**CSC 1101 – Problem Solving and Programming Laboratory**

**Lab 10 – Trevor Trusty**

**25 points – Due February 18, 11pm**

**a)** Save this document with your name and the homework number somewhere in the file name.

**b)** Type/paste your answers into the document.

**c)** Submit this document and your .cpp file(s) to the Canvas item where you downloaded this document. Do not submit a zip file but individually attach your files.

**1) [12 points]** Remember that great application you wrote for *Letter Lovers* (Lab 8-3). Now they want you to add a data validation feature that restricts the entered string, and analyzes all characters in the string. Start with the key or your Lab 8-3 code. Modify the application header and close to say “Letter Lovers, v2”. Use a validation loop to prompt for and get from the user a string containing at least ten characters. Use **getline()** to read the string and **length()** to get the length of the string. The validation loop continues to prompt the user for a string as long its length is less than ten characters. Once a string with a valid length is entered, loop through the string and count the following types of characters in the string:

● Digits

● Alphabetic

● Punctuation

● Other

Create four counting variables and an indexing variable. Use the indexing variable, a while statement, and the length() to read and analyze each character in the string. Within the loop, use an if statement to test what type of character has been read and update the appropriate counter. Use formatted output manipulators (setw, left/right) to print the rows and columns shown below. Define constants for the column widths and the minimum string length (10). Run the program five times with different strings. Get a screenshot of each run. The output should look like this:

Welcome to Letter Lovers

------------------------

Enter a string with at least 10 characters: Hello 2020!

String: Hello 2020!

