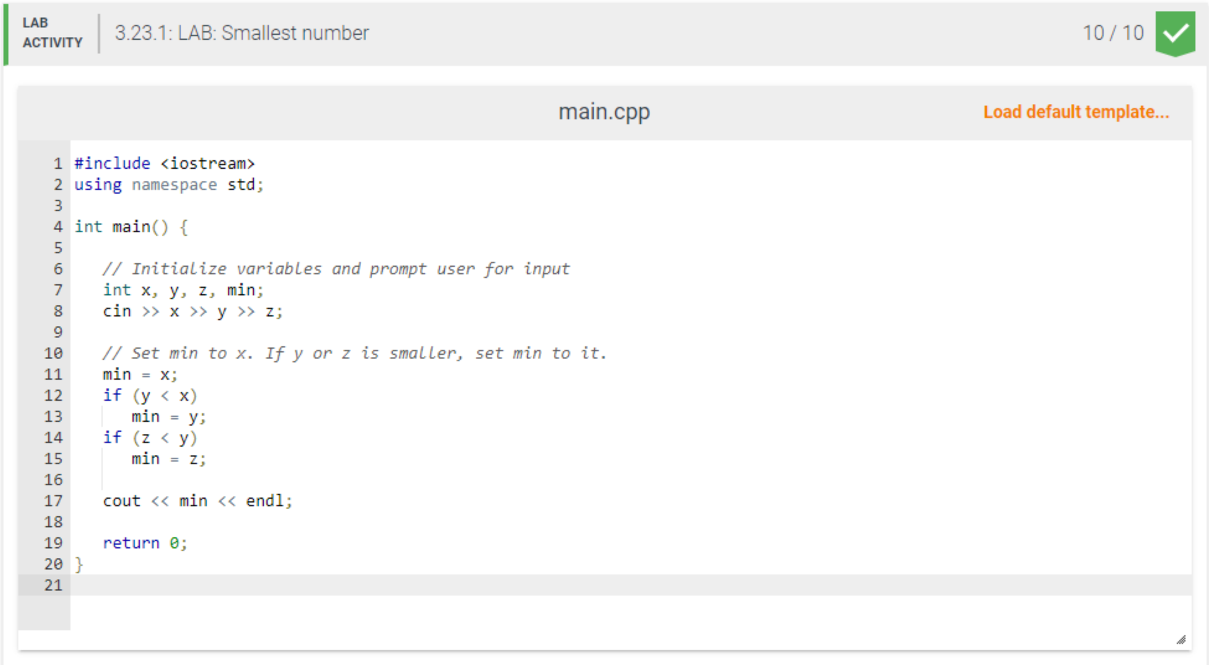
CSCI 140 PA #2 Submission

Due Date: 3/7/2023

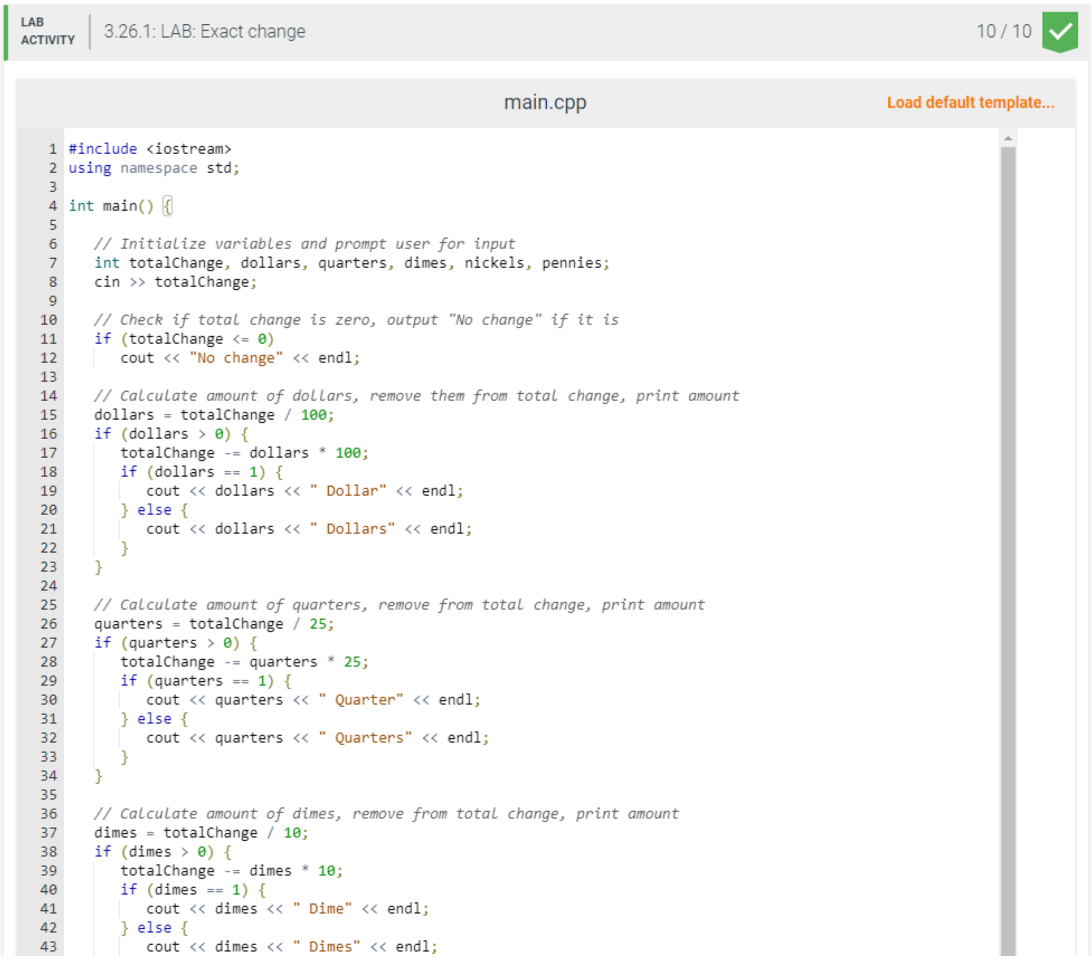
Name(s): Ali Mortada

Exercise 1 –**zyBook 3.23 LAB: Smallest number**
-- check if completely done in zyBook __X__; otherwise, discuss issues below
Include a screenshot of current status/score



```
#include <iostream>
using namespace std;

int main() {

   // Initialize variables and prompt user for input
