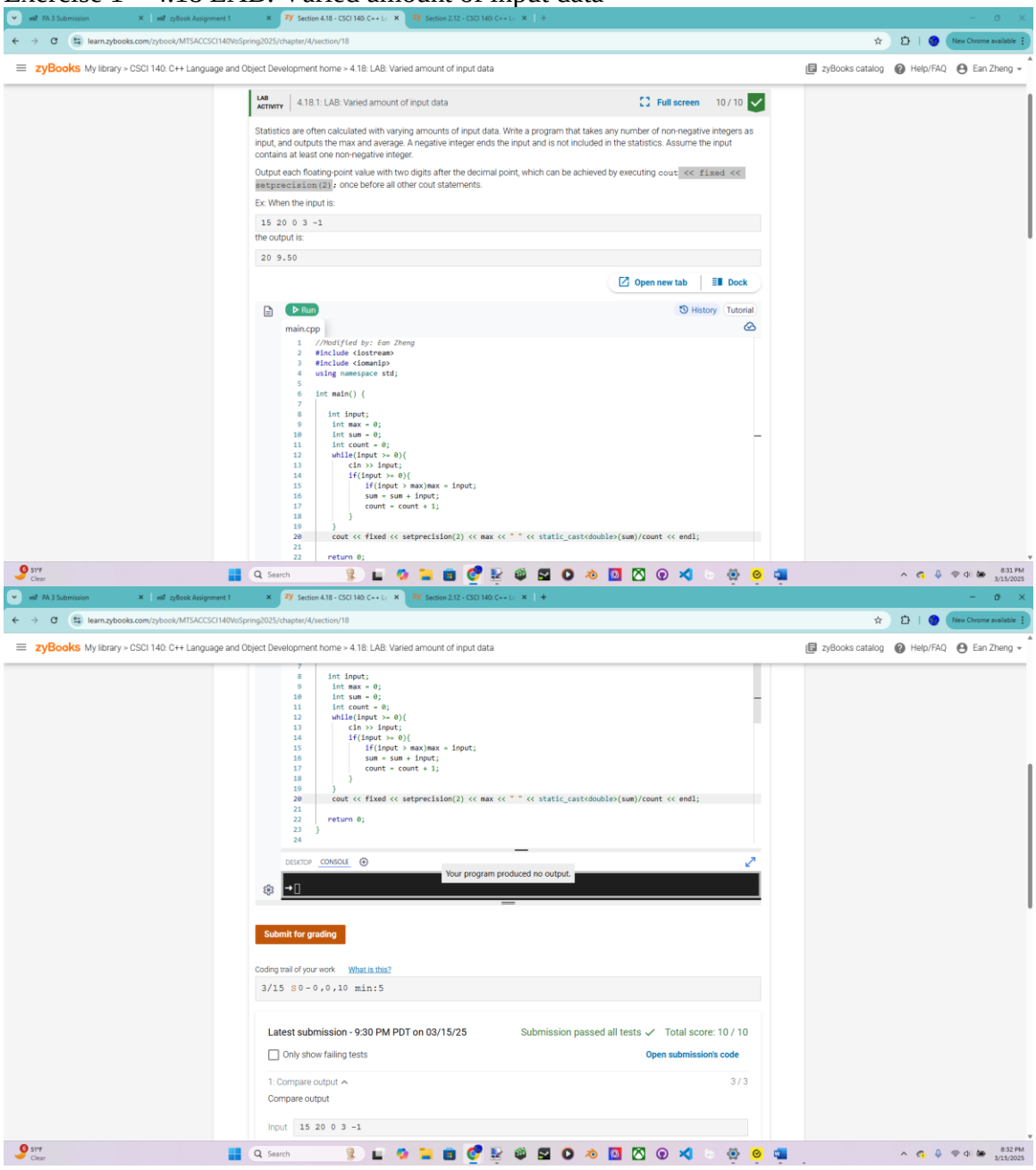
CSCI 140 PA 3 Submission

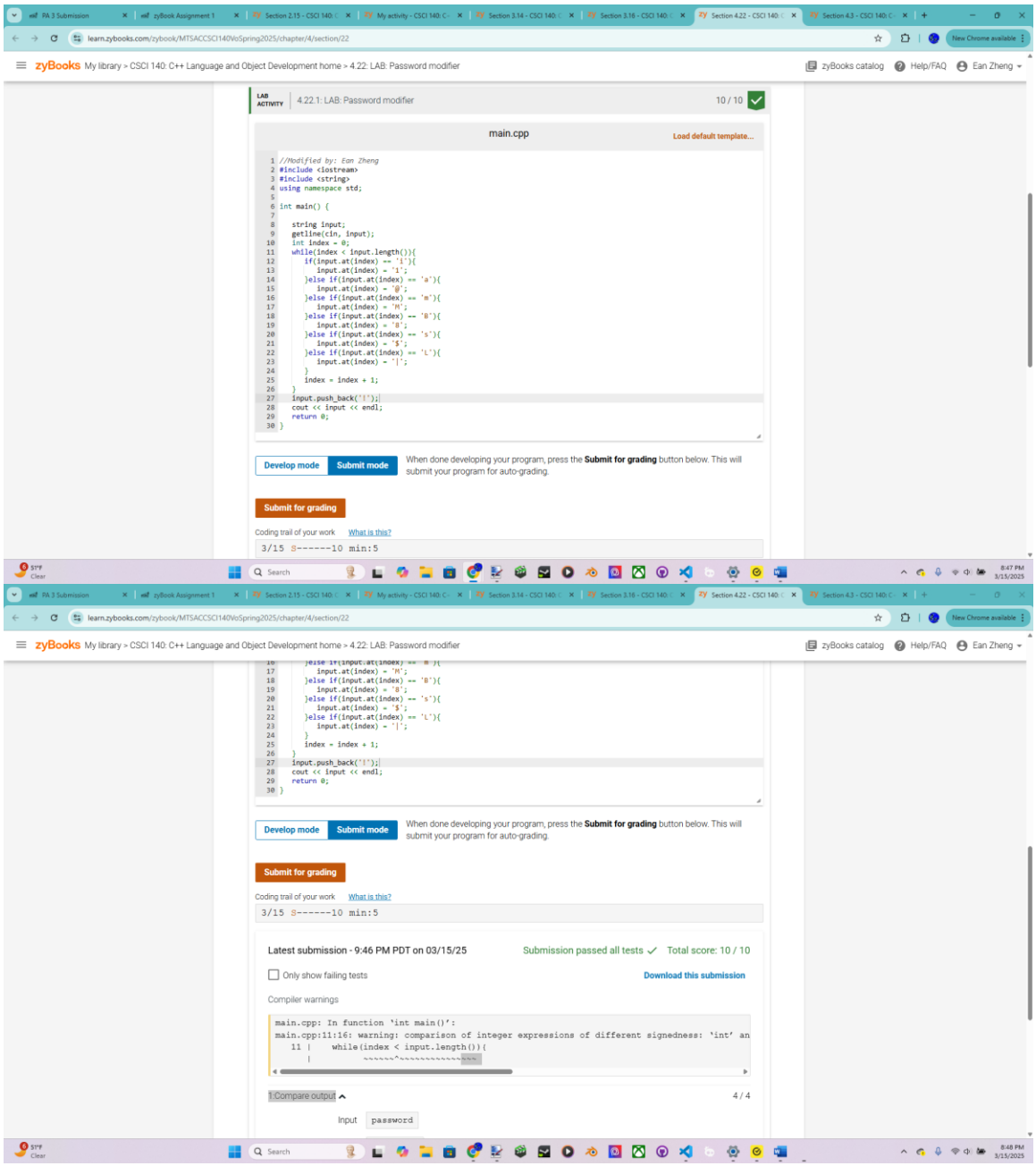
Due Date:3/17/2025 Late (date and time):_____

Name: Ean Zheng

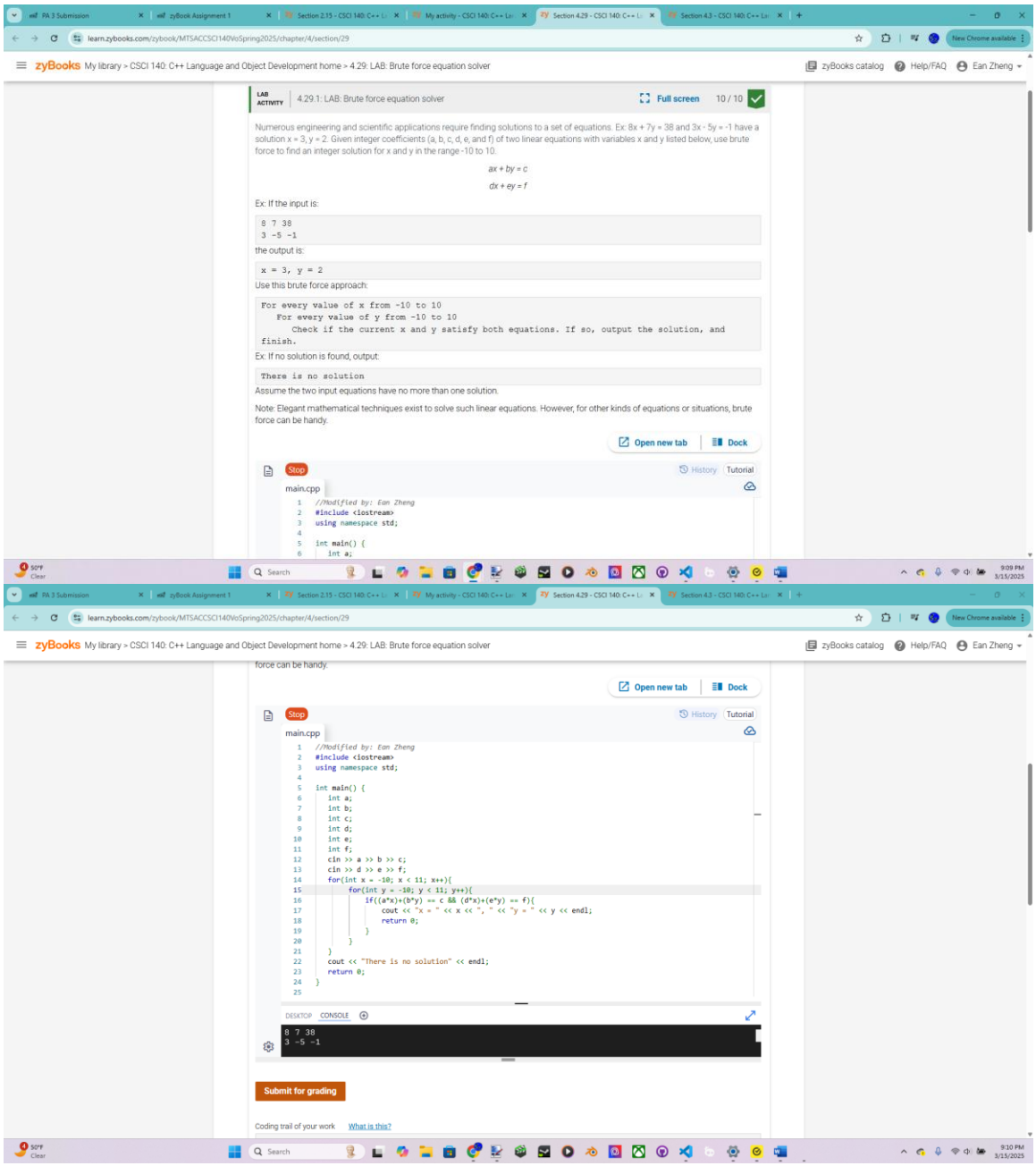
Exercise 1 – 4.18 LAB: Varied amount of input data