Length: 11

Digits: 4

Alphas: 5

Puncts: 1

Other: 1

End of Letter Lovers

*[your program code here]\**

**//==========================================================**

**//**

**// Title: Letter Lovers v2**

**// Course: CSC 1101**

**// Lab Number: Lab10-01**

**// Author: Trevor Trusty**

**// Date: 2/18/2019**

**// Description:**

**// Analyzes the types of characters in a string entered by the user,**

**// and displays the count of numbers, letters, punctuation marks and spaces in the string.**

**//**

**//==========================================================**

**#include <conio.h> // For function getch()**

**#include <cstdlib> // For several general-purpose functions**

**#include <fstream> // For file handling**

**#include <iomanip> // For formatted output**

**#include <iostream> // For cin, cout, and system**

**#include <string> // For string data type**

**using namespace std; // So "std::cout" may be abbreviated to "cout"**

**int main()**

**{**

**string iString;**

**int digit = 0;**

**int alpha = 0;**

**int punct = 0;**

**int other = 0;**

**int index;**

**const int Col1 = 20;**

**const int Col2 = 15;**

**const string TITLE = "Letter Lovers, v2";**

**cout << "Welcome to " << TITLE << endl;**

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

**//Promt user for string**

**cout << "Type a string of at least 10 characters: ";**

**getline(cin, iString);**

**cout << endl;**

**//Test string for proper length**

**while (iString.length() < 10)**

**{**

**//Error message**

**cout << '\"' << iString << "\" " " is not valid. String must be at least 10 characters long." << endl;**

**//Reprompt for string**

**cout << "Type a string of at least 10 characters: ";**

**getline(cin, iString);**

**cout << endl;**

**}**

**//Count Digits in string**

**int length = iString.length();**

**index = 0;**

