Tatyana Vlaskin

Final Project

For the final project I took few programs from the book, and connected them to a farm theme. I’ve decided that was the easiest way to meet all or most of the requirements of the project. I’ve done the following problems from the book with few modifications.

**CHAPTER 5: ARRAYS**

15. Traditional password entry schemes are susceptible to “shoulder surfing” in which

an attacker watches an unsuspecting user enter their password or PIN number and

uses it later to gain access to the account. One way to combat this problem is with

a randomized challenge-response system. In these systems, the user enters different

information every time based on a secret in response to a randomly generated

challenge. Consider the following scheme in which the password consists of a five-digit

PIN number (00000 to 99999). Each digit is assigned a random number that is

1, 2, or 3. The user enters the random numbers that correspond to their PIN

instead of their actual PIN numbers.

For example, consider an actual PIN number of 12345. To authenticate, the user

would be presented with a screen such as

PIN: 0 1 2 3 4 5 6 7 8 9

NUM: 3 2 3 1 1 3 2 2 1 3

The user would enter 23113 instead of 12345. This does not divulge the password

even if an attacker intercepts the entry because 23113 could correspond to other

PIN numbers, such as 69440 or 70439. The next time the user logs in, a different

sequence of random numbers would be generated, such as

PIN: 0 1 2 3 4 5 6 7 8 9

NUM: 1 1 2 3 1 2 2 3 3 3

Your program should simulate the authentication process. Store an actual PIN

number in your program. The program should use an array to assign random

numbers to the digits from 0 to 9. Output the random digits to the screen, input

the response from the user, and output whether or not the user’s response correctly

matches the PIN number.

**CHAPTER 6: STRUCTURES AND CLASSES**

12. Your Community Supported Agriculture (CSA) farm delivers a box of fresh fruits

and vegetables to your house once a week. For this Programming Project, define the

class BoxOfProduce that contains exactly three bundles of fruits or vegetables. You

can represent the fruits or vegetables as an array of type string . Add accessor and

mutator functions to get or set the fruits or vegetables stored in the array. Also write

an output function that displays the complete contents of the box on the console.

Next, write a main function that creates a BoxOfProduce with three items

randomly selected from this list:

Broccoli

Tomato

Kiwi

Kale

Tomatillo

This list should be stored in a text file that is read in by your program. For now

you can assume that the list contains exactly five types of fruits or vegetables.

Do not worry if your program randomly selects duplicate produce for the three

items. Next, the main function should display the contents of the box and allow

the user to substitute any one of the five possible fruits or vegetables for any of the

fruits or vegetables selected for the box. After the user is done with substitutions

output the final contents of the box to be delivered.

**CHAPTER 5: ARRAYS**

**THIS PROGRAM WILL BE SLIGHLY MODIFIED**

14. The user will be asked to rate baskets that were delivered to them before. I am talking about baskets from the program **CHAPTER 6: STRUCTURES AND CLASSES** above. First the user will be asked how many baskets they want to grade. This will let me use dynamic array. Once the entry is accepted, the used will be asked to enter rating for each product in the basket. The ratings range from 1 (terrible) to 5 (excellent) and 0 – product was not in the basket. Once the user is done reviewing the products, they will be able to see their review in a table:

Tomatillo Broccoli Tomato Kiwi Kale

1 1 5 2 2 3

2 1 1 1 1 1

At the same time, the user will be able to see the average rating for each product.

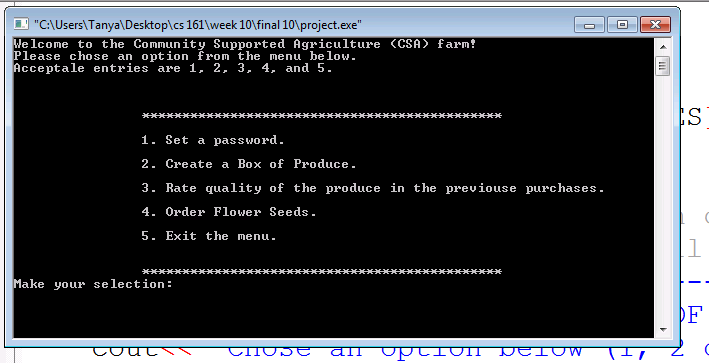
**FINALLY:**

Ill have another program that will present list of flower seed along with the prices to the user. The user will be asked what they want to buy. The user will be able to add one item at a time and once the user does not want to add any more items, a check will be presented to the user.

