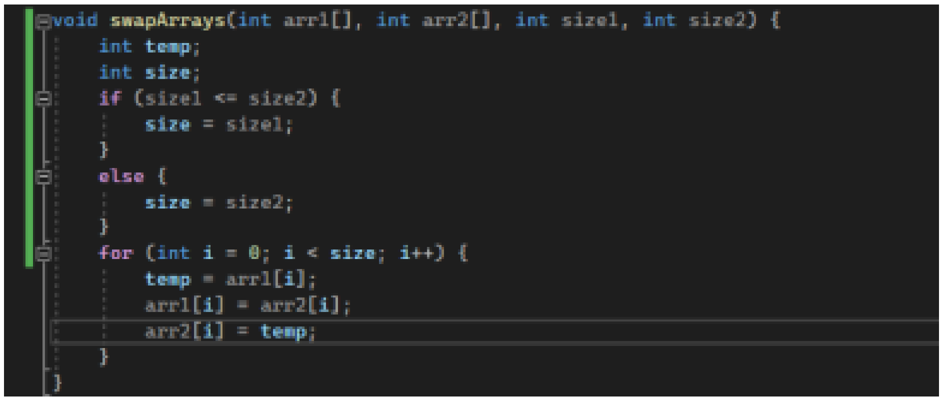


```
Enter test score (1-100): 100
Average is: 100
Max score is: 100
Min score is: 100
Enter test score (1-100): 50
Average is: 50
Max score is: 100
Min score is: 50
Enter test score (1-100): 29
Average is: 33.3333
Max score is: 100
Min score is: 29
Enter test score (1-100): -1
Average is: 25
Max score is: 100
Min score is: -1
```

Questions 1 Explanation:In the main.cpp file I created functions max test, avgtest, and mintest which do the calculations using static int to accumulate data from the user and store it. In the main when score is not equal to -1 the program asks the user to input score 1-100 and uses the functions to calculate the average, max score, and min score after each input.

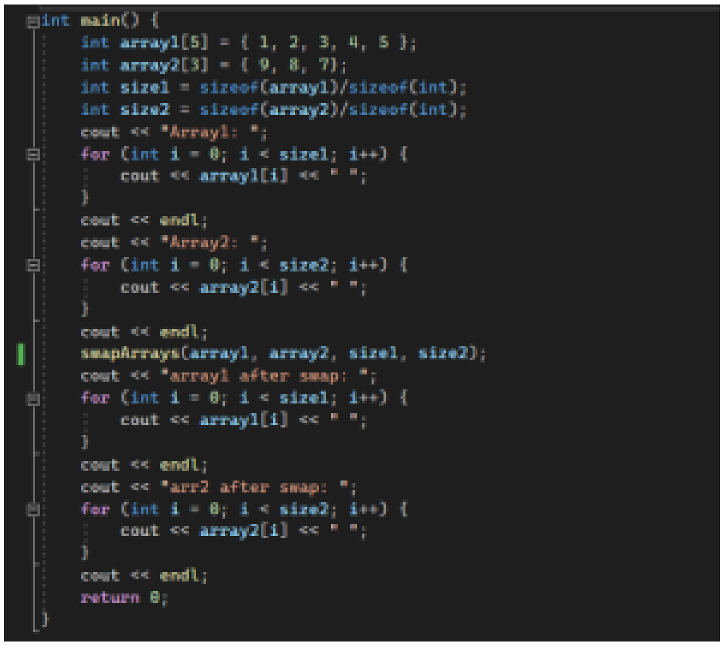
Q2:

Function:



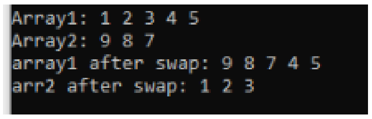
```cpp
void swapArrays(int arr1[], int arr2[], int size1, int size2) {
    int temp;
    int size;
    if (size1 <= size2) {
        size = size1;
    }
    else {
        size = size2;
    }
    for (int i = 0; i < size; i++) {
        temp = arr1[i];
        arr1[i] = arr2[i];
        arr2[i] = temp;
    }
}
```

Main:



```cpp
int main() {
    int array1[5] = { 1, 2, 3, 4, 5 };
    int array2[3] = { 9, 8, 7};
    int size1 = sizeof(array1)/sizeof(int);
    int size2 = sizeof(array2)/sizeof(int);
    cout << "Array1: ";
    for (int i = 0; i < size1; i++) {
        cout << array1[i] << " ";
    }
    cout << endl;
    cout << "Array2: ";
    for (int i = 0; i < size2; i++) {
        cout << array2[i] << " ";
    }
    cout << endl;
    swapArrays(array1, array2, size1, size2);
    cout << "array1 after swap: ";
    for (int i = 0; i < size1; i++) {
        cout << array1[i] << " ";
    }
    cout << endl;
    cout << "arr2 after swap: ";
    for (int i = 0; i < size2; i++) {
        cout << array2[i] << " ";
    }
    cout << endl;
    return 0;
}
```

Result:



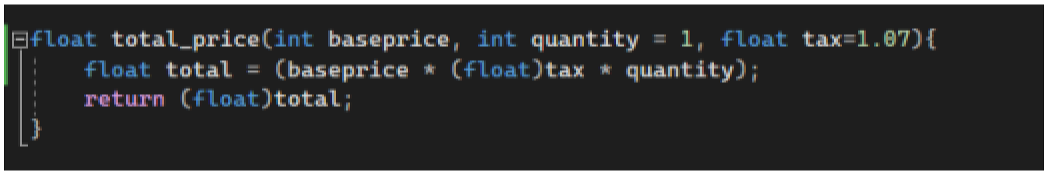
```
Array1: 1 2 3 4 5
Array2: 9 8 7
array1 after swap: 9 8 7 4 5
arr2 after swap: 1 2 3
```

HWQ2 Explain: Created a function that swaps arrays, this uses a if loop to determine the size, which will be used in the for loop to swap the 3 digits of the second array to the first 3 didigts of the first array.