**while (index < length)**

**{**

**if (isdigit(iString.at(index)))**

**{**

**digit++;**

**}**

**index++;**

**}**

**//for (int index = 0; index < length; index++)**

**//{**

**// if (isdigit(iString.at(index)))**

**// {**

**// digit++;**

**// }**

**//}**

**//Count letters in string**

**index = 0;**

**while (index < length)**

**{**

**if (isalpha(iString.at(index)))**

**{**

**alpha++;**

**}**

**index++;**

**}**

**//for (int index = 0; index < length; index++)**

**//{**

**// if (isalpha(iString.at(index)))**

**// {**

**// alpha++;**

**// }**

**//}**

**//Count punctuations in string**

**index = 0;**

**while (index < length)**

**{**

**if (ispunct(iString.at(index)))**

**{**

**punct++;**

**}**

**index++;**

**}**

**//for (int index = 0; index < length; index++)**

**//{**

**// if (ispunct(iString.at(index)))**

**// {**

**// punct++;**

**// }**

**//}**

**//Count other/whitespace in string**

**index = 0;**

**while (index < length)**

**{**

**if (!isdigit(iString.at(index)) && !isalpha(iString.at(index)) && !ispunct(iString.at(index)))**

**{**

**other++;**

**}**

**index++;**

**}**

**//for (int index = 0; index < length; index++)**

**//{**

**// if (!isdigit(iString.at(index)) && !isalpha(iString.at(index)) && !ispunct(iString.at(index)))**

**// {**

**// other++;**

**// }**

**//}**

**cout << "====================================" << endl;**

**cout << setw(Col1) << left << "String:";**

**cout << setw(Col2) << right << iString << endl;**

**cout << setw(Col1) << left << "Length:";**

**cout << setw(Col2) << right << iString.length() << endl;**

**cout << setw(Col1) << left << "Digits:";**

**cout << setw(Col2) << right << digit << endl;**

**cout << setw(Col1) << left << "Alpha:";**

**cout << setw(Col2) << right << alpha << endl;**

**cout << setw(Col1) << left << "Punct:";**

**cout << setw(Col2) << right << punct << endl;**

**cout << setw(Col1) << left << "Other:";**

**cout << setw(Col2) << right << other << endl;**

**cout << "====================================" << endl;**

**//cout << setw(Col1) << left << "Third Character:";**

**//cout << setw(Col2) << right << iString.at(2) << endl;**

**//cout << setw(Col1) << left << "-Digit?";**

**//cout << setw(Col2) << right << isdigit(iString.at(2)) << endl;**

**//cout << setw(Col1) << left << "-Alpha?";**

**//cout << setw(Col2) << right << isalpha(iString.at(2)) << endl;**

**//cout << setw(Col1) << left << "-Punct?";**

**//cout << setw(Col2) << right << ispunct(iString.at(2)) << endl << endl;**

**//cout << setw(Col1) << left << "6th Character:";**

**//cout << setw(Col2) << right << iString.at(5) << endl;**

**//cout << setw(Col1) << left << "-Digit?";**

**//cout << setw(Col2) << right << isdigit(iString.at(5)) << endl;**

**//cout << setw(Col1) << left << "-Alpha?";**

**//cout << setw(Col2) << right << isalpha(iString.at(5)) << endl;**

**//cout << setw(Col1) << left << "-Punct?";**

**//cout << setw(Col2) << right << ispunct(iString.at(5)) << endl << endl;**

**//cout << setw(Col1) << left << "9th Character:";**

**//cout << setw(Col2) << right << iString.at(8) << endl;**

**//cout << setw(Col1) << left << "-Digit?";**

**//cout << setw(Col2) << right << isdigit(iString.at(8)) << endl;**

**//cout << setw(Col1) << left << "-Alpha?";**

**//cout << setw(Col2) << right << isalpha(iString.at(8)) << endl;**

**//cout << setw(Col1) << left << "-Punct?";**

**//cout << setw(Col2) << right << ispunct(iString.at(8)) << endl << endl;**

**//cout << "====================================" << endl;**

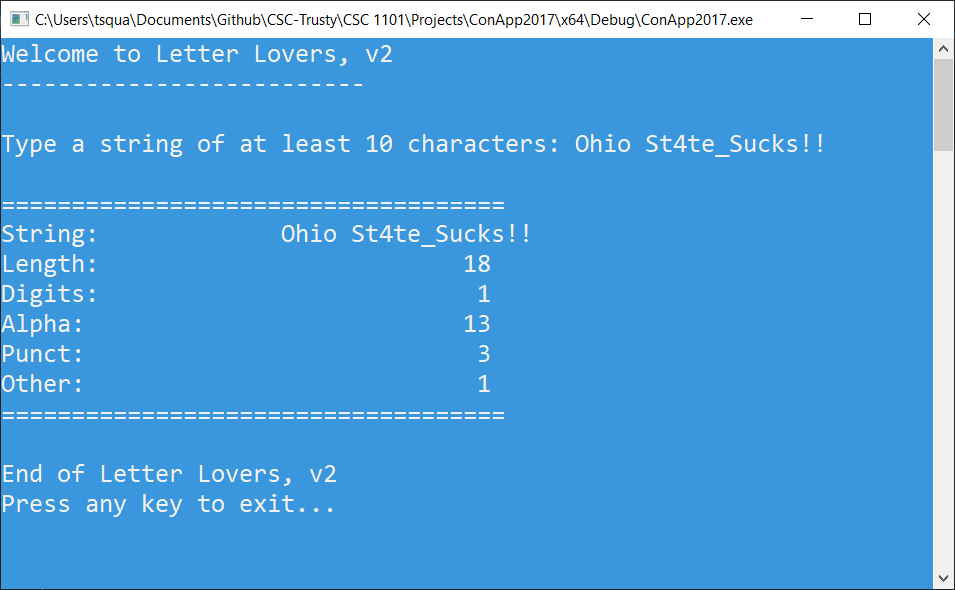
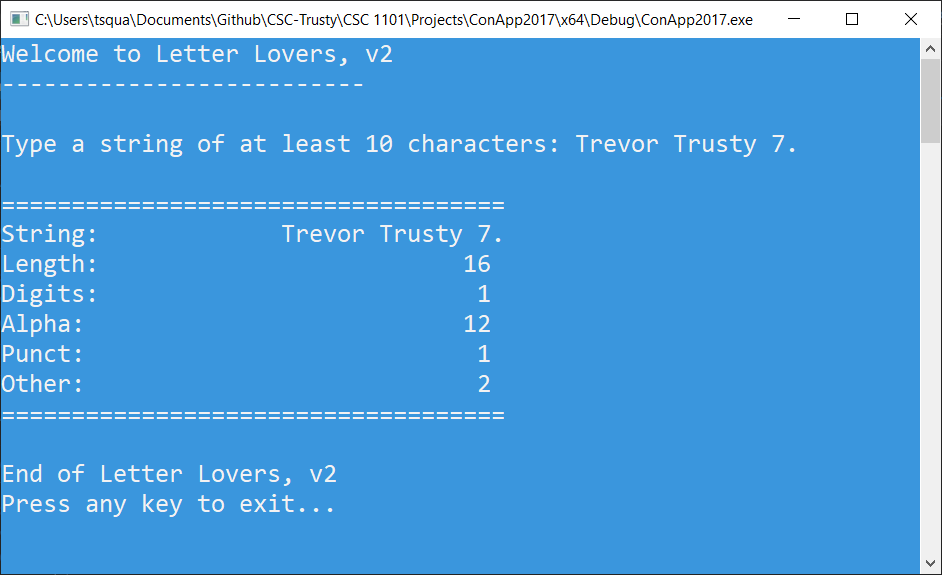
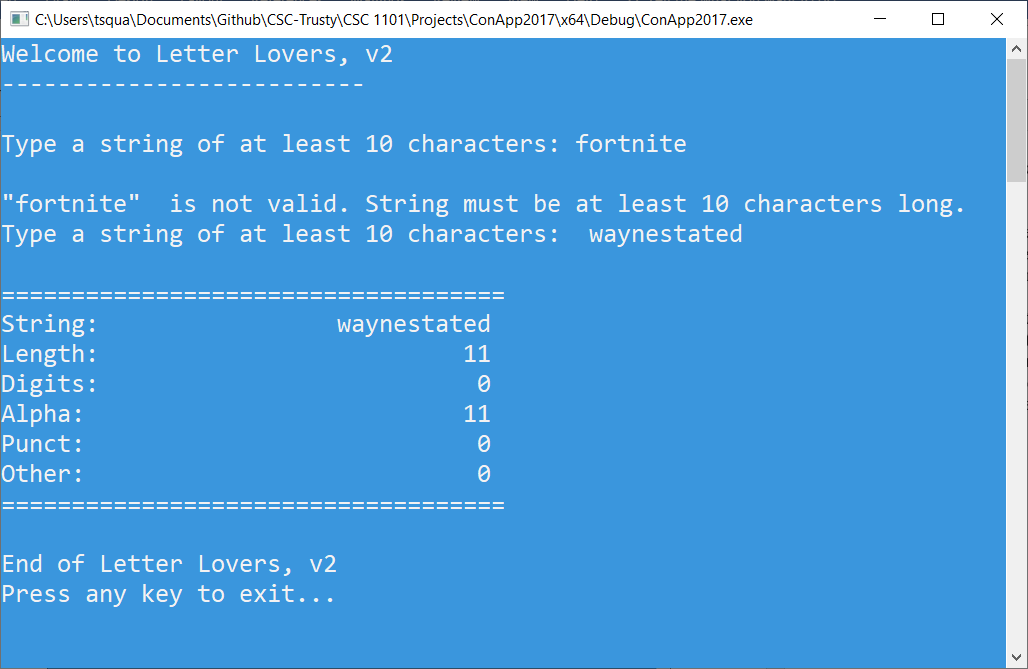
**cout << "\nEnd of " << TITLE << endl;**