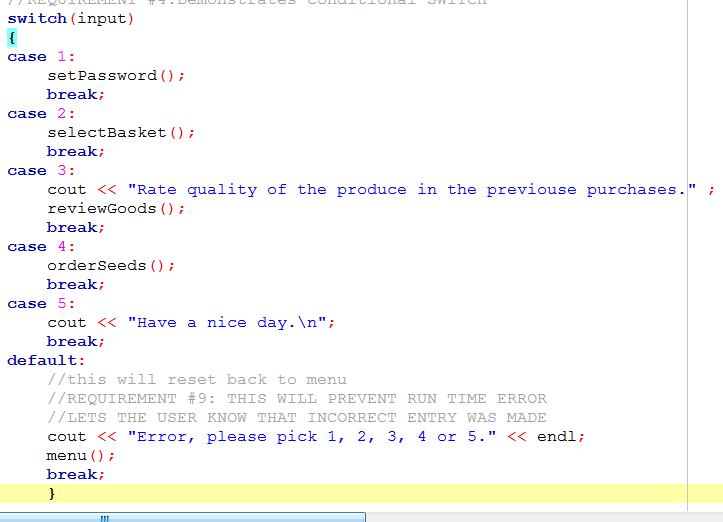
IMPLEMENTATION AND DESIGN:

Per discussion board and per the requirements of the project, this was done simultaneously.

The user will be presented with the following message and menu:



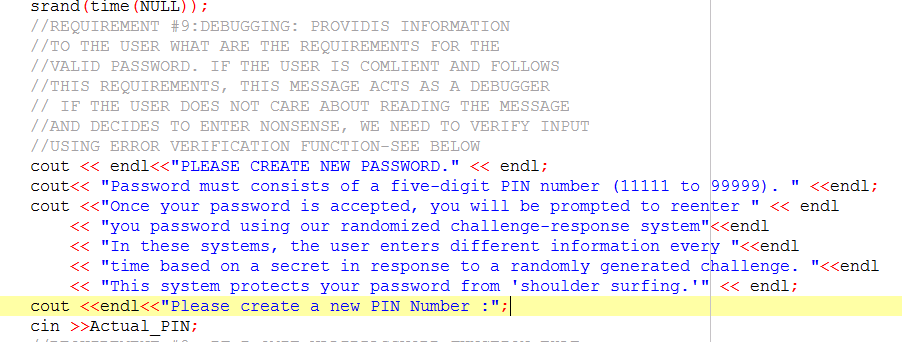
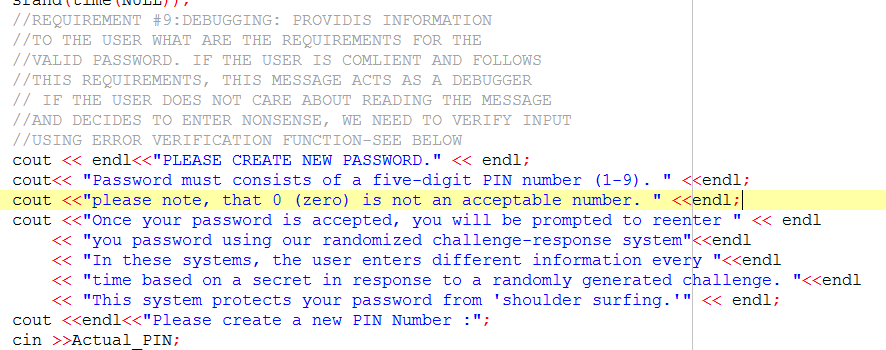
The input of the user will be verified and if the user made invalid input, the user will be asked to make a different entry. One the entry is accepted, the following switch statement will be entered, to implement the corresponding function:



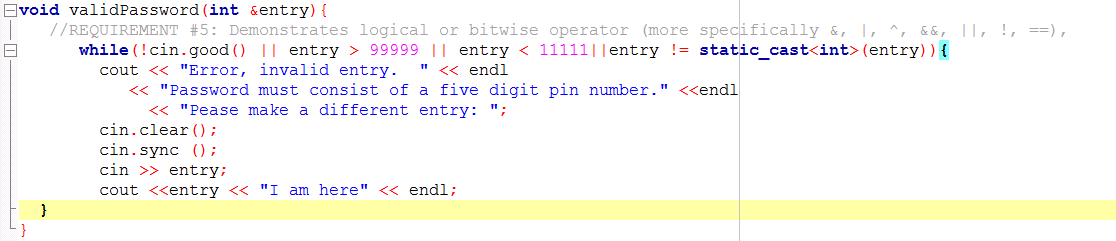
Lets take a look at each subsection closely:

SETPASSWORD:

The user will be provided with the following message:

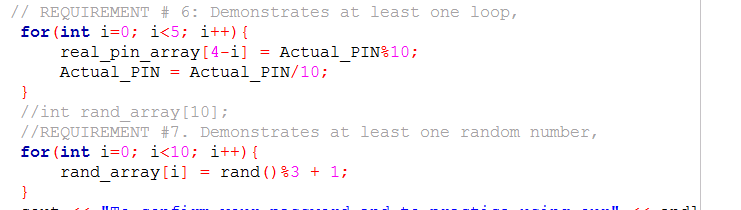


Once the password is entered, it will be checked for validity with the following code:

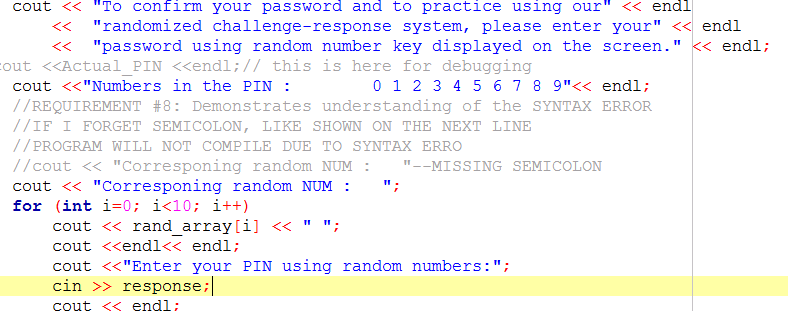


Letters, doubles, and passwords that are not between 11111-99999 will be rejected. Initially, I was planning to accept passwords like 00000, however, I was having a hard time with zeroes and making sure that password is 5 characters long, so 0 will not be acceptable number for password.