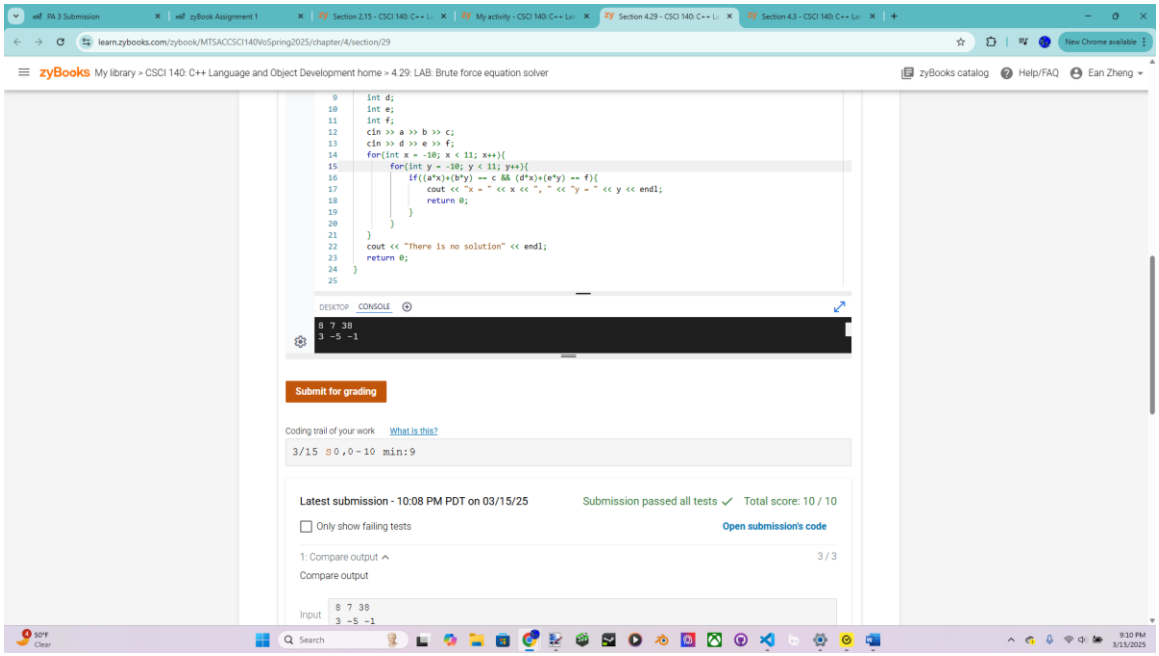


Exercise 2 – 4.22 LAB: Password modifier (only available in Classic mode)



Exercise 3 – 4.29 LAB: Brute force equation solver




zyBooks My library > CSCI 140: C++ Language and Object Development home > 4.29: LAB: Brute force equation solver
zyBooks catalog
Help/FAQ
Ean Zheng
int d;
int e;
int f;
cin >> a >> b >> c;
cin >> d >> e >> f;
for(int x = -10; x < 11; x++){
for(int y = -10; y < 11; y++){
if((a*x)+(b*y) == c && (d*x)+(e*y) == f){
cout << "x = " << x << ", " << "y = " << y << endl;
return 0;
}
}
}
cout << "There is no solution" << endl;
return 0;
}
DESKTOP
CONSOLE
8 7 38
3 -5 -1
Submit for grading
Coding trail of your work
What is this?
3/15 at 0,0 - 10 min: 9
Latest submission - 10:08 PM PDT on 03/15/25
Submission passed all tests ✓
Total score: 10 / 10
Only show failing tests
Open submission's code
1: Compare output
3 / 3
Compare output
Input
8 7 38
3 -5 -1

Exercise 4 – Simple Vending Machine Version 2 – more points for this exercise
Modify "Simple Vending Machine Version 1" from previous PA to support the following new requirements:

- Fill machine with coins via keyboard input before any purchase (quarters, dimes, and nickels).
- Must allow purchases to be repeated until a sentinel value of 0 is entered.
- Keep track the number of dollar bills, coins in the machine and print them at the end.