   int x, y, z, min;
   cin >> x >> y >> z;

   // Set min to x. If y or z is smaller, set min to it.
   min = x;
   if (y < x)
      min = y;
   if (z < y)
      min = z;

   cout << min << endl;

   return 0;
}

```

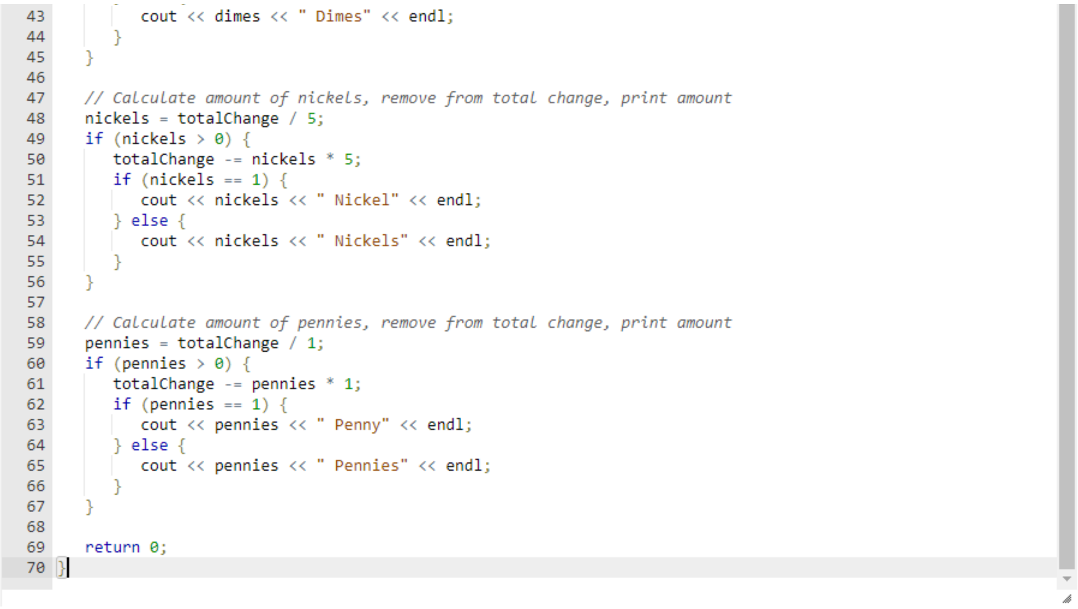
Exercise 2 – **zyBook 3.26 LAB: Exact change**
-- check if completely done in zyBook __X__; otherwise, discuss issues below
Include a screenshot of current status/score