In the main multiple for loops are used to print each of the arrays before and after the swap.

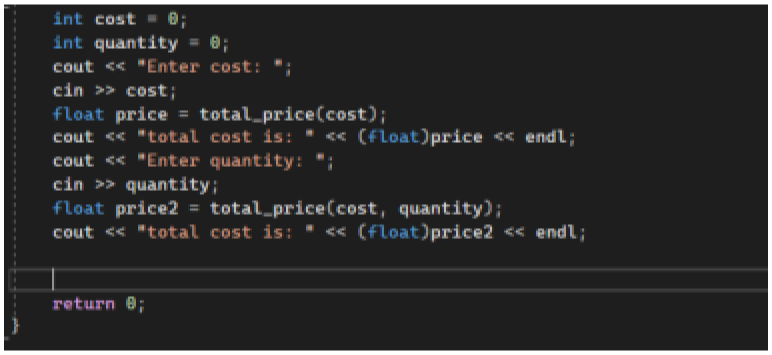
Q3:

Function:



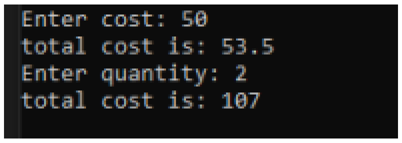
```cpp
float total_price(int baseprice, int quantity = 1, float tax=1.07){
    float total = (baseprice * (float)tax * quantity);
    return (float)total;
}
```

Main:



```cpp
    int cost = 0;
    int quantity = 0;
    cout << "Enter cost: ";
    cin >> cost;
    float price = total_price(cost);
    cout << "total cost is: " << (float)price << endl;
    cout << "Enter quantity: ";
    cin >> quantity;
    float price2 = total_price(cost, quantity);
    cout << "total cost is: " << (float)price2 << endl;


    return 0;
}
```

Result:



```
Enter cost: 50
total cost is: 53.5
Enter quantity: 2
total cost is: 107
```

HWQ3 Explanation: This program uses a function called total price to calculate the total price after the user inputs the cost of an item and quantity. It uses a float as the output and inputs of the baseprice, quantity which defaults to 1 if nothing is entered. And a tax rate of 7% is applied to the order, it prints the total cost.

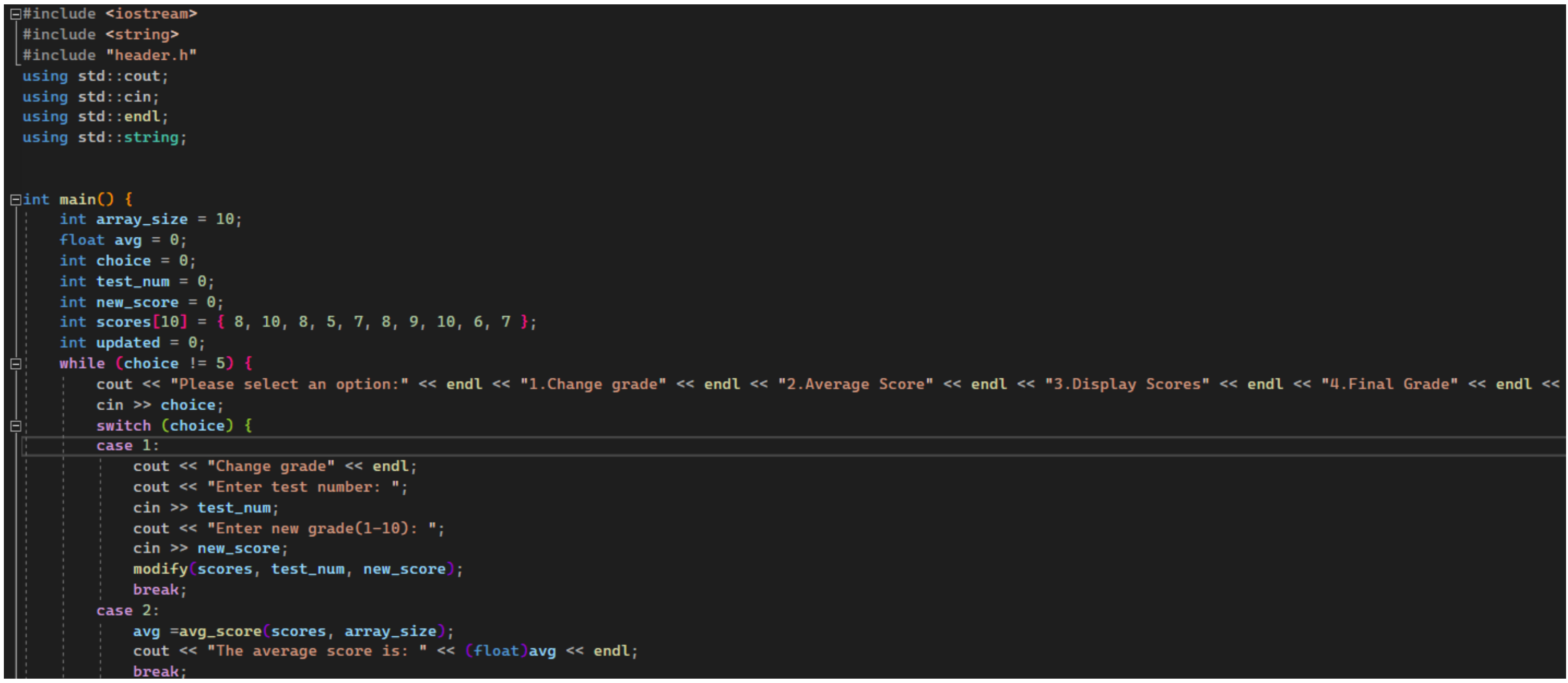
Score	Grade
Greater than 93	A
85-93	B
70 - 85	C
60 - 70	D
Below 60	F