Note: this new version might not work for all test cases, but that is not a problem (see one test under extra credit option). Pseudocode is not required since we modify it from the previous PA, but feel free to use it. Follow the interface below and you must try the following test case:

Vending Machine Version 2 by [Your Name]
Enter number of quarters, dimes, and nickels --> 2 4 4<Enter>
Number of quarters: 2
Number of dimes : 4
Number of nickels : 4
Machine balance : $1.10
Only one-dollar bill will be accepted.
Only amount between 0 to 100 is accepted.
Enter 0 to stop the program.
Enter a purchase amount [0 - 100] --> 36<Enter>

You entered a purchase amount of 36 cents.
Inserting one-dollar bill.
Processing your purchase ...
Your change of 64 cents is rounded to 65 cents.
Your change of 65 cents is given as:
quarter(s): 2
dime(s): 1
nickel(s): 1
Enter a purchase amount [0 - 100] --> -1<Enter>
You entered a purchase amount of -1 cents.
Invalid amount (outside valid range). Try again.
Enter a purchase amount [0 - 100] --> 35<Enter>
You entered a purchase amount of 35 cents.
Please insert one-dollar bill.
Processing your purchase ...
Insufficient changes!
Your transaction cannot be processed.
Please take back your dollar bill.
Enter a purchase amount [0 - 100] --> 73<Enter>
You entered a purchase amount of 73 cents.
Inserting one-dollar bill.
Processing your purchase ...
Your change of 23 cents is rounded to 25 cents.

Your change of 25 cents is given as:
quarter(s): 0
dime(s): 2
nickel(s): 1
Enter a purchase amount [0 - 100] --> 105<Enter>
You entered a purchase amount of 105 cents.
Invalid amount (outside valid range). Try again.
Enter a purchase amount [0 - 100] --> 0<Enter>
Number of dollars: 2
Number of quarters: 0
Number of dimes : 1
Number of nickels : 2
Machine balance : $2.20

Pseudocode below if applicable:

Input quarters, dimes, and nickels as variables with prompt.

Output coin values, then calculate machine balance and output it with this equation: (quarters * 0.25) + (dimes*0.1)+(nickels*0.05). Set precision output to 2 decimal places

Add new prompts of specification

Add a do while loop around the enter purchase amount process with a condition of the input not equal to 0.

Add an else if statement for the if statement checking invalid range, with condition of input isn't 0, and put code for valid range there.

Redo process of detection: Add new variables in do while loop for given quarters, dimes, and nickels. Increment given coins for every time a coin is needed. Conditions for coins rewrite if given coin variables are not equal to normal coin variables. If processing successful, subtract given coin variables from normal coin variables, and output the given coin variables.

Update invalid amount message and add new messages for processing. Update change process fail message.

Add dollars variable initialized as 0. Increment it if change process is successful.

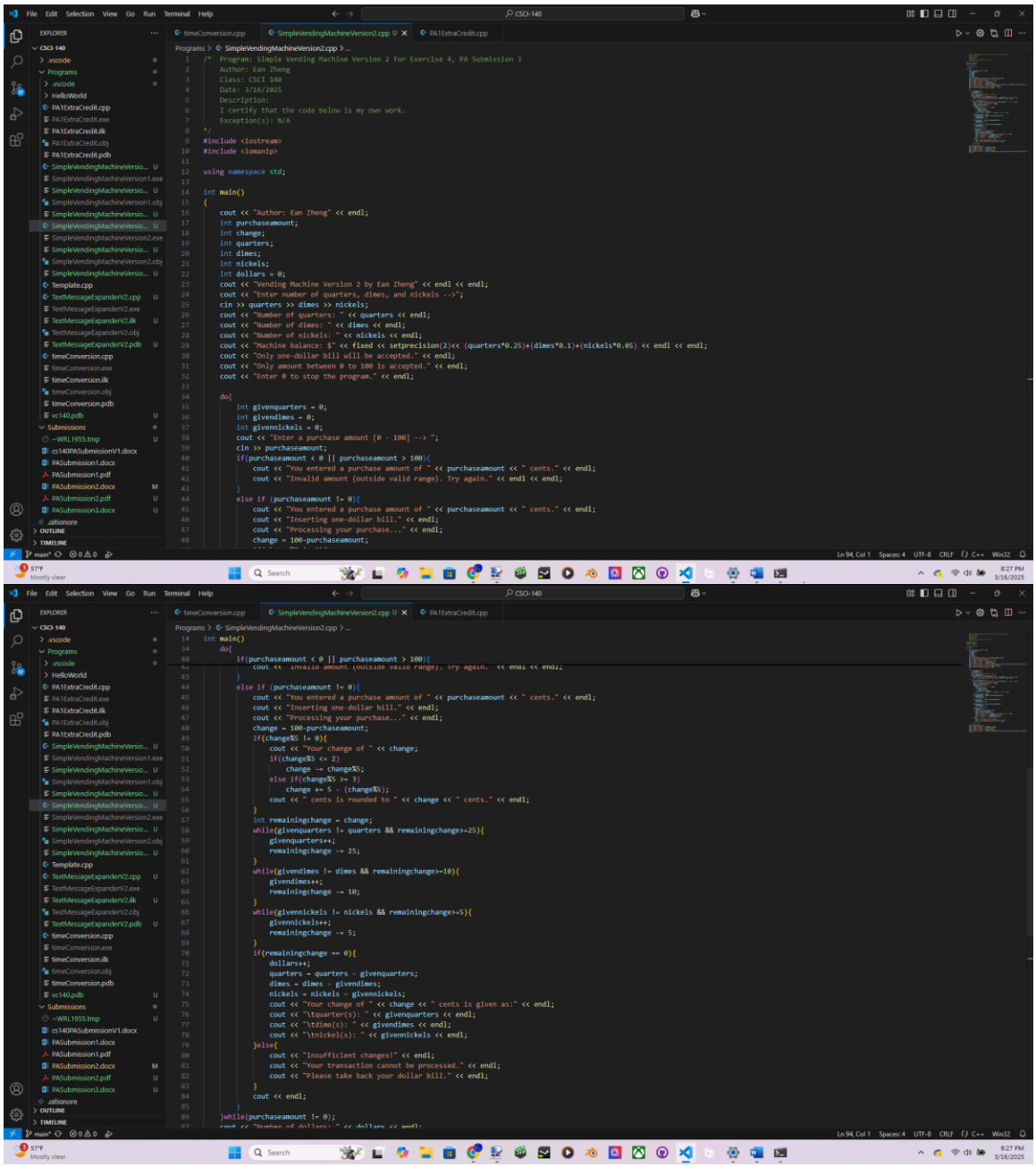
After loop is exited, output money left in machine, dollars, quarters, dimes, and nickels. Output machine balance to with same equation as before except with dollars added.