LAB ACTIVITY | 3.26.1: LAB: Exact change | 10 / 10

main.cpp

```cpp
#include <iostream>
using namespace std;

int main() {

   // Initialize variables and prompt user for input
   int totalChange, dollars, quarters, dimes, nickels, pennies;
   cin >> totalChange;

   // Check if total change is zero, output "No change" if it is
   if (totalChange <= 0)
      cout << "No change" << endl;

   // Calculate amount of dollars, remove them from total change, print amount
   dollars = totalChange / 100;
   if (dollars > 0) {
      totalChange -= dollars * 100;
      if (dollars == 1) {
         cout << dollars << " Dollar" << endl;
      } else {
         cout << dollars << " Dollars" << endl;
      }
   }

   // Calculate amount of quarters, remove from total change, print amount
   quarters = totalChange / 25;
   if (quarters > 0) {
      totalChange -= quarters * 25;
      if (quarters == 1) {
         cout << quarters << " Quarter" << endl;
      } else {
         cout << quarters << " Quarters" << endl;
      }
   }

   // Calculate amount of dimes, remove from total change, print amount
   dimes = totalChange / 10;
   if (dimes > 0) {
      totalChange -= dimes * 10;
      if (dimes == 1) {
         cout << dimes << " Dime" << endl;
      } else {
         cout << dimes << " Dimes" << endl;
```



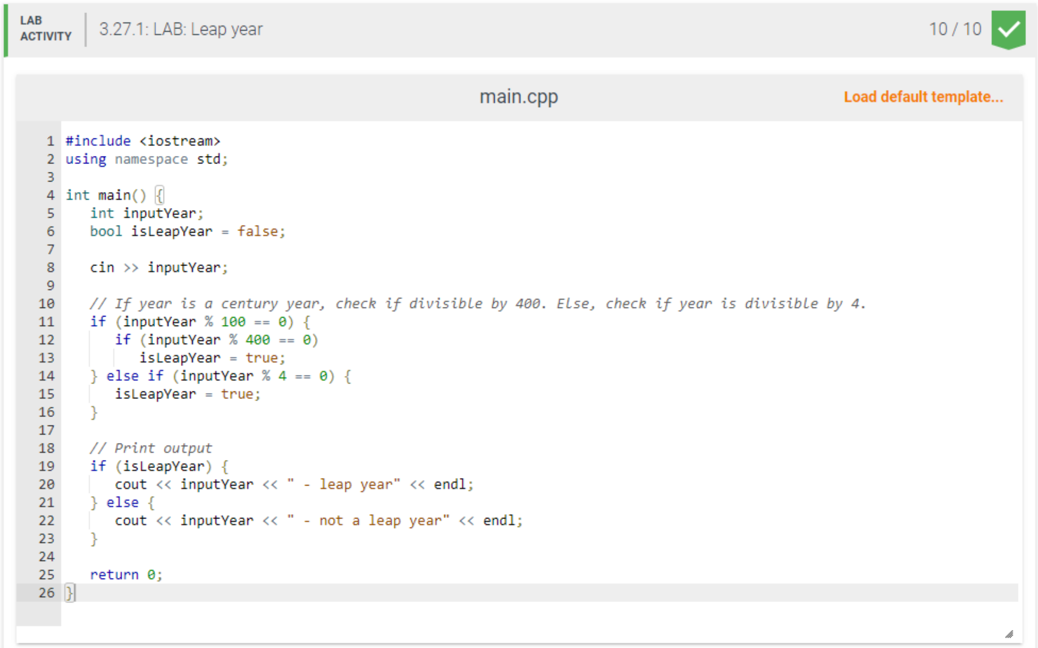
cout << dimes << " Dimes" << endl;

```cpp
      }
   }

   // Calculate amount of nickels, remove from total change, print amount
   nickels = totalChange / 5;
   if (nickels > 0) {
      totalChange -= nickels * 5;
      if (nickels == 1) {
         cout << nickels << " Nickel" << endl;
      } else {
         cout << nickels << " Nickels" << endl;
      }
   }

   // Calculate amount of pennies, remove from total change, print amount
   pennies = totalChange / 1;
   if (pennies > 0) {
      totalChange -= pennies * 1;
      if (pennies == 1) {
         cout << pennies << " Penny" << endl;
      } else {
         cout << pennies << " Pennies" << endl;
      }
   }

   return 0;
}
```