Question 1:

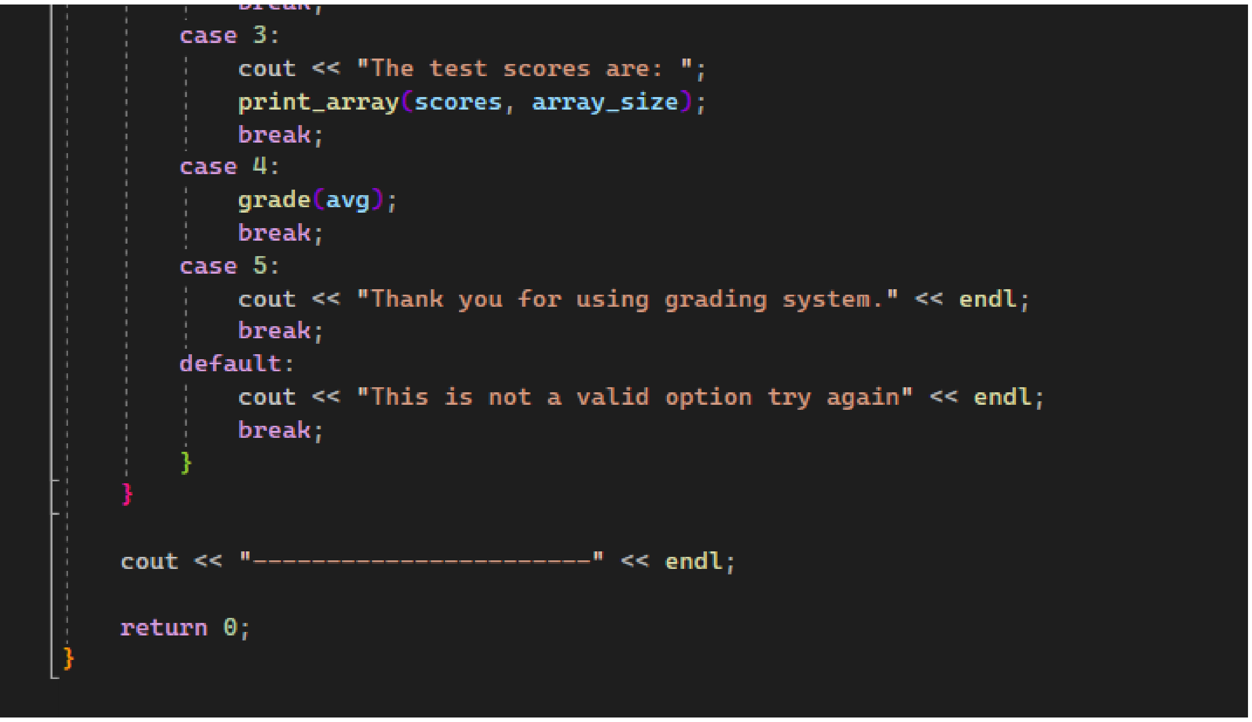
Assumptions: Assuming initial score of a students are

- 8, 10, 8, 5, 7, 8, 9, 10, 6, 7
- There are 10 tests, maximum score of each test is 10

Main:

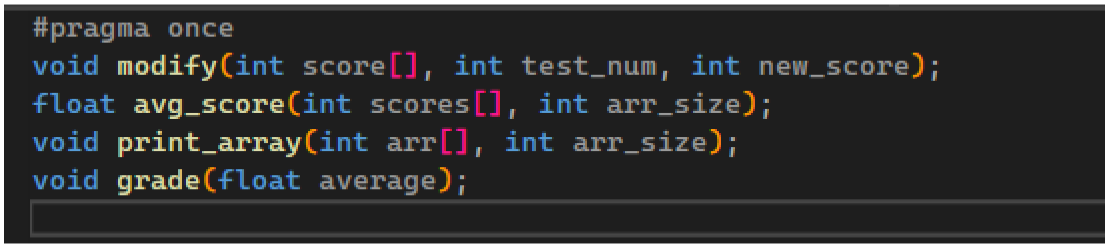


```cpp
#include <iostream>
#include <string>
#include "header.h"
using std::cout;
using std::cin;
using std::endl;
using std::string;


int main() {
    int array_size = 10;
    float avg = 0;
    int choice = 0;
    int test_num = 0;
    int new_score = 0;
    int scores[10] = { 8, 10, 8, 5, 7, 8, 9, 10, 6, 7 };
    int updated = 0;
    while (choice != 5) {
        cout << "Please select an option:" << endl << "1.Change grade" << endl << "2.Average Score" << endl << "3.Display Scores" << endl << "4.Final Grade" << endl <<
        cin >> choice;
        switch (choice) {
        case 1:
            cout << "Change grade" << endl;
            cout << "Enter test number: ";
            cin >> test_num;
            cout << "Enter new grade(1-10): ";
            cin >> new_score;
            modify(scores, test_num, new_score);
            break;
        case 2:
            avg =avg_score(scores, array_size);
            cout << "The average score is: " << (float)avg << endl;
            break;
```



