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

**Lab 12 – [your name]**

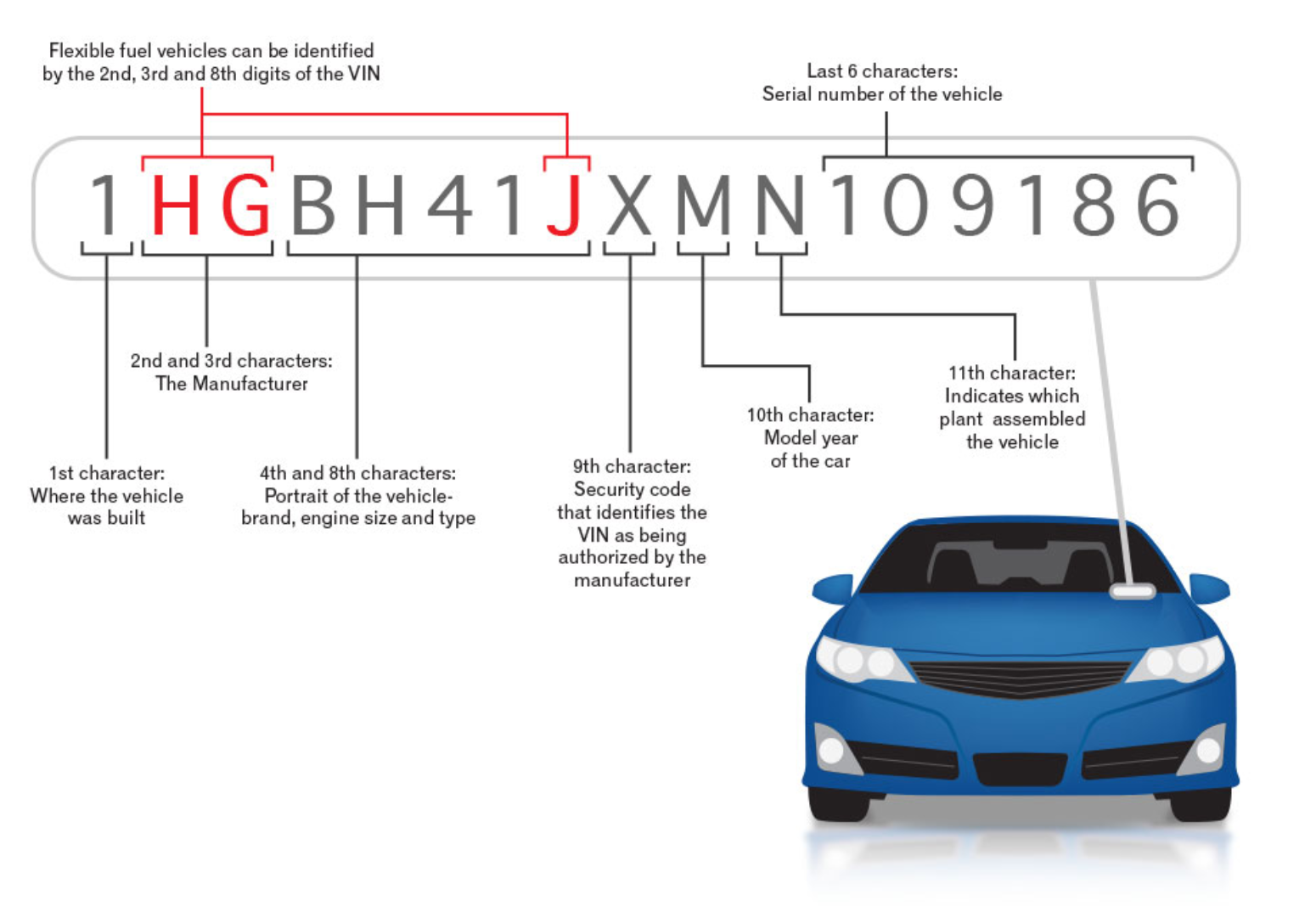
**25 points – Due March 4, 11pm**

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

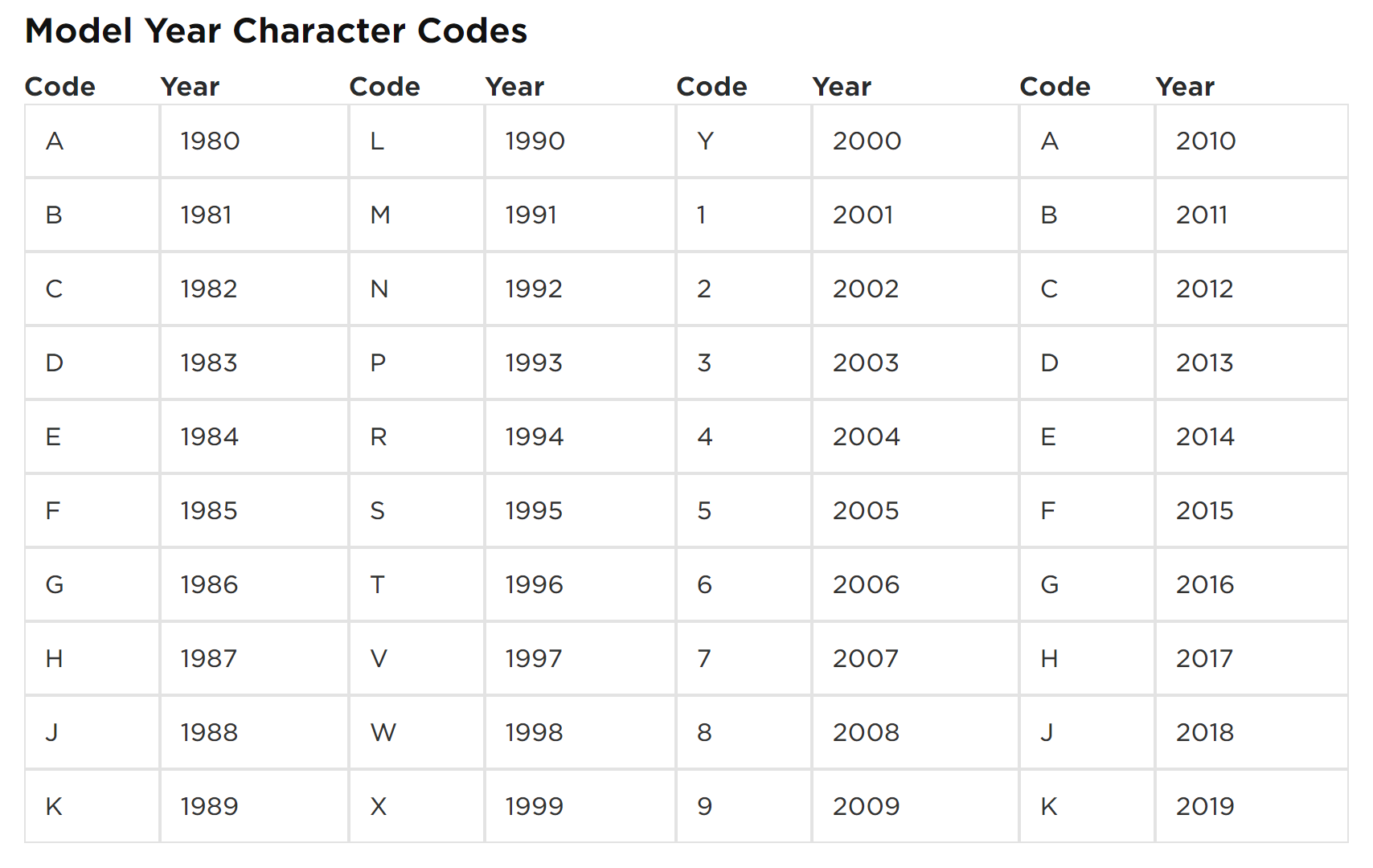
**b)** Type/paste your answers into the document.

**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) [10 points]** You've been hired by *VIN Finders* to write a C++ console application that analyzes vehicle identification numbers (VINs). Download text file **VINs.txt** from Canvas. Place the file in a folder where your development tool can locate it (on Visual Studio, in folder <project-name>\<project-name>). Each line in the file contains one field or token, a VIN number. A VIN has format:



Read one VIN number at a time into a string and determine the model year (character 10) per the following table:



Note that some of the codes map to more than one model year. If that is the case, use the later year. Use a (long) switch statement to determine the year. Use formatted output manipulators (setw, left/right) to print the following two columns (with column headers) for each VIN:

● The left-justified VIN.

● The right-justified model year.

Define constants for the column widths. The start of the output should look like this:

Welcome to VIN Finders

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

Reading lines from file 'VINs.txt' ...