Exercise 3 – **zyBook 3.27 LAB: Leap Year**

-- check if completely done in zyBook __X__; otherwise, discuss issues below

Include a screenshot of current status/score

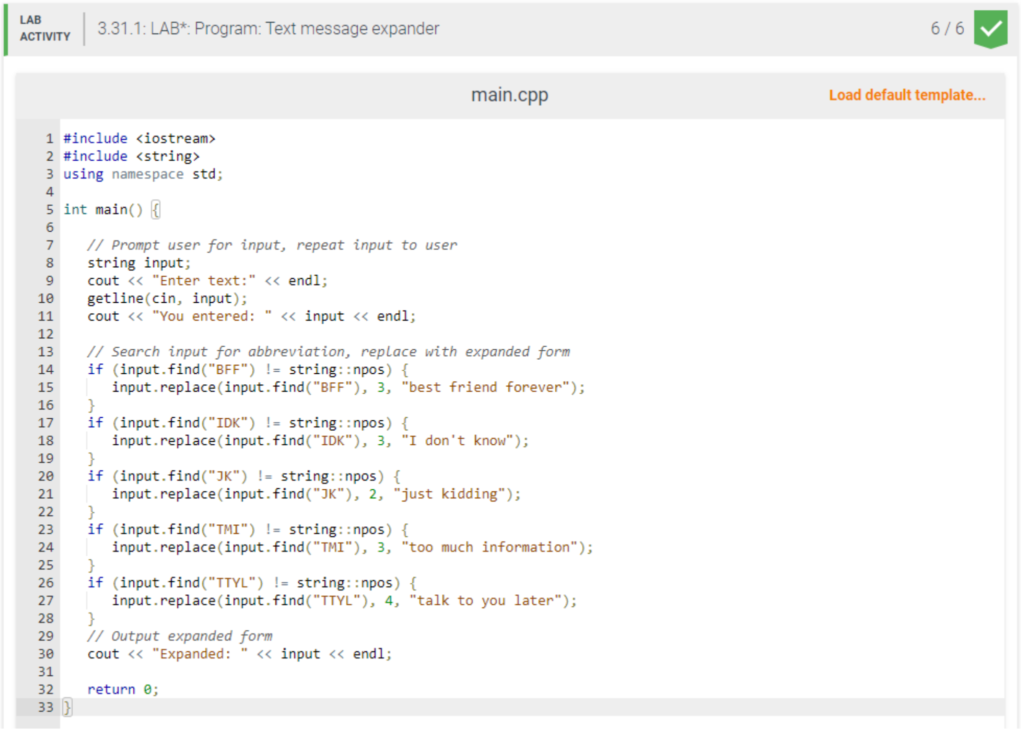


```cpp
#include <iostream>
using namespace std;

int main() {
   int inputYear;
   bool isLeapYear = false;

   cin >> inputYear;

   // If year is a century year, check if divisible by 400. Else, check if year is divisible by 4.
   if (inputYear % 100 == 0) {
      if (inputYear % 400 == 0)
         isLeapYear = true;
   } else if (inputYear % 4 == 0) {
      isLeapYear = true;
   }

   // Print output
   if (isLeapYear) {
      cout << inputYear << " - leap year" << endl;
   } else {
      cout << inputYear << " - not a leap year" << endl;
   }

   return 0;
}
```