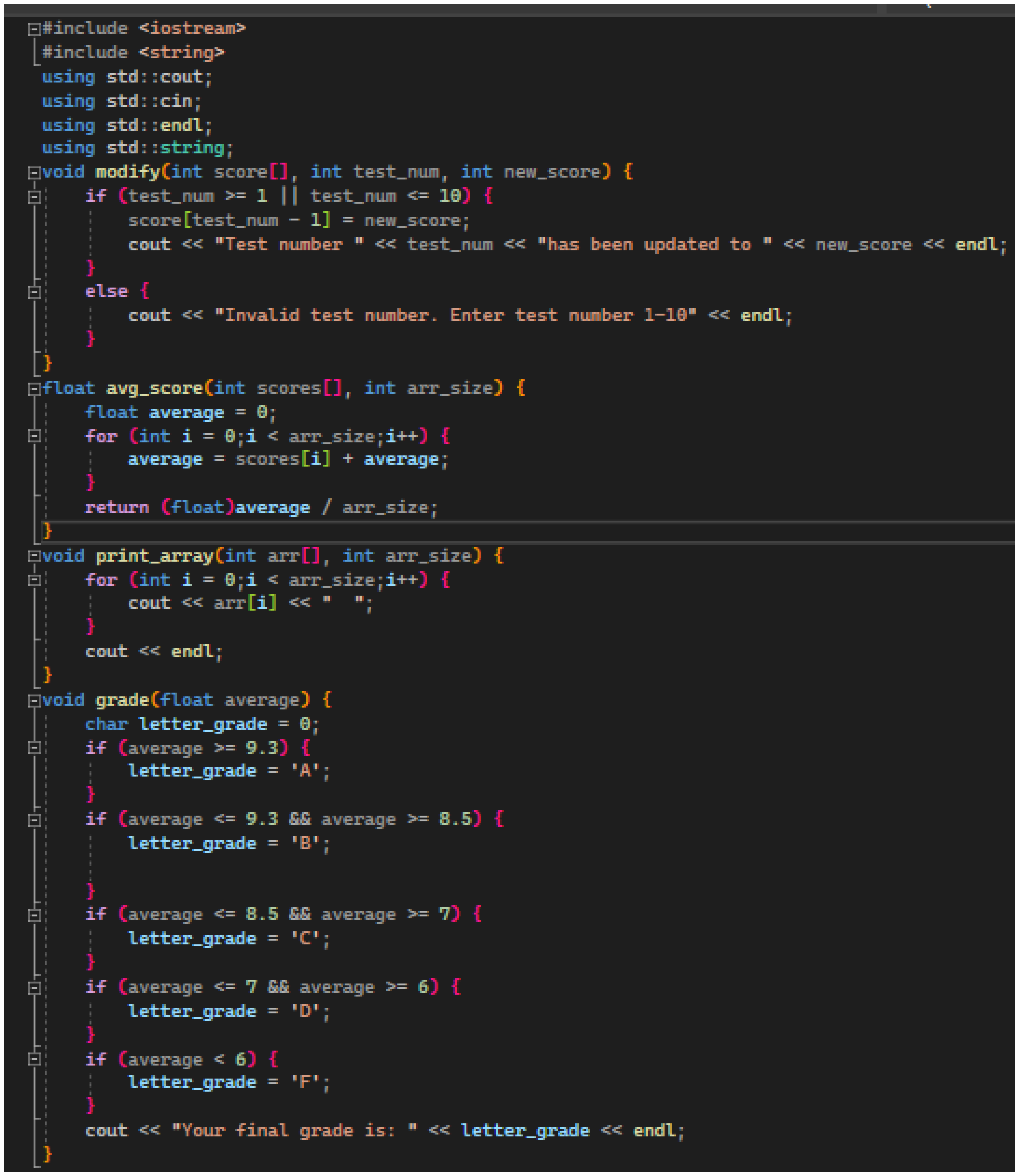
```cpp
        case 3:
            cout << "The test scores are: ";
            print_array(scores, array_size);
            break;
        case 4:
            grade(avg);
            break;
        case 5:
            cout << "Thank you for using grading system." << endl;
            break;
        default:
            cout << "This is not a valid option try again" << endl;
            break;
        }
    }

    cout << "-----------------------" << endl;

    return 0;
}
```

Header:



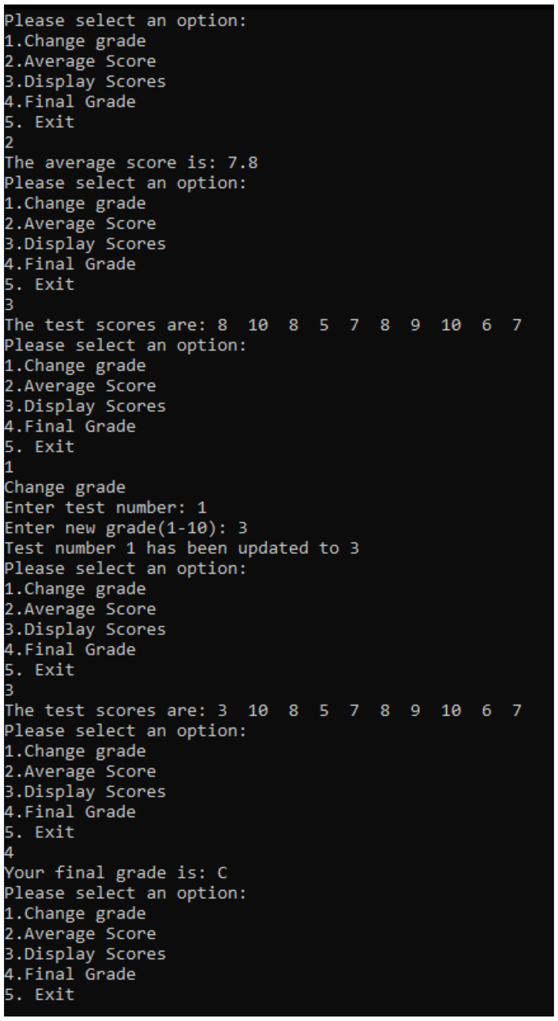
```
#pragma once
void modify(int score[], int test_num, int new_score);
float avg_score(int scores[], int arr_size);
void print_array(int arr[], int arr_size);
void grade(float average);
```