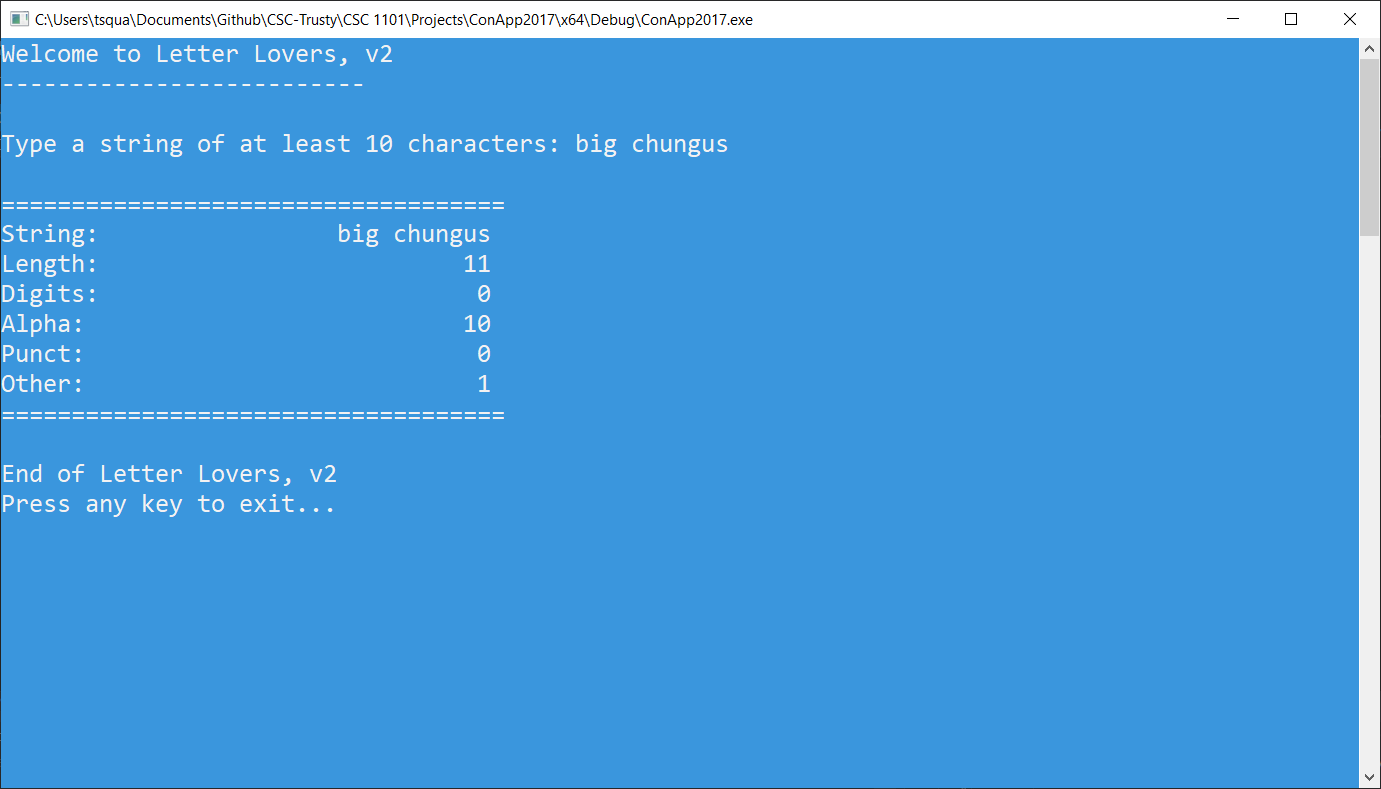
**cout << "Press any key to exit...";**

**\_getch();**

**}**

*[your program output here]\*\**

**2) [13 points]** You've been hired by *Cereal Suppers* to write a C++ console application that calculates and displays the cost of a customer’s cereal purchase. Use a validation loop to prompt for and get from the user the number of cereal boxes purchased in the range 1-9. Then use a validation loop to prompt for and get from the user the coupon discount in the range 0-20% *(but who would choose less than 20% tho?)*. Then use a validation loop to prompt for and get from the user one of the following cereal codes:

|  |  |  |
| --- | --- | --- |
| Cereal code | Cereal type | Cost per gallon |
| w | Wheaties | $4.00 |
| c | Cheerios | $3.50 |
| r | Rice Krispies | $3.00 |

*50 gallon of coco puffs please*

Use an if or switch statement to determine which cereal code the user entered. Determine the cereal type from the code, and calculate the cost of the purchase minus the discount. Use formatted output manipulators (setw, left/right) to print the following five rows:

● Boxes

● Discount

● Cereal code

● Cereal type

● Purchase amount

And two columns:

● A left-justified label (including units)

● A right-justified value.

Define constants for the column widths and cereal costs. Format all real numbers to two decimal places. Run the program with invalid and valid inputs. The output should look like this:

Welcome to Cereal Suppers

-------------------------

Enter the number of boxes purchased (1-9): 0

Error: '0' is an invalid number of boxes.

Enter the number of boxes purchased (1-9): 2

Enter the percentage discount (0-20): 30

Error: '30.00' is an invalid percentage discount.

Enter the percentage discount (0-20): 10

Enter the cereal code (w-wheaties, c-cheerios, r-rice krispies): q

Error: 'q' is an invalid cereal code.

Enter the cereal code (w-wheaties, c-cheerios, r-rice krispies): w

Boxes: 2

Discount (%): 10.00

Cereal code: w

Cereal type: Wheaties

Cereal cost ($): 4.00

Purchase cost ($): 7.20

End of Cereal Suppers

*[your program code here]\**

//==========================================================

//

// Title: Cereal Suppers

// Course: CSC 1101

// Lab Number: Lab 10-2

// Author: Trevor Trusty

// Date: <date>

// Description:

// calculates and displays the cost of a customer’s cereal purchase

//

//==========================================================

#include <conio.h> // For function getch()

#include <cstdlib> // For several general-purpose functions

#include <fstream> // For file handling

#include <iomanip> // For formatted output

#include <iostream> // For cin, cout, and system

#include <string> // For string data type

using namespace std; // So "std::cout" may be abbreviated to "cout"

int main()

{

double iNum1;

double iNum2;

double w = 4;

double c = 3.50;

double r = 3;

double discount;

char option;

double total;

const int COL1 = 20;

const int COL2 = 15;

// Show application header

cout << "Welcome to Cereal Suppers!" << endl;

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

//Prompt user for quantity of cereal

cout << "Enter quantity of cereal boxes: ";

cin >> iNum1;

cout << endl;