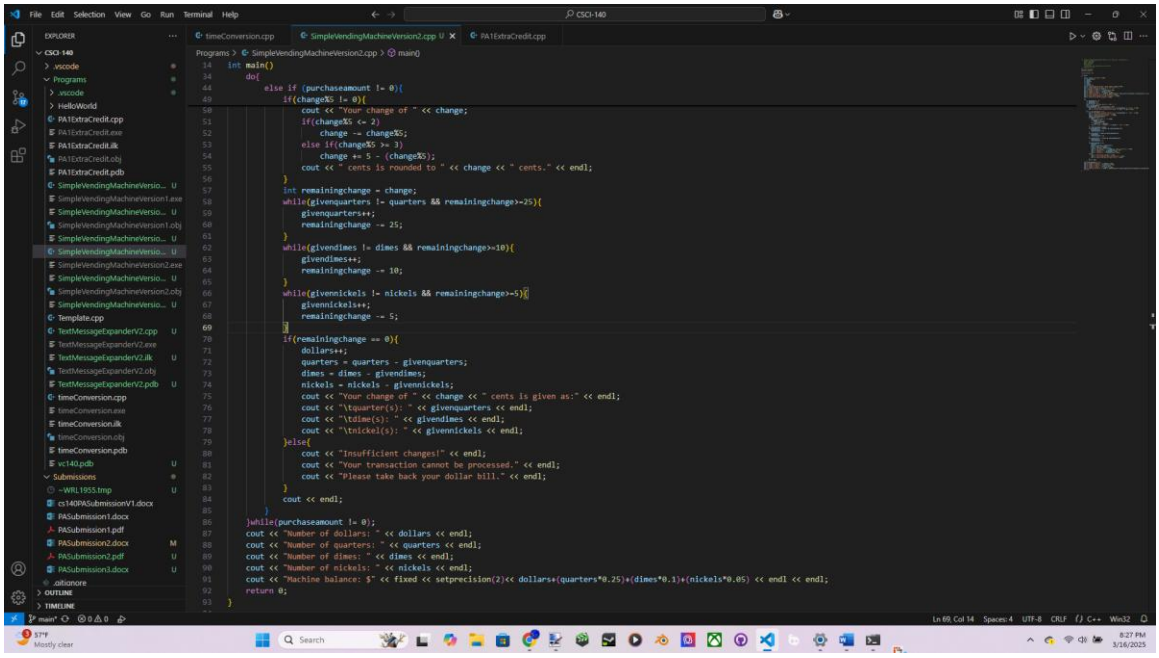
Source code below:

```
/*  Program: Simple Vending Machine Version 2 for Exercise 4, PA Submission 3
    Author: Ean Zheng
    Class: CSCI 140
    Date: 3/16/2025
    Description:
    I certify that the code below is my own work.
    Exception(s): N/A
*/
#include <iostream>
#include <iomanip>

using namespace std;

int main()
{
    cout << "Author: Ean Zheng" << endl;
    int purchaseamount;
    int change;
    int quarters;
    int dimes;
    int nickels;
    int dollars = 0;
    cout << "Vending Machine Version 2 by Ean Zheng" << endl << endl;
    cout << "Enter number of quarters, dimes, and nickels -->";
    cin >> quarters >> dimes >> nickels;
    cout << "Number of quarters: " << quarters << endl;
    cout << "Number of dimes: " << dimes << endl;
    cout << "Number of nickels: " << nickels << endl;
    cout << "Machine balance: $" << fixed << setprecision(2)<<
(quarters*0.25)+(dimes*0.1)+(nickels*0.05) << endl << endl;
    cout << "Only one-dollar bill will be accepted." << endl;
    cout << "Only amount between 0 to 100 is accepted." << endl;
    cout << "Enter 0 to stop the program." << endl;

    do{
        int givenquarters = 0;
        int givendimes = 0;
        int givennickels = 0;
        cout << "Enter a purchase amount [0 - 100] --> ";
        cin >> purchaseamount;
        if(purchaseamount < 0 || purchaseamount > 100){
            cout << "You entered a purchase amount of " << purchaseamount << " cents." <<
endl;
            cout << "Invalid amount (outside valid range). Try again." << endl << endl;
```

```
        }
        else if (purchaseamount != 0){
            cout << "You entered a purchase amount of " << purchaseamount << " cents." <<
endl;
            cout << "Inserting one-dollar bill." << endl;
            cout << "Processing your purchase..." << endl;
            change = 100-purchaseamount;
            if(change%5 != 0){
                cout << "Your change of " << change;
                if(change%5 <= 2)
                    change -= change%5;
                else if(change%5 >= 3)
                    change += 5 - (change%5);
                cout << " cents is rounded to " << change << " cents." << endl;
            }
            int remainingchange = change;
            while(givenquarters != quarters && remainingchange>=25){
                givenquarters++;
                remainingchange -= 25;
            }
            while(givendimes != dimes && remainingchange>=10){
                givendimes++;
                remainingchange -= 10;
            }
            while(givennickels != nickels && remainingchange>=5){
                givennickels++;
                remainingchange -= 5;
            }
            if(remainingchange == 0){
                dollars++;
                quarters = quarters - givenquarters;
                dimes = dimes - givendimes;
                nickels = nickels - givennickels;
                cout << "Your change of " << change << " cents is given as:" << endl;
                cout << "\tquarter(s): " << givenquarters << endl;
                cout << "\tdime(s): " << givendimes << endl;
                cout << "\tnickel(s): " << givennickels << endl;
            }else{
                cout << "Insufficient changes!" << endl;
                cout << "Your transaction cannot be processed." << endl;
                cout << "Please take back your dollar bill." << endl;
            }
            cout << endl;
        }
    }while(purchaseamount != 0);
    cout << "Number of dollars: " << dollars << endl;
```

```
    cout << "Number of quarters: " << quarters << endl;
    cout << "Number of dimes: " << dimes << endl;
    cout << "Number of nickels: " << nickels << endl;
    cout << "Machine balance: $" << fixed << setprecision(2)<<
dollars+(quarters*0.25)+(dimes*0.1)+(nickels*0.05) << endl << endl;
    return 0;
}
```