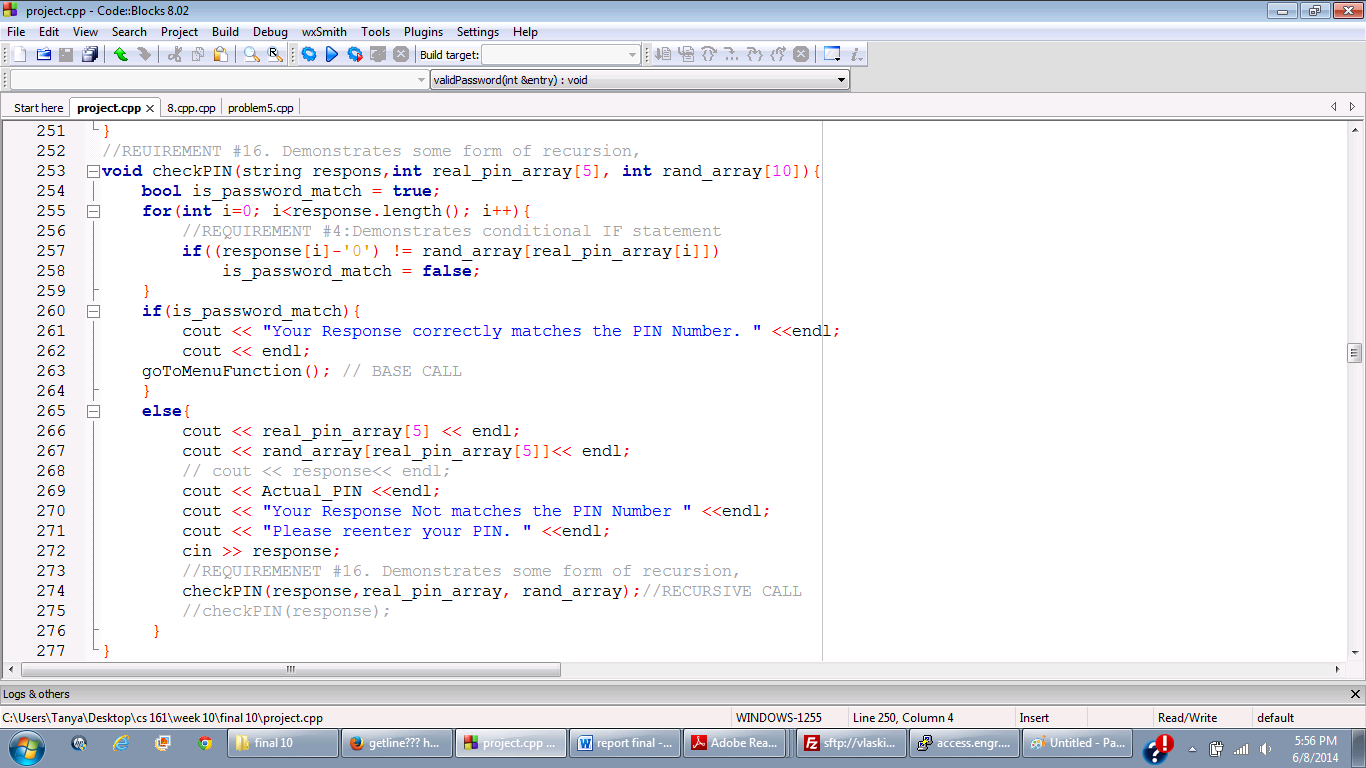
Once the password is accepted, 2 different arrays will be generated. One array will be filled with the password that was entered by the user and the second one will be filled with randomly generated numbers ( I will use only numbers 1-3):



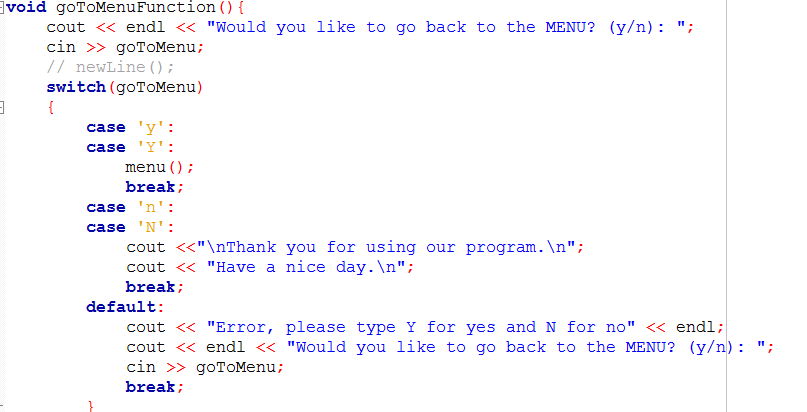
Next, the user will be asked to enter a password using randomized challenged system:



Passwords will be checked using the: checkPIN(response,real\_pin\_array, rand\_array); function.