Mycode.cpp:



```
#include <iostream>
#include <string>
using std::cout;
using std::cin;
using std::endl;
using std::string;
void modify(int score[], int test_num, int new_score) {
    if (test_num >= 1 || test_num <= 10) {
        score[test_num - 1] = new_score;
        cout << "Test number " << test_num << "has been updated to " << new_score << endl;
    }
    else {
        cout << "Invalid test number. Enter test number 1-10" << endl;
    }
}
float avg_score(int scores[], int arr_size) {
    float average = 0;
    for (int i = 0;i < arr_size;i++) {
        average = scores[i] + average;
    }
    return (float)average / arr_size;
}
void print_array(int arr[], int arr_size) {
    for (int i = 0;i < arr_size;i++) {
        cout << arr[i] << "  ";
    }
    cout << endl;
}
void grade(float average) {
    char letter_grade = 0;
    if (average >= 9.3) {
        letter_grade = 'A';
    }
    if (average <= 9.3 && average >= 8.5) {
        letter_grade = 'B';

    }
    if (average <= 8.5 && average >= 7) {
        letter_grade = 'C';
    }
    if (average <= 7 && average >= 6) {
        letter_grade = 'D';
    }
    if (average < 6) {
        letter_grade = 'F';
    }
    cout << "Your final grade is: " << letter_grade << endl;
}
```

Results:



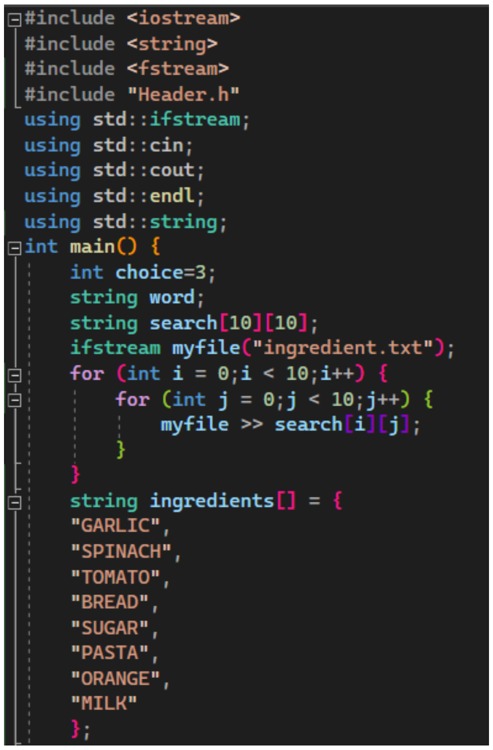
```
Please select an option:
1.Change grade
2.Average Score
3.Display Scores
4.Final Grade
5. Exit
2
The average score is: 7.8
Please select an option:
1.Change grade
2.Average Score
3.Display Scores
4.Final Grade
5. Exit
3
The test scores are: 8  10  8  5  7  8  9  10  6  7
Please select an option:
1.Change grade
2.Average Score
3.Display Scores
4.Final Grade
5. Exit
1
Change grade
Enter test number: 1
Enter new grade(1-10): 3
Test number 1 has been updated to 3
Please select an option:
1.Change grade
2.Average Score
3.Display Scores
4.Final Grade
5. Exit
3
The test scores are: 3  10  8  5  7  8  9  10  6  7
Please select an option:
1.Change grade
2.Average Score
3.Display Scores
4.Final Grade
5. Exit
4
Your final grade is: C
Please select an option:
1.Change grade
2.Average Score
3.Display Scores
4.Final Grade
5. Exit
```

Question 1 Explainaton: I started by making the main function menu. This allows the user to select an option to change grade, calcuate average score, display scores, display final grade, or exit when selected 5 in while loop. The array is determed by assumption.For each of the cases the function is called, in the mycode.cpp I made the functions, modify changes the grade of the array number so if the test is between 1-10 it changes the test score for the array index. Average score uses a float and a for loop to add all the items in the array and divide by array size and return average. Print array uses a simple for loop to print the array. Grade function has no output and simply uses if statements to place the following grade into a letter grade catergory and prints in console.