File Edit Selection View Go Run Terminal Help
CSCI-140
EXPLORER
timeConversion.cpp
SimpleVendingMachineVersion2.cpp U
PA1ExtraCredit.cpp
Programs > SimpleVendingMachineVersion2.cpp > main()
int main()
do{
else if (purchaseamount != 0){
if(change%5 != 0){
cout << "Your change of " << change;
if(change%5 <= 2)
change -= change%5;
else if(change%5 >= 3)
change += 5 - (change%5);
cout << " cents is rounded to " << change << " cents." << endl;
}
int remainingchange = change;
while(givenquarters != quarters && remainingchange>=25){
givenquarters++;
remainingchange -= 25;
}
while(givendimes != dimes && remainingchange>=10){
givendimes++;
remainingchange -= 10;
}
while(givennickels != nickels && remainingchange>=5){
givennickels++;
remainingchange -= 5;
}
if(remainingchange == 0){
dollars++;
quarters = quarters - givenquarters;
dimes = dimes - givendimes;
nickels = nickels - givennickels;
cout << "Your change of " << change << " cents is given as:" << endl;
cout << "\tquarter(s): " << givenquarters << endl;
cout << "\tdime(s): " << givendimes << endl;
cout << "\tnickel(s): " << givennickels << endl;
}else{
cout << "Insufficient changes!" << endl;
cout << "Your transaction cannot be processed." << endl;
cout << "Please take back your dollar bill." << endl;
}
cout << endl;
}
}while(purchaseamount != 0);
cout << "Number of dollars: " << dollars << endl;
cout << "Number of quarters: " << quarters << endl;
cout << "Number of dimes: " << dimes << endl;
cout << "Number of nickels: " << nickels << endl;
cout << "Machine balance: $" << fixed << setprecision(2)<< dollars+(quarters*0.25)+(dimes*0.1)+(nickels*0.05) << endl << endl;
return 0;
}
Ln 69, Col 14 Spaces: 4 UTF-8 CRLF C++ Win32
57°F
Mostly clear
Search
8:27 PM
3/16/2025

Input/output below:

Author: Ean Zheng
Vending Machine Version 2 by Ean Zheng

Enter number of quarters, dimes, and nickels -->2 4 4
Number of quarters: 2
Number of dimes: 4
Number of nickels: 4
Machine balance: $1.10

Only one-dollar bill will be accepted.
Only amount between 0 to 100 is accepted.
Enter 0 to stop the program.
Enter a purchase amount [0 - 100] --> 36
You entered a purchase amount of 36 cents.
Inserting one-dollar bill.
Processing your purchase...
Your change of 64 cents is rounded to 65 cents.
Your change of 65 cents is given as:
    quarter(s): 2
    dime(s): 1
    nickel(s): 1