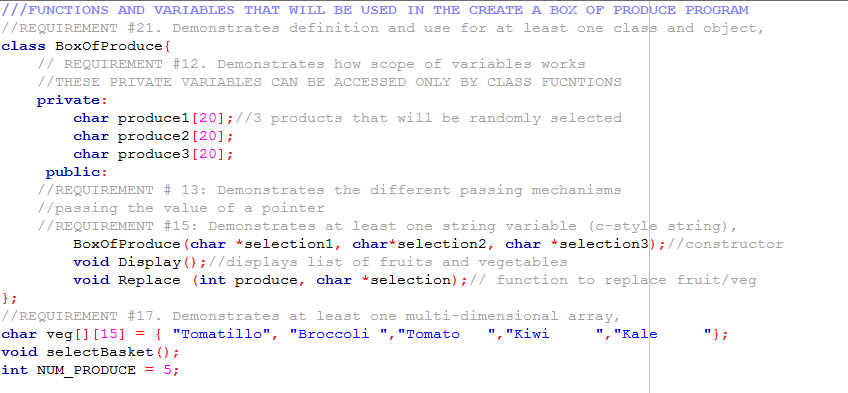
This function will check if the entry that was made by the randomized challenge system matches the password that was entered. If the password does not match, the user will be asked to reenter the password. If the password match, the user will get a message that the password matches and will be asked if they want to go to the main menu:



Please note that this function is not 100% perfect, so please entry y or n. I did not have time to check it thoroughly and did not have time to improve it to check for valid entry, so please enter only y or n.

**CREATE BOX OF PRODUCE OPTION:**

List of functions and variables used in this program:



I decided to demonstrate my understanding of classes, and multidimensional arrays.

First step is to generate basket randomly. The basket will contain one of the products listed in the array, duplicate products will be accepted (example: tomato, kiwi and tomato) will be an acceptable basket.

BoxOfProduce box(veg[rand()%5], veg[rand()%5], veg[rand()%5] );

Please note that this is a constructor with parameters passed as references.

BoxOfProduce::BoxOfProduce(char \*selection1, char\*selection2, char \*selection3){

strcpy(produce1, selection1);

strcpy(produce2, selection2);

strcpy(produce3, selection3);

}

Next the contents of the basket will be displayed to the user and the user will be asked if they want to replace product in the basket.

void BoxOfProduce::Display(){

cout<<"1. "<<produce1<<endl;

cout<<"2. "<<produce2<<endl;

cout<<"3. "<<produce3<<endl;

}

If the user decides that they want to replace something in the basket the following function will be called:

void BoxOfProduce::Replace (int choice, char \*selection){

switch(choice){

case 1: strcpy(produce1, selection); break;

case 2: strcpy(produce2, selection); break;

case 3: strcpy(produce3, selection); break;

}

}

The function will be called in the following way:

cout<<"\nChose the index of the frut or vegetable that you want to choose : ";

getline(cin,rep\_with1);

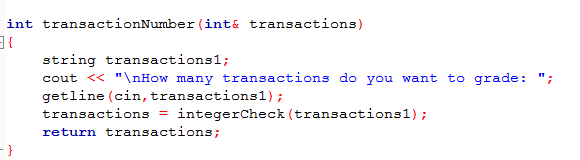
rep\_with = integerCheck(rep\_with1);

box.Replace(rep, veg[rep\_with-1]);

If the user does not want to replace any good, they will be asked if they want to go back to the main menu.

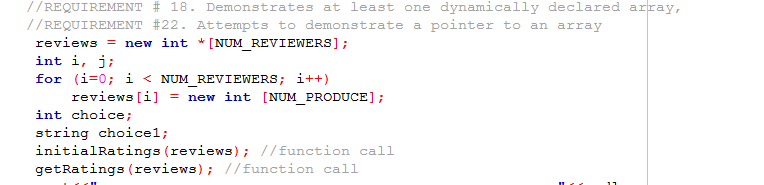
**RATE QUALITY OF THE PRODUCE IN THE PREVIOUS TRANSACTIONS:**

First user is asked how many transactions they want to grade:



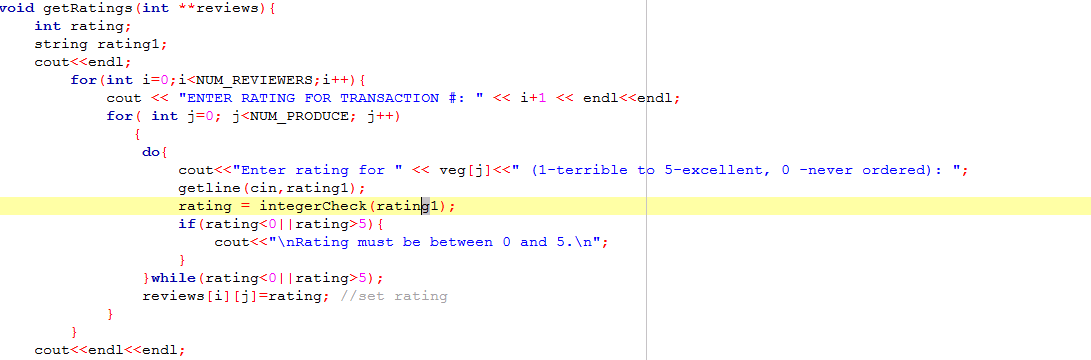
Next a dynamic array is generated. This array will hold review of each product in each transaction.

transactionNumber(NUM\_REVIEWERS);

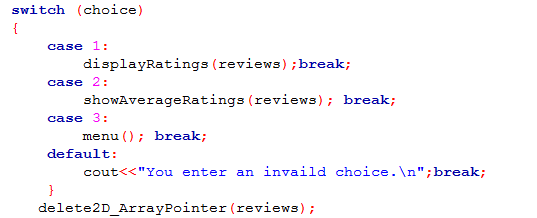


Next step is initialize rating to 0, by using initialRatings function.

Next is to ask the user to fill in the array by inputting their rating scores.

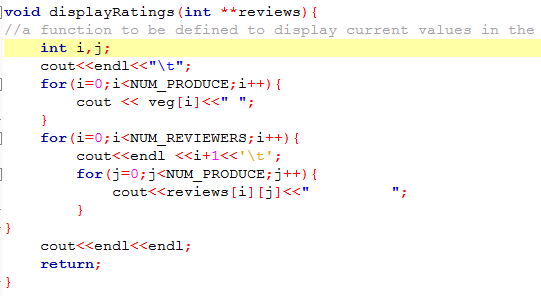


One the user makes their entries; they are asked what they want to do with the entries using the following menu:



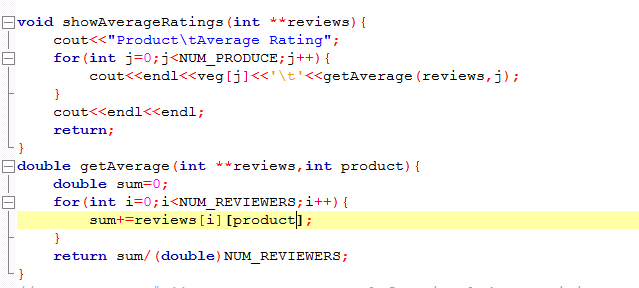
I think these functions are self-explanatory.

Display rating is a simple functions that takes 2D array and displays it to the user:

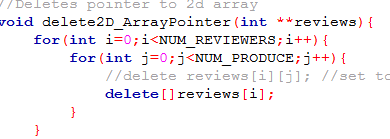


Show average rating is a function that shows average rating for each product in the basket.