VIN Model year

3B7KF23Z91G223647 2001

…

*[your program code here]\**

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

**//**

**// Title: Welcome to Text File Input - One Line Per Read**

**// Course: CSC 1101**

**// Lab Number: 11-01**

**// Author: Trevor Trusty**

**// Date: 3/1/2019**

**// Description:**

**// This C++ console application demonstrates reading one**

**// line at a time from a text file.**

**//**

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

**#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"**

**const int Col1 = 17;**

**const int Col2 = 12;**

**void vinCase(string line, string year)**

**{**

**cout << setw(Col1) << left << line;**

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

**}**

**int main()**

**{**

**ifstream inFile;**

**int lineCount = 0;**

**string line;**

**const string VIN\_NUMBERS = "VINs.txt";**

**// Show application header**

**cout << "Welcome to Welcome to Text File Input - One Line Per Read!" << endl;**

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

**//Attempt to open file**

**inFile.open(VIN\_NUMBERS);**

**if (!inFile.is\_open())**

**{**

**cout << "Error: could not open file!" << endl;**

**}**

**else**

**{**

**cout << "Reading from file '" << VIN\_NUMBERS << "'..." << endl << endl;**

**vinCase("VIN", "Year");**

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

**while (inFile.good())**

**{**

**getline(inFile, line);**

**switch (line.at(9))**

**{**

**case 'A':**

**vinCase(line, "2010");**

**break;**

**case 'B':**

**vinCase(line, "2011");**

**break;**

**case 'C':**

**vinCase(line, "2012");**

**break;**

**case 'D':**

**vinCase(line, "2013");**

**break;**

**case 'E':**

**vinCase(line, "2014");**

**break;**

**case 'F':**

**vinCase(line, "2015");**

**break;**

**case 'G':**

**vinCase(line, "2016");**

**break;**

**case 'H':**

**vinCase(line, "2017");**

**break;**

**case 'J':**

**vinCase(line, "2018");**

**break;**

**case 'K':**

**vinCase(line, "2019");**

**break;**

**case 'L':**

**vinCase(line, "1990");**

**break;**

**case 'M':**

**vinCase(line, "1991");**

**break;**

**case 'N':**

**vinCase(line, "1992");**

**break;**

**case 'P':**

**vinCase(line, "1993");**