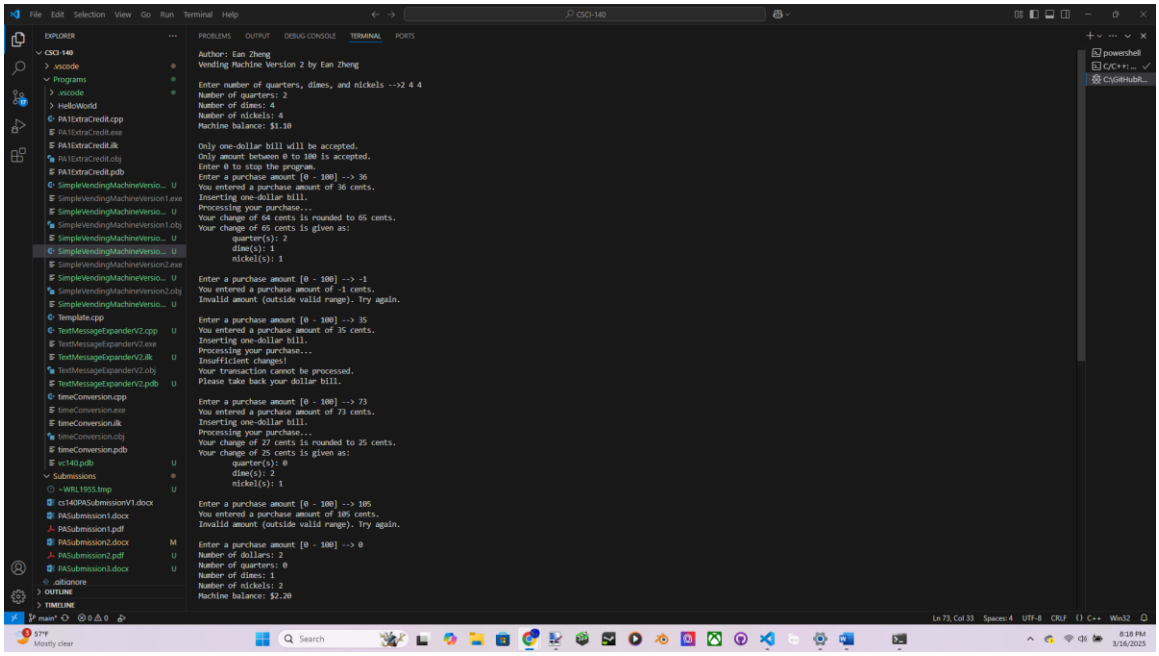
Enter a purchase amount [0 - 100] --> -1
You entered a purchase amount of -1 cents.
Invalid amount (outside valid range). Try again.

Enter a purchase amount [0 - 100] --> 35
You entered a purchase amount of 35 cents.
Inserting one-dollar bill.
Processing your purchase...
Insufficient changes!
Your transaction cannot be processed.
Please take back your dollar bill.

Enter a purchase amount [0 - 100] --> 73
You entered a purchase amount of 73 cents.
Inserting one-dollar bill.
Processing your purchase...
Your change of 27 cents is rounded to 25 cents.
Your change of 25 cents is given as:
    quarter(s): 0
    dime(s): 2
    nickel(s): 1

Enter a purchase amount [0 - 100] --> 105
Enter a purchase amount [0 - 100] --> 105
You entered a purchase amount of 105 cents.
Invalid amount (outside valid range). Try again.

Enter a purchase amount [0 - 100] --> 0
Number of dollars: 2
Number of quarters: 0
Number of dimes: 1
Number of nickels: 2
Machine balance: $2.20



Question 1: A while loop and a for loop are equivalent, but we may prefer to use one type of loop to another type of loop depending on the situation. Discuss when it is best to use a for loop over a while loop.

When we have a set number of repetitions for that loop. For example, if we know we have to repeat this 4 times, a for loop can easily have that condition in a much simpler, compact, and faster way.

Question 2: You are testing a program with a while loop. Which of the following test cases can be safe to ignore (select one best option that you might not have to try)? Pick one and briefly explain why it might not be needed.
a. Skip the loop entirely
b. Run the loop once
c. Run the loop a few times
d. Run the loop forever

C: run the loop a few times. The only things necessary to know about the loop is what makes it iterate and what prevents it. One time is good enough to know what makes it iterate. Therefore, a few times isn't necessary when it can be solved with a 1 iteration case.

Extra Credit (2 points): The strategy with maximizing the coin with highest value first will work for most cases, but there are a few cases where it would not work (such as Q = 2, D = 4, N = 0, and you need to give back 55 cents). You do not have to handle such situation with the regular version, and it is okay reject such transaction as insufficient coins (try it out with current version to confirm it). Modify your program to handle the above case and it must be a general solution so it would work with similar situation (think of another situation). Try your new solution with the original test cases above (should work like before), the special test case below, and another special situation that would not work with the original version.

Vending Machine Version 2 EC by [Your Name]
Enter number of quarters, dimes, and nickels --> 2 4 0<Enter>
Number of quarters: 2
Number of dimes : 4
Number of nickels : 0
Machine balance : $0.90
Only one-dollar bill will be accepted.
Only amount between 0 to 100 is accepted.
Enter 0 to stop the program.
Enter a purchase amount [0 - 100] --> 45<Enter>
You entered a purchase amount of 45 cents.
Inserting one-dollar bill.
Processing your purchase ...
Your change of 55 cents is given as:
quarter(s): 1
dime(s): 3
nickel(s): 0
Enter a purchase amount [0 - 100] --> 0<Enter>
Number of dollars: 1
Number of quarters: 1
Number of dimes : 1
Number of nickels : 0
Machine balance : $1.35

Fill out and turn in the PA submission file with this lab (save as PDF format).