Exercise 4 – **zyBook 3.31 LAB*: Program: Text message expander**

-- check if completely done in zyBook __X__; otherwise, discuss issues below

Include a screenshot of current status/score



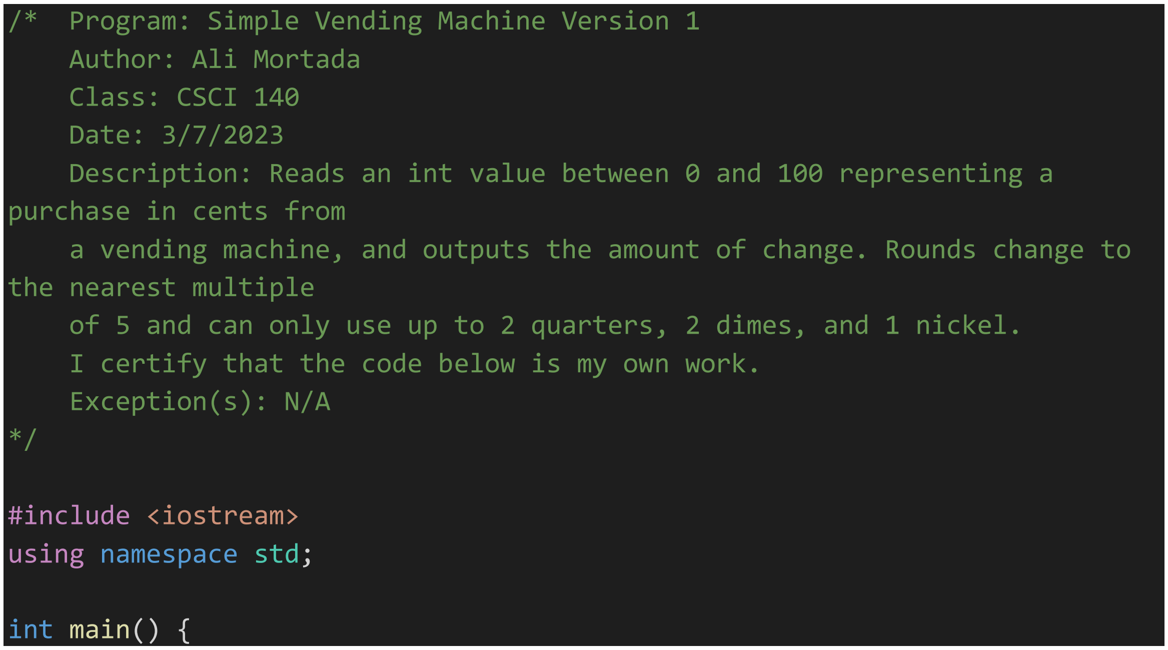
```cpp
#include <iostream>
#include <string>
using namespace std;

int main() {

   // Prompt user for input, repeat input to user
   string input;
   cout << "Enter text:" << endl;
   getline(cin, input);
   cout << "You entered: " << input << endl;

   // Search input for abbreviation, replace with expanded form
   if (input.find("BFF") != string::npos) {
      input.replace(input.find("BFF"), 3, "best friend forever");
   }
   if (input.find("IDK") != string::npos) {
      input.replace(input.find("IDK"), 3, "I don't know");
   }
   if (input.find("JK") != string::npos) {
      input.replace(input.find("JK"), 2, "just kidding");
   }
   if (input.find("TMI") != string::npos) {
      input.replace(input.find("TMI"), 3, "too much information");
   }
   if (input.find("TTYL") != string::npos) {
      input.replace(input.find("TTYL"), 4, "talk to you later");
   }
   // Output expanded form
   cout << "Expanded: " << input << endl;

   return 0;
}
```

Exercise 5 -- **Simple Vending Machine Version 1**
-- check if completely done __X__; otherwise, discuss issues below.