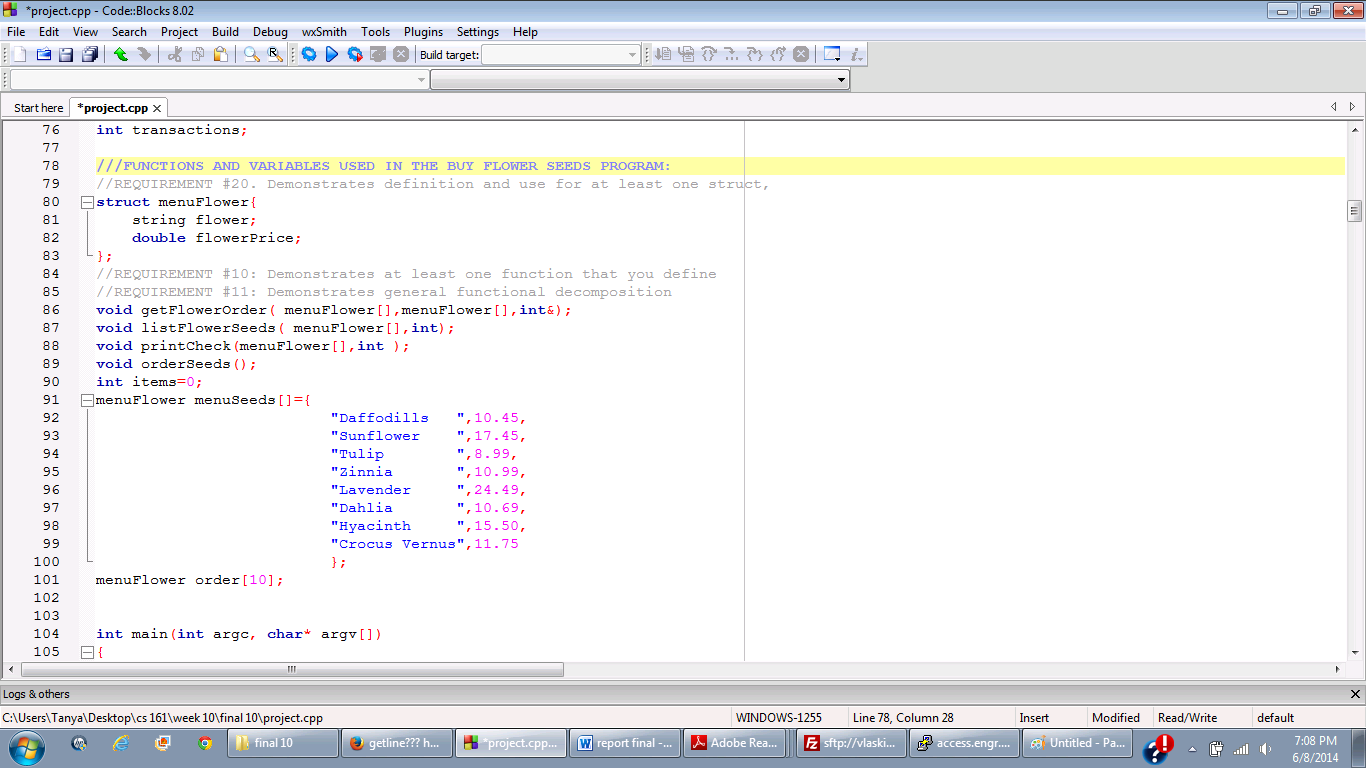
This function uses getAverage function calculate average rating.



Another thing that I almost forgot do to is to delete pointers to 2D dynamic array. I wrote the following function. I hope it does the job:

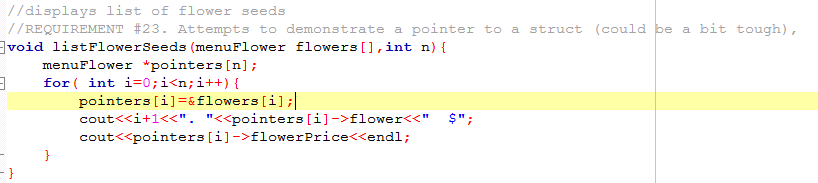


**ORDER SEEDS:**

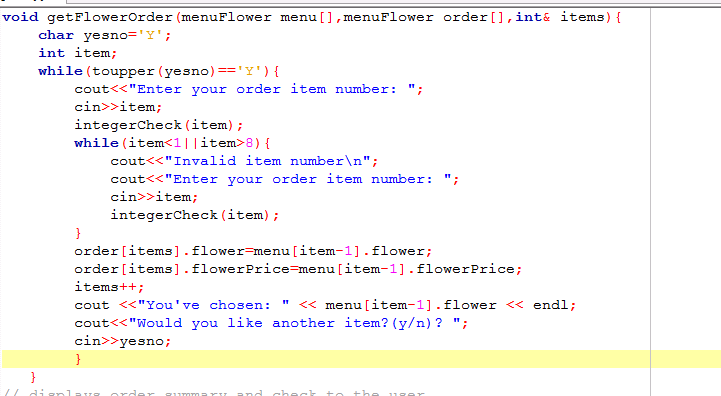


Ill use structures for this program.

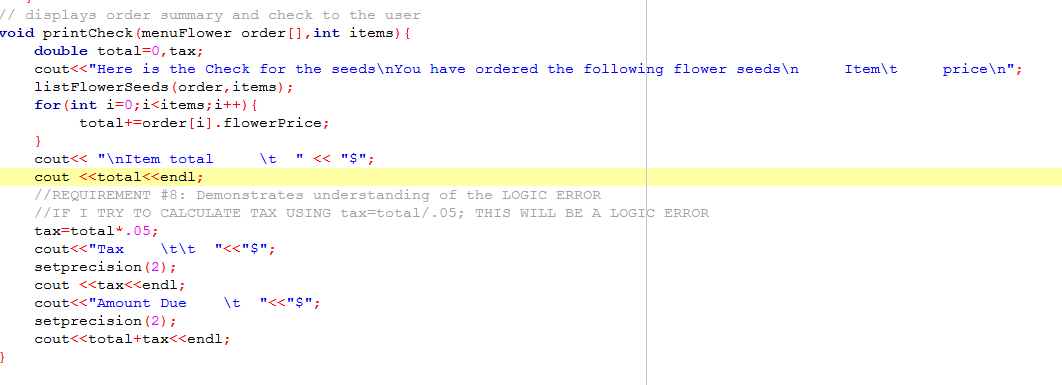
First user will be presented with the list of flower seeds that they can buy:



Next the user will be asked what they want to order:



If the user does not want to order more items, a check will be displayed:



Precision will be placed to 2 using the following function:

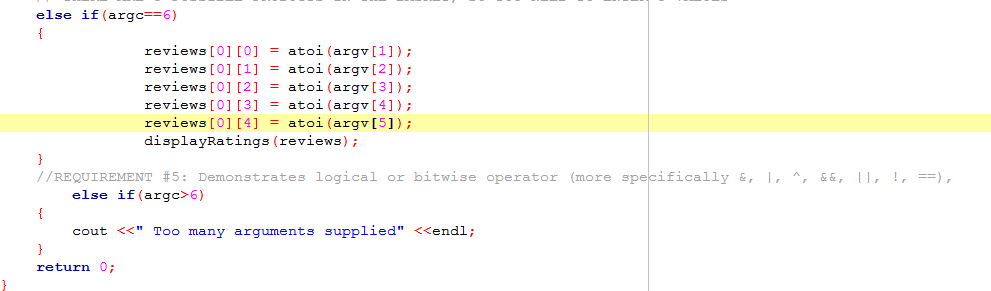
void setprecision(int number){

cout.setf(ios::fixed);

cout.setf(ios::showpoint);

cout.precision(number);

}

It seems that I met most of the requirements of the project; the last one is command line input. My plan was to use displayRating function for 1 transaction by using the following code. 

The idea was that if the user enters 5 numbers, one for each product, I should be able to use displayRating functions to see the ratings that were entered. However, I keep getting Segmentation fault (core dumped error.) I do not have time to troubleshoot it right now, so I’ll leave it as it is. I have not met requirement 19. I do not even understand why anyone would prefer command line over regular cin/cout. You do not even know what you are doing/entering in command line, while when you use cin/cout you can follow directions displayed on the screen.

THANKS FOR GRADING MY ASSIGNMENTS.