Source Code:

```
/*  Program: Simple Vending Machine Version 2 for Extra Credit, PA Submission 3
    Author: Ean Zheng
    Class: CSCI 140
    Date: 3/16/2025
    Description:
    I certify that the code below is my own work.
    Exception(s): N/A
*/
#include <iostream>
#include <iomanip>

using namespace std;

int main()
{
    cout << "Author: Ean Zheng" << endl;
    int purchaseamount;
    int change;
    int quarters;
    int dimes;
    int nickels;
    int dollars = 0;
    cout << "Vending Machine Version 2 by Ean Zheng" << endl << endl;
    cout << "Enter number of quarters, dimes, and nickels -->";
    cin >> quarters >> dimes >> nickels;
    cout << "Number of quarters: " << quarters << endl;
    cout << "Number of dimes: " << dimes << endl;
    cout << "Number of nickels: " << nickels << endl;
    cout << "Machine balance: $" << fixed << setprecision(2)<<
(quarters*0.25)+(dimes*0.1)+(nickels*0.05) << endl << endl;
    cout << "Only one-dollar bill will be accepted." << endl;
    cout << "Only amount between 0 to 100 is accepted." << endl;
    cout << "Enter 0 to stop the program." << endl;

    do{
        int givenquarters = 0;
        int givendimes = 0;
        int givennickels = 0;
        cout << "Enter a purchase amount [0 - 100] --> ";
        cin >> purchaseamount;
        if(purchaseamount < 0 || purchaseamount > 100){
            cout << "You entered a purchase amount of " << purchaseamount << " cents." <<
endl;
            cout << "Invalid amount (outside valid range). Try again." << endl << endl;
```

```
    }
    else if (purchaseamount != 0){
        cout << "You entered a purchase amount of " << purchaseamount << " cents." <<
endl;
        cout << "Inserting one-dollar bill." << endl;
        cout << "Processing your purchase..." << endl;
        change = 100-purchaseamount;
        if(change%5 != 0){
            cout << "Your change of " << change;
            if(change%5 <= 2)
                change -= change%5;
            else if(change%5 >= 3)
                change += 5 - (change%5);
            cout << " cents is rounded to " << change << " cents." << endl;
        }
        for(givenquarters = 0; (givenquarters <= quarters && givenquarters <=
4)&&((givenquarters*25)+(givendimes*10)+(givennickels*5)!=change);
givenquarters++){
            for(givendimes = 0; (givendimes <= dimes && givendimes <=
10)&&((givenquarters*25)+(givendimes*10)+(givennickels*5)!=change);
givendimes++){
                for(givennickels = 0; (givennickels <= nickels && givennickels <=
20)&&((givenquarters*25)+(givendimes*10)+(givennickels*5)!=change);
givennickels++){
                    if((givenquarters*25)+(givendimes*10)+(givennickels*5)==change)break;
                }
                if(givennickels > nickels || givennickels > 20)givennickels = 0;
                if((givenquarters*25)+(givendimes*10)+(givennickels*5)==change)break;
            }
            if(givendimes > dimes || givendimes > 10)givendimes = 0;
            if((givenquarters*25)+(givendimes*10)+(givennickels*5)==change)break;
        }
        if((givenquarters*25)+(givendimes*10)+(givennickels*5)==change){
            dollars++;
            quarters = quarters - givenquarters;
            dimes = dimes - givendimes;
            nickels = nickels - givennickels;
            cout << "Your change of " << change << " cents is given as:" << endl;
            cout << "\tquarter(s): " << givenquarters << endl;
            cout << "\tdime(s): " << givendimes << endl;
            cout << "\tnickel(s): " << givennickels << endl;
        }else{
            cout << "Insufficient changes!" << endl;
            cout << "Your transaction cannot be processed." << endl;
            cout << "Please take back your dollar bill." << endl;
        }
```

```
            cout << endl;
        }
    }while(purchaseamount != 0);
    cout << "Number of dollars: " << dollars << endl;
    cout << "Number of quarters: " << quarters << endl;
    cout << "Number of dimes: " << dimes << endl;
    cout << "Number of nickels: " << nickels << endl;
    cout << "Machine balance: $" << fixed << setprecision(2)<<
dollars+(quarters*0.25)+(dimes*0.1)+(nickels*0.05) << endl << endl;
    return 0;
}
```

Output/Input:

```
Author: Ean Zheng
Vending Machine Version 2 by Ean Zheng

Enter number of quarters, dimes, and nickels -->2 4 0
Number of quarters: 2
Number of dimes: 4
Number of nickels: 0
Machine balance: $0.90

Only one-dollar bill will be accepted.
Only amount between 0 to 100 is accepted.
Enter 0 to stop the program.
Enter a purchase amount [0 - 100] --> 45
You entered a purchase amount of 45 cents.
Inserting one-dollar bill.
Processing your purchase...
Your change of 55 cents is given as:
        quarter(s): 1
        dime(s): 3
        nickel(s): 0

Enter a purchase amount [0 - 100] --> 0
Number of dollars: 1
Number of quarters: 1
Number of dimes: 1
Number of nickels: 0
Machine balance: $1.35
```