//Validation loop--testing cereal quantity

while (iNum1 < 1 || iNum1 > 9)

{

cout << iNum1 << " is invalid. Please enter quantity between 1 and 9.";

cout << "Enter quantity of cereal boxes: ";

cin >> iNum1;

cout << endl;

}

//Prompt user for discount percent

cout << "Enter your coupon discount % (0-20): ";

cin >> iNum2;

cout << endl;

//Validation loop--testing discount percentage

while (iNum2 < 0 || iNum2 > 20)

{

cout << iNum2 << "% is invalid. Please enter discount between 0% and 20%.";

cout << "Enter your coupon discount % (0-20): ";

cin >> iNum2;

cout << endl;

}

do

{

//Menu/sentinel loop: loop until user exits menu

cout << "\nCereal Code Menu" << endl;

cout << "w - Wheaties" << endl;

cout << "c - Cheerios" << endl;

cout << "r - Rice Krispies" << endl;

cout << "e - Exit" << endl << endl;

cout << "\nEnter a cereal code: ";

cin >> option;

switch (option)

{

case 'w':

//Calculate Cost

discount = iNum2 / 100;

discount = 1 - discount;

total = iNum1 \* w;

total = total \* discount;

cout << setw(COL1) << left << "Boxes:";

cout << setw(COL2) << right << iNum1 << endl;

cout << setw(COL1) << left << "Discount (%):";

cout << setw(COL2) << right << iNum2 << endl;

cout << setw(COL1) << left << "Cereal code:";

cout << setw(COL2) << right << 'w' << endl;

cout << setw(COL1) << left << "Cereal type:";

cout << setw(COL2) << right << "Wheaties" << endl;

cout << setw(COL1) << left << "Purchase amount:";

cout << setw(COL2) << right << total << endl;

break;

case 'c':

//Calculate Cost

discount = iNum2 / 100;

discount = 1 - discount;

total = iNum1 \* c;

total = total \* discount;

cout << setw(COL1) << left << "Boxes:";

cout << setw(COL2) << right << iNum1 << endl;

cout << setw(COL1) << left << "Discount (%):";

cout << setw(COL2) << right << iNum2 << endl;

cout << setw(COL1) << left << "Cereal code:";

cout << setw(COL2) << right << 'c' << endl;

cout << setw(COL1) << left << "Cereal type:";

cout << setw(COL2) << right << "Cheerios" << endl;

cout << setw(COL1) << left << "Purchase amount:";

cout << setw(COL2) << right << total << endl;

break;

case 'r':

//Calculate Cost

discount = iNum2 / 100;

discount = 1 - discount;

total = iNum1 \* r;

total = total \* discount;

cout << setw(COL1) << left << "Boxes:";

cout << setw(COL2) << right << iNum1 << endl;

cout << setw(COL1) << left << "Discount (%):";

cout << setw(COL2) << right << iNum2 << endl;

cout << setw(COL1) << left << "Cereal code:";

cout << setw(COL2) << right << 'r' << endl;

cout << setw(COL1) << left << "Cereal type:";

cout << setw(COL2) << right << "Rice Krispies" << endl;

cout << setw(COL1) << left << "Purchase amount:";

cout << setw(COL2) << right << total << endl;

break;

case 'e':

break;

default :

cout << '\'' << option << '\'' << " is not a valid cereal code.";

}

} while (option != 'w' && option != 'c' && option != 'r' && option != 'e');

// Show application close

cout << "\nEnd of Cereal Suppers" << endl << endl;