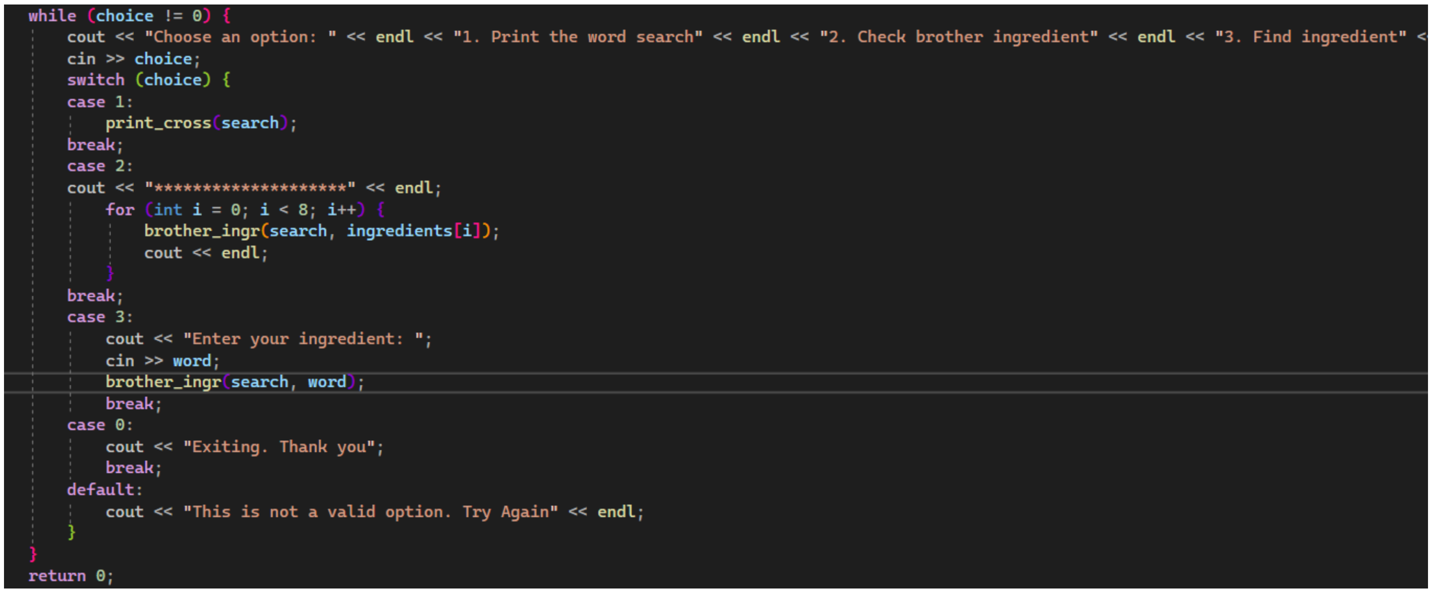
Questions 2:

Assumptions: ingredients.txt, ingredients on list: Garlic, Spinach, Tomato, Bread, Sugar, Pasta, Orange. Milk

Main.cpp:

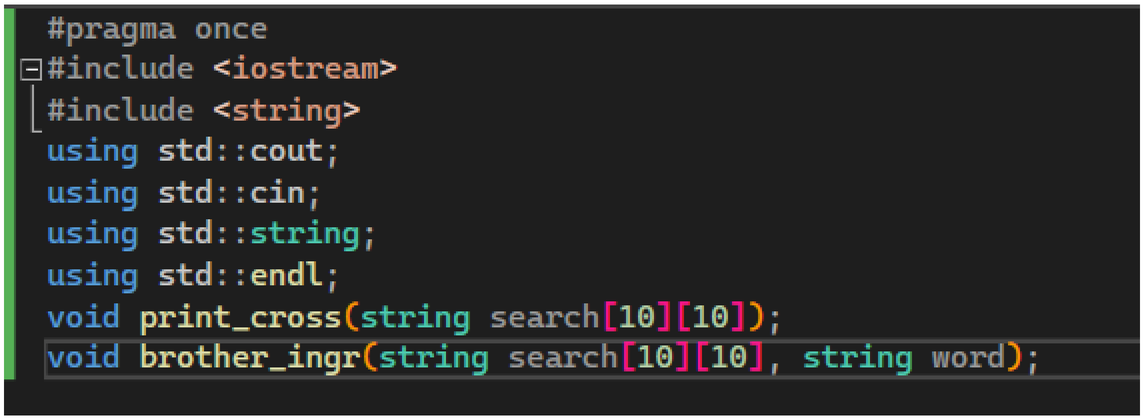


```cpp
#include <iostream>
#include <string>
#include <fstream>
#include "Header.h"
using std::ifstream;
using std::cin;
using std::cout;
using std::endl;
using std::string;
int main() {
    int choice=3;
    string word;
    string search[10][10];
    ifstream myfile("ingredient.txt");
    for (int i = 0;i < 10;i++) {
        for (int j = 0;j < 10;j++) {
            myfile >> search[i][j];
        }
    }
    string ingredients[] = {
    "GARLIC",
    "SPINACH",
    "TOMATO",
    "BREAD",
    "SUGAR",
    "PASTA",
    "ORANGE",
    "MILK"
    };
```