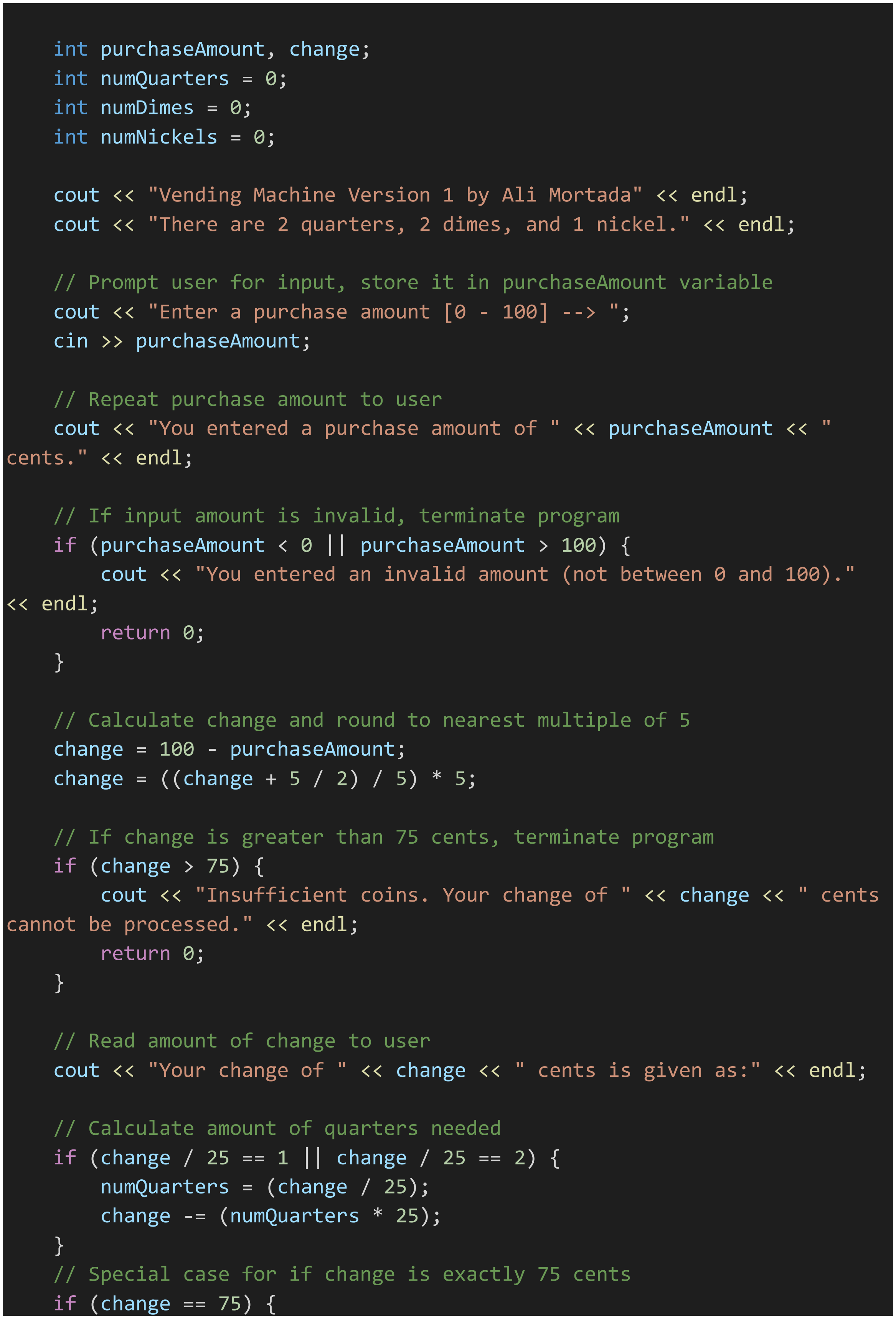
Pseudocode below if applicable:

```
-read input from user of integer value between 0 and 100

        -if input value is not between range, output error message

-Determine amount of change returned from one dollar

-round change to nearest multiple of 5

-reject if more than 75 cents are needed

        -use up to 2 quarters, then find the remainder and remove number of quarters
        from change

        -use up to 2 dimes, then find the remainder and remove number of dimes from
        change

        -use up to 1 nickel

-print number of quarters, dimes, and nickels used
```