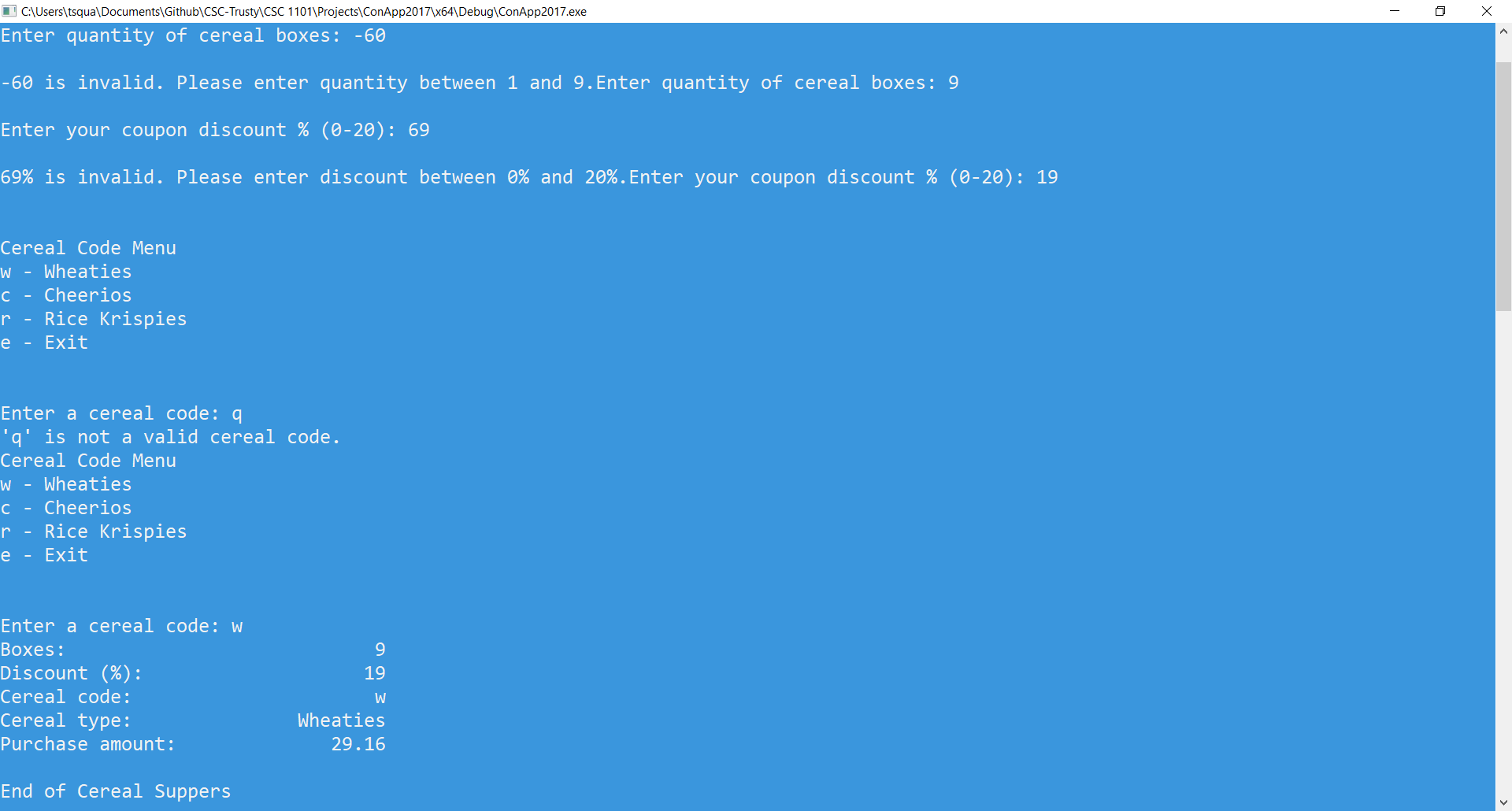
// Pause before application window closes

cout << "Press any key to exit ..." << endl;

\_getch();

}

*[your program output here]\*\**



\* **Copying-and-pasting Visual C++ code to a Word document**

1) From within the Visual C++ program, press **CTRL-A** and press **CTRL-C**.

2) From within the Word document, press **CTRL-V**.

\*\* **Copying-and-pasting Visual C++ console application output to a Word document**

1) From the Visual C++ console, press **ALT-PrintScreen**.

2) From within the Word document, press **CTRL-V**.