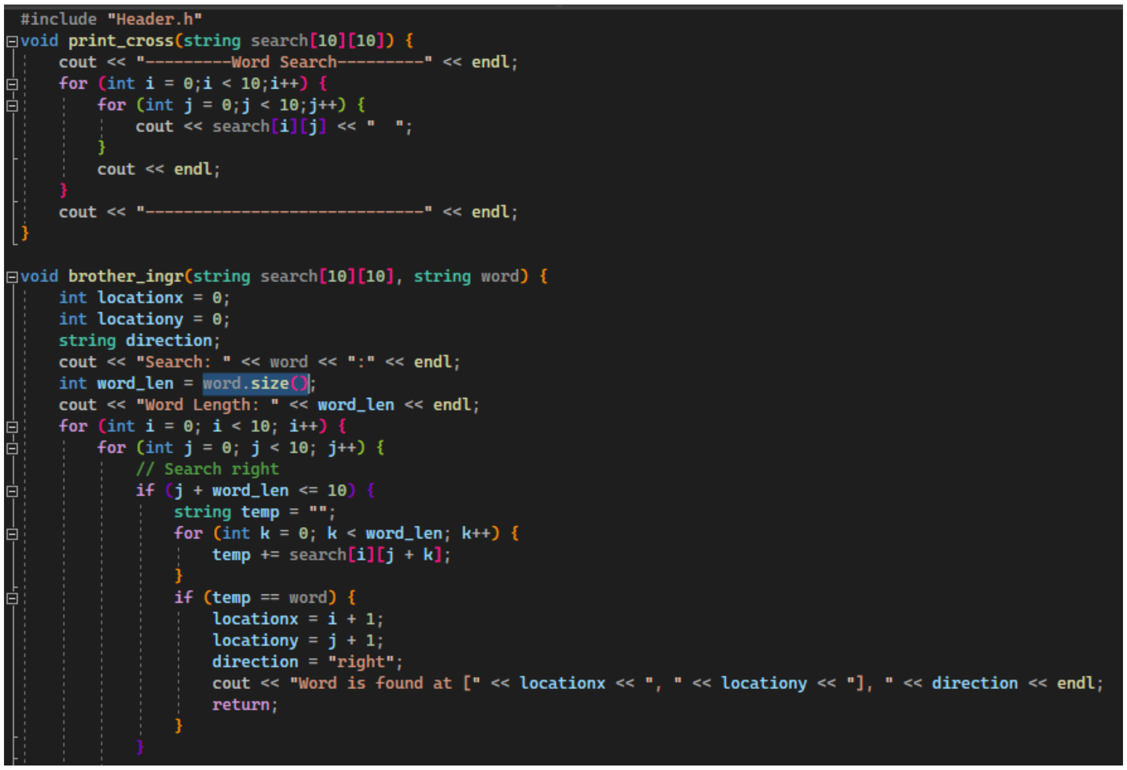
```cpp
while (choice != 0) {
    cout << "Choose an option: " << endl << "1. Print the word search" << endl << "2. Check brother ingredient" << endl << "3. Find ingredient" <
    cin >> choice;
    switch (choice) {
    case 1:
        print_cross(search);
    break;
    case 2:
    cout << "********************" << endl;
        for (int i = 0; i < 8; i++) {
            brother_ingr(search, ingredients[i]);
            cout << endl;
        }
    break;
    case 3:
        cout << "Enter your ingredient: ";
        cin >> word;
        brother_ingr(search, word);
        break;
    case 0:
        cout << "Exiting. Thank you";
        break;
    default:
        cout << "This is not a valid option. Try Again" << endl;
    }
}
return 0;
```

header.h:

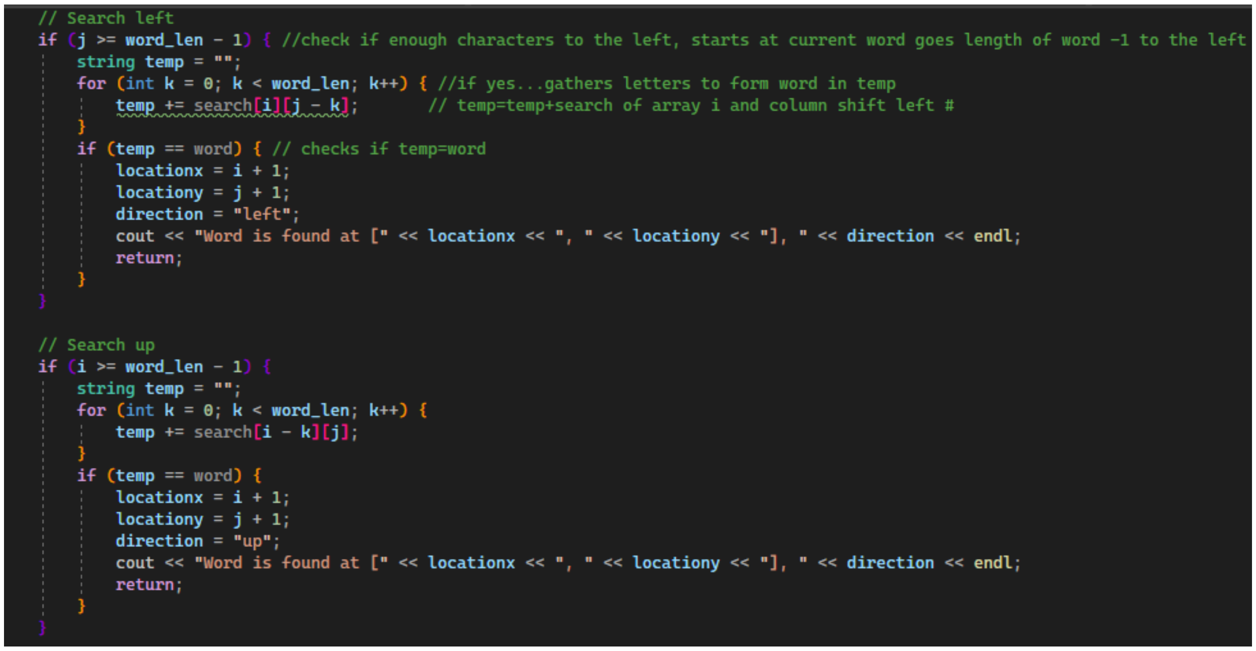


```cpp
#pragma once
#include <iostream>
#include <string>
using std::cout;
using std::cin;
using std::string;
using std::endl;
void print_cross(string search[10][10]);
void brother_ingr(string search[10][10], string word);
```