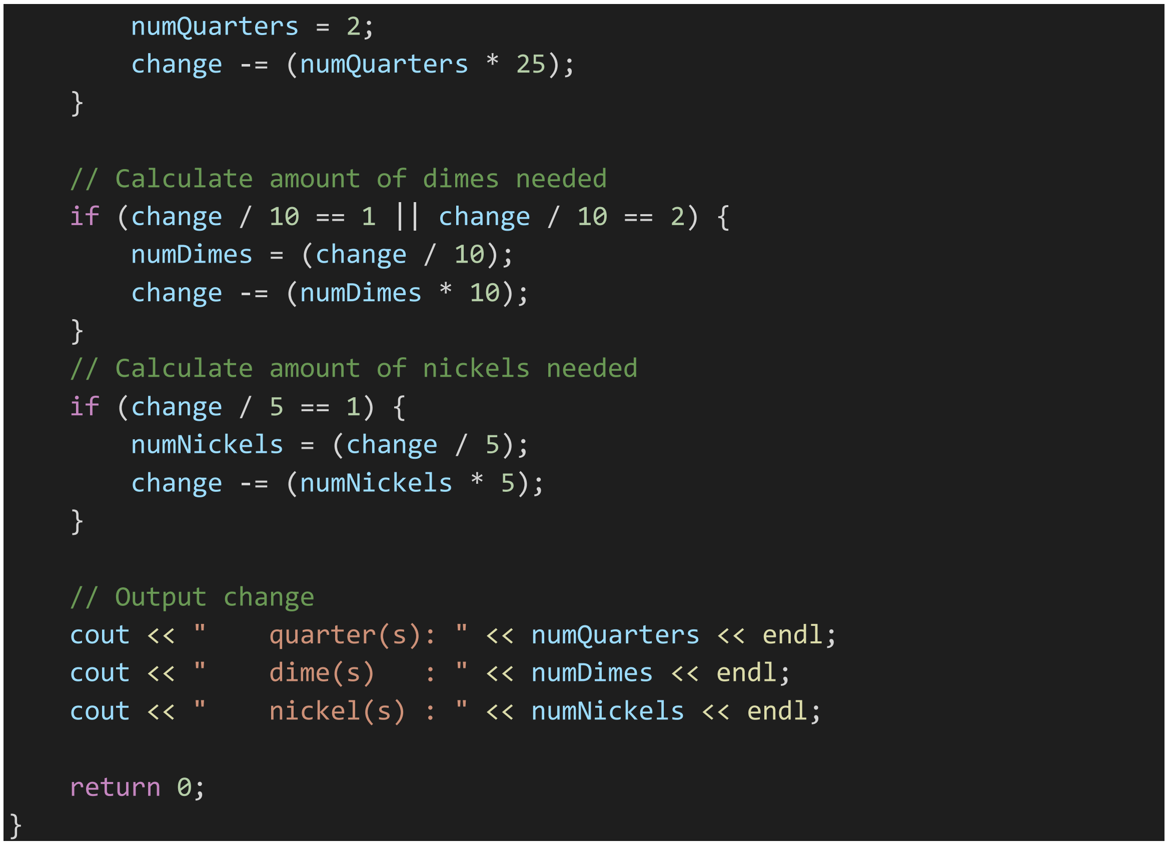
Source code below:



```
/*  Program: Simple Vending Machine Version 1
    Author: Ali Mortada
    Class: CSCI 140
    Date: 3/7/2023
    Description: Reads an int value between 0 and 100 representing a
purchase in cents from
    a vending machine, and outputs the amount of change. Rounds change to
the nearest multiple
    of 5 and can only use up to 2 quarters, 2 dimes, and 1 nickel.
    I certify that the code below is my own work.
    Exception(s): N/A
*/

#include <iostream>
using namespace std;

int main() {
```

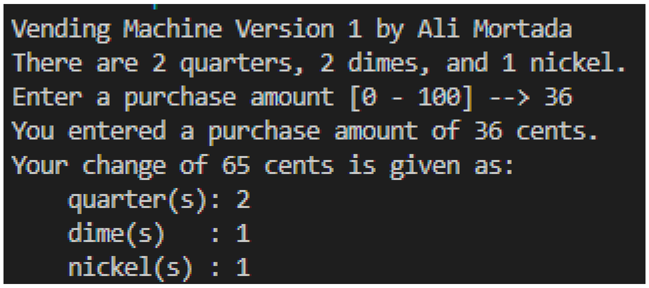


```
    int purchaseAmount, change;
    int numQuarters = 0;
    int numDimes = 0;
    int numNickels = 0;

    cout << "Vending Machine Version 1 by Ali Mortada" << endl;
    cout << "There are 2 quarters, 2 dimes, and 1 nickel." << endl;

    // Prompt user for input, store it in purchaseAmount variable
    cout << "Enter a purchase amount [0 - 100] --> ";
    cin >> purchaseAmount;

    // Repeat purchase amount to user
    cout << "You entered a purchase amount of " << purchaseAmount << "
cents." << endl;

    // If input amount is invalid, terminate program
    if (purchaseAmount < 0 || purchaseAmount > 100) {
        cout << "You entered an invalid amount (not between 0 and 100)."
<< endl;
        return 0;
    }

    // Calculate change and round to nearest multiple of 5
    change = 100 - purchaseAmount;
    change = ((change + 5 / 2) / 5) * 5;

    // If change is greater than 75 cents, terminate program
    if (change > 75) {
        cout << "Insufficient coins. Your change of " << change << " cents
cannot be processed." << endl;
        return 0;
    }

    // Read amount of change to user
    cout << "Your change of " << change << " cents is given as:" << endl;

    // Calculate amount of quarters needed
    if (change / 25 == 1 || change / 25 == 2) {
        numQuarters = (change / 25);
        change -= (numQuarters * 25);
    }
    // Special case for if change is exactly 75 cents
    if (change == 75) {
```

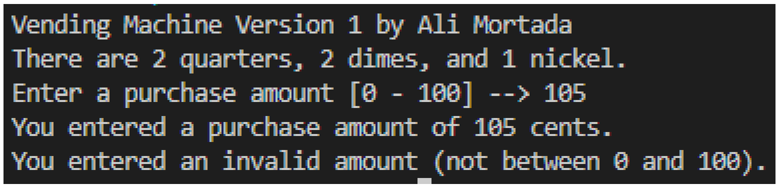


```
        numQuarters = 2;
        change -= (numQuarters * 25);
    }

    // Calculate amount of dimes needed
    if (change / 10 == 1 || change / 10 == 2) {
        numDimes = (change / 10);
        change -= (numDimes * 10);
    }
    // Calculate amount of nickels needed
    if (change / 5 == 1) {
        numNickels = (change / 5);
        change -= (numNickels * 5);
    }

    // Output change
    cout << "    quarter(s): " << numQuarters << endl;
    cout << "    dime(s)   : " << numDimes << endl;
    cout << "    nickel(s) : " << numNickels << endl;

    return 0;
}
```

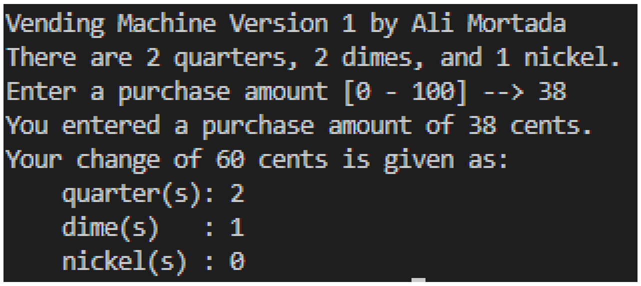
Input/output below:



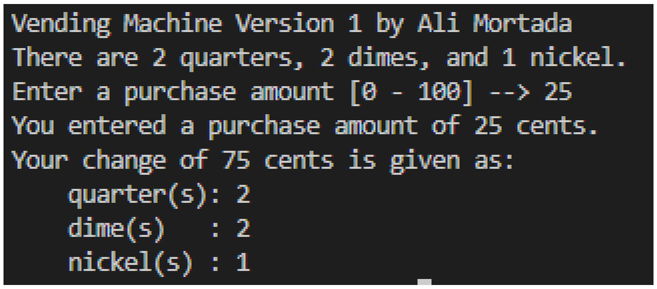
```
Vending Machine Version 1 by Ali Mortada
There are 2 quarters, 2 dimes, and 1 nickel.
Enter a purchase amount [0 - 100] --> 36
You entered a purchase amount of 36 cents.
Your change of 65 cents is given as:
    quarter(s): 2
    dime(s)   : 1
    nickel(s) : 1
```



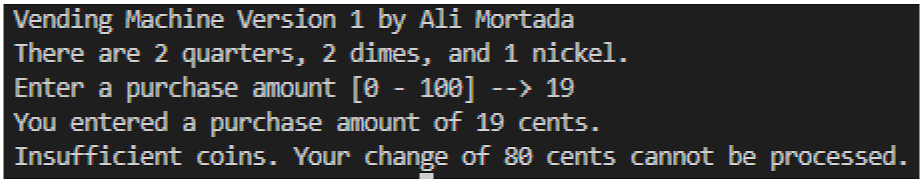
```
Vending Machine Version 1 by Ali Mortada
There are 2 quarters, 2 dimes, and 1 nickel.
Enter a purchase amount [0 - 100] --> 105
You entered a purchase amount of 105 cents.
You entered an invalid amount (not between 0 and 100).
```



```
Vending Machine Version 1 by Ali Mortada
There are 2 quarters, 2 dimes, and 1 nickel.
Enter a purchase amount [0 - 100] --> 38
You entered a purchase amount of 38 cents.
Your change of 60 cents is given as:
    quarter(s): 2
    dime(s)   : 1
    nickel(s) : 0
```



```
Vending Machine Version 1 by Ali Mortada
There are 2 quarters, 2 dimes, and 1 nickel.
Enter a purchase amount [0 - 100] --> 25
You entered a purchase amount of 25 cents.
Your change of 75 cents is given as:
    quarter(s): 2
    dime(s)   : 2
    nickel(s) : 1
```



```
Vending Machine Version 1 by Ali Mortada
There are 2 quarters, 2 dimes, and 1 nickel.
Enter a purchase amount [0 - 100] --> 19
You entered a purchase amount of 19 cents.
Insufficient coins. Your change of 80 cents cannot be processed.
```

Answer for Question 1

An expression is an executable line of code, whereas a condition is a comparison or inequality which is either true or false. You can use an expression when a condition is required in C++, which is something carried over from the C language. In this case, 0 would be false and any other value would be true.

Answer for Question 2

Extra Credit – provide if applicable