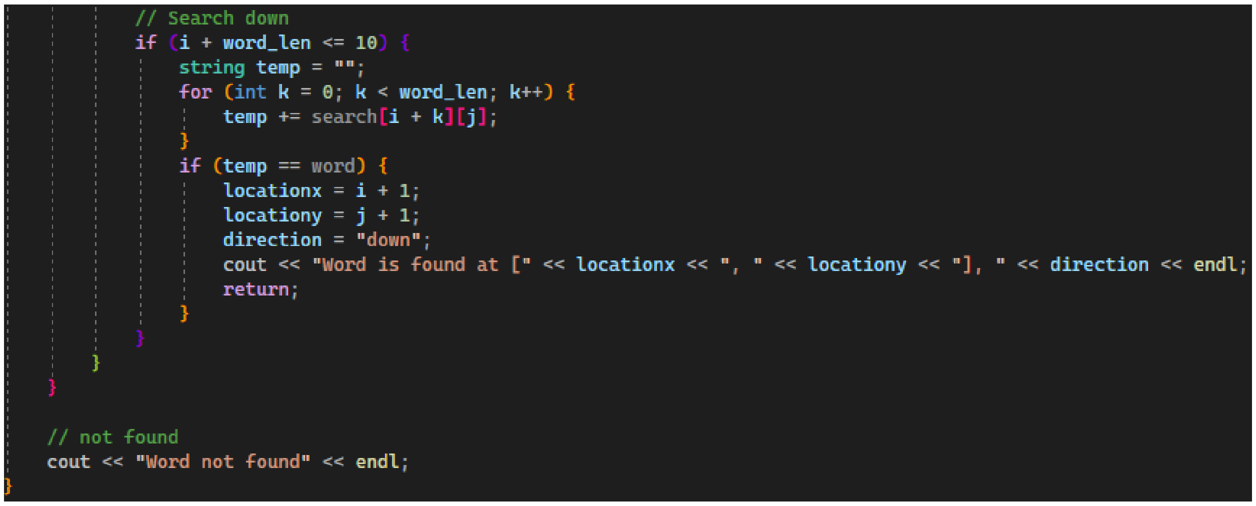
Mycode.cpp:



```cpp
#include "Header.h"
void print_cross(string search[10][10]) {
    cout << "---------Word Search---------" << endl;
    for (int i = 0;i < 10;i++) {
        for (int j = 0;j < 10;j++) {
            cout << search[i][j] << "  ";
        }
        cout << endl;
    }
    cout << "-----------------------------" << endl;
}

void brother_ingr(string search[10][10], string word) {
    int locationx = 0;
    int locationy = 0;
    string direction;
    cout << "Search: " << word << ":" << endl;
    int word_len = word.size();
    cout << "Word Length: " << word_len << endl;
    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
            // Search right
            if (j + word_len <= 10) {
                string temp = "";
                for (int k = 0; k < word_len; k++) {
                    temp += search[i][j + k];
                }
                if (temp == word) {
                    locationx = i + 1;
                    locationy = j + 1;
                    direction = "right";
                    cout << "Word is found at [" << locationx << ", " << locationy << "], " << direction << endl;
                    return;
                }
            }
```



```
// Search left
if (j >= word_len - 1) { //check if enough characters to the left, starts at current word goes length of word -1 to the left
    string temp = "";
    for (int k = 0; k < word_len; k++) { //if yes...gathers letters to form word in temp
        temp += search[i][j - k];       // temp=temp+search of array i and column shift left #
    }
    if (temp == word) { // checks if temp=word
        locationx = i + 1;
        locationy = j + 1;
        direction = "left";
        cout << "Word is found at [" << locationx << ", " << locationy << "], " << direction << endl;
        return;
    }
}

// Search up
if (i >= word_len - 1) {
    string temp = "";
    for (int k = 0; k < word_len; k++) {
        temp += search[i - k][j];
    }
    if (temp == word) {
        locationx = i + 1;
        locationy = j + 1;
        direction = "up";
        cout << "Word is found at [" << locationx << ", " << locationy << "], " << direction << endl;
        return;
    }
}
```



```
            // Search down
            if (i + word_len <= 10) {
                string temp = "";
                for (int k = 0; k < word_len; k++) {
                    temp += search[i + k][j];
                }
                if (temp == word) {
                    locationx = i + 1;
                    locationy = j + 1;
                    direction = "down";
                    cout << "Word is found at [" << locationx << ", " << locationy << "], " << direction << endl;
                    return;
                }
            }
        }
    }

    // not found
    cout << "Word not found" << endl;
}
```

Question 2 explanation: I started off by making the main, I used ifstream with the given text file to load the ingredients into the array with a for loop. Then I made a array of the ingredients on the brothers list which will be used in case 2. The loop continues as long as 0 isnt selected. In mycode.cpp I made the functions that will be called in the main. The printcross function uses a for loop to print the letters in the given crossword and prints it to the user. The next function was complex, but it starts of with the x and y coordinates that will be checked. It has the user input the word searched for, and uses size command to print number of letters in the word. By using a for loop < 10 for the rows and another imbedded for loop < 10 for the columns, the value of I in the coordinate plane is used in the following searches. The way the search performs in each direction is by: checking there is enough characters in that direction of the current position. If there Is, create a temp string by taking characters from that direction. It will then compare the temp with the target word, a if loop is used to compare, if it matches it will return immediately and print the position and direction it was on by printing i and j. If it is not found in one direction it proceeds to the next and if it not found at all it prints word not found.

For case 2, it uses a for loop which refers to the array of strings which are on the list, the for loop enters the ingredient into the function instead of a user input which is used for vast simplification.