**break;**

**case 'R':**

**vinCase(line, "1994");**

**break;**

**case 'S':**

**vinCase(line, "1995");**

**break;**

**case 'T':**

**vinCase(line, "1996");**

**break;**

**case 'V':**

**vinCase(line, "1997");**

**break;**

**case 'W':**

**vinCase(line, "1998");**

**break;**

**case 'X':**

**vinCase(line, "1999");**

**break;**

**case 'Y':**

**vinCase(line, "2000");**

**break;**

**case '1':**

**vinCase(line, "2001");**

**break;**

**case '2':**

**vinCase(line, "2002");**

**break;**

**case '3':**

**vinCase(line, "2003");**

**break;**

**case '4':**

**vinCase(line, "2004");**

**break;**

**case '5':**

**vinCase(line, "2005");**

**break;**

**case '6':**

**vinCase(line, "2006");**

**break;**

**case '7':**

**vinCase(line, "2007");**

**break;**

**case '8':**

**vinCase(line, "2008");**

**break;**

**case '9':**

**vinCase(line, "2009");**

**break;**

**}**

**lineCount++;**

**}**

**}**

**inFile.close();**

**// Show application close**

**cout << "\nEnd of Welcome to Text File Input - One Line Per Read" << endl << endl;**

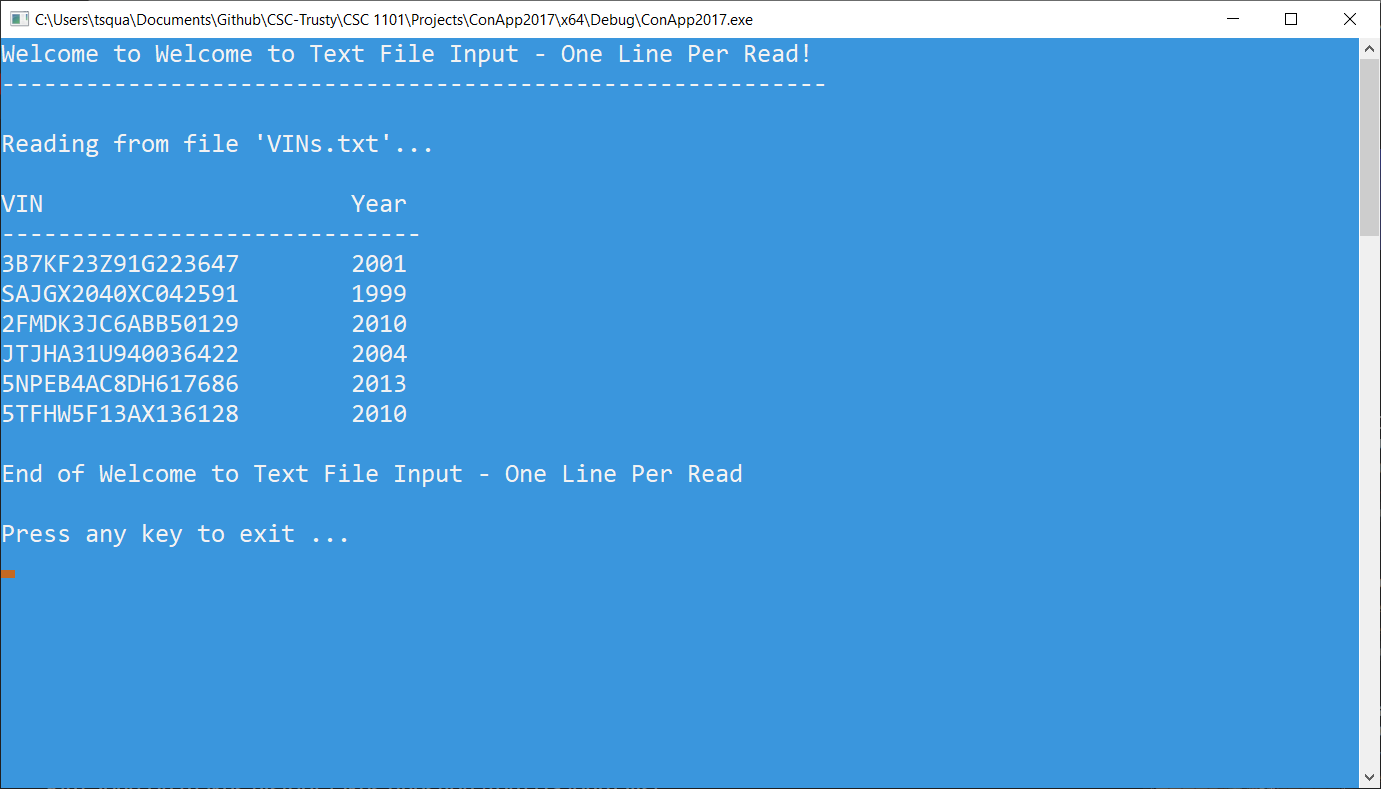
**// Pause before application window closes**

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

**\_getch();**

**}**

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



**2) [15 points]** You’ve been hired by *Lizard Lenders* to write a C++ console application that determines the number of months to repay a car loan. Use a validation loop to prompt for and get from the user the car loan amount in the range $1,000-5,000. Then use a validation loop to prompt for and get from the user the monthly payment in the range $100-500. Then use a validation loop to prompt for and get from the user the annual interest rate in the range 2-5%. Convert the valid rate to a monthly rate and to a decimal number with formula:e

rate / 12 / 100

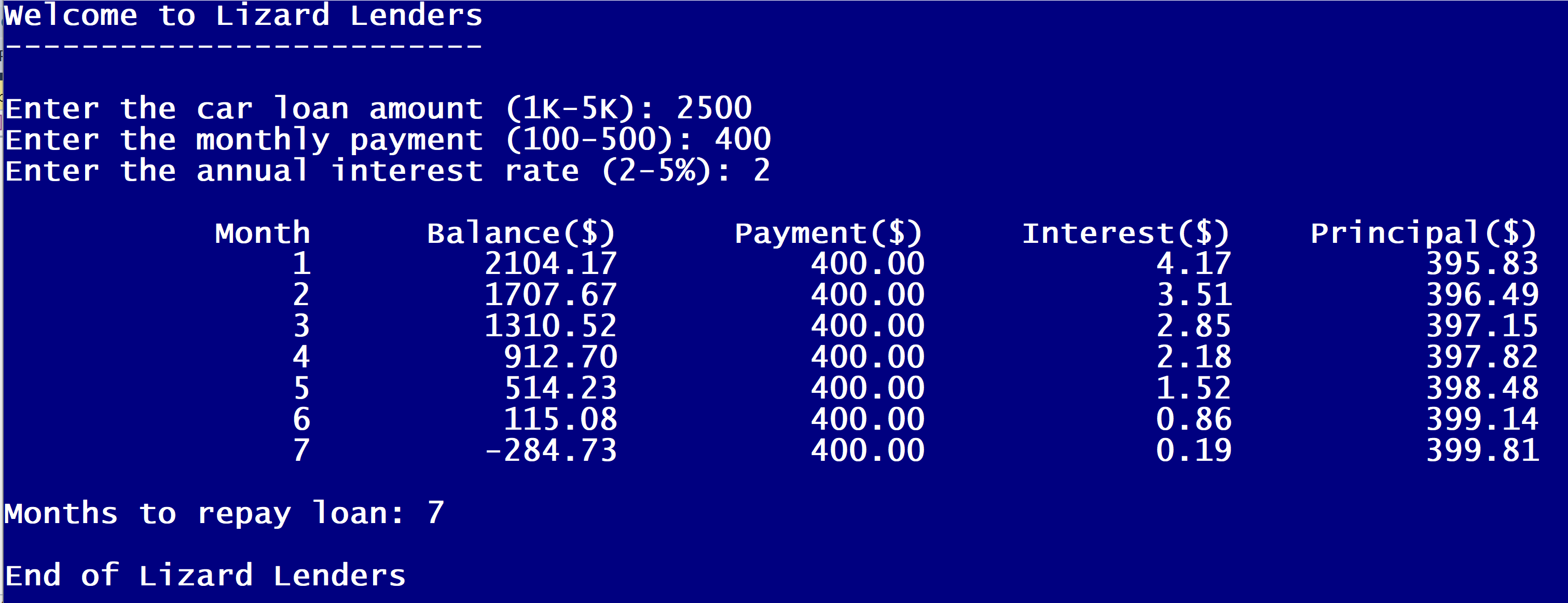
Loop to show the loan being repaid and keep track of the number of months it takes to do so. Before the loop, initialize the balance to the loan amount and months to zero. Within the loop, calculate the interest portion of each monthly payment, the principal portion of the payment, and the new balance with formulas:

interest = balance \* rate;

principal = payment - interest;

balance = balance - principal;

Continue to loop while the balance is greater than zero. After the loop, print the number of months it took to repay the loan. Format the output in columns and show all real numbers to two decimal places. Here is sample output:



Run the program with the following inputs:

|  |  |  |
| --- | --- | --- |
| Loan amount | Monthly payment | Annual interest rate |
| 6,000 | ~ | ~ |
| 3,000 | 700 | ~ |
|  | 256.82 | 8 |
|  |  | 5 |
| 4,000 | 292.91 | 4 |
| 5,000 | 319.18 | 3 |

*[your program code here]*

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

**//**

**// Title: Lizard Lenders**

**// Course: CSC 1101**

**// Lab Number: 11-02**

**// Author: Trevor Trusty**

**// Date: 3/1/2019**

**// Description:**

**// This C++ console application uses a for loop to calculate**

**// how many months to repay user's car loan.**

**//**

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

**#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"**

**void vLoop(double money, int x, int y, string type)**

**{**

**while (money < x || money > y)**

**{**

**cout << '$' << money << " is an incorrect " << type << " amount." << endl;**

**cout << "\nEnter " << type << " ($) amount between $ " << x << " and $" << y << ": ";**

**cin >> money;**

**cout << endl;**

**}**

**}**

**int main()**

**{**

**//Declare variables**

**double loan, monthPay, rate, balance, interest, principal, month;**

**//Declare constants**

**const int W = 14; //column width**

**// Show application header**

**cout << "Welcome to Lizard Lenders!" << endl;**

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

**//Prompt for and get loan amount**

**cout << "Enter car loan ($) between $1000 and $5000: ";**

**cin >> loan;**

**//Check for incorrect loan, reprompt**

**vLoop(loan, 1000, 5000, "loan");**

**//Prompt for and get monthly payment**

**cout << "Enter current monthly payment ($) between $100 and $500: ";**

**cin >> monthPay;**

**//Check for incorrect amount, reprompt**

**vLoop(monthPay, 100, 500, "payment");**

**//Prompt for and get interest factor**

**cout << "\nEnter annual interest (%) between 2% and 5%: ";**

**cin >> rate;**

**cout << endl;**

**//Check for incorrect value, reprompt**

**while (rate < 2 || rate > 5)**

**{**

**cout << rate << "% is an incorrect interest value." << endl;**

**cout << "\nEnter annual interest (%) between 2% and 5%: ";**

**cin >> rate;**

**cout << endl;**

**}**

**//Convert rate to decimal**

**rate = rate / 12 / 100;**

**balance = loan;**

**//Start table-header**

**cout << setw(W) << right << "Month";**

**cout << setw(W) << right << "Balance($)";**

**cout << setw(W) << right << "Payment($)";**

**cout << setw(W) << right << "Interest($)";**

**cout << setw(W) << right << "Principal($)" << endl;**

**//loop until balance is paid**

**for (int m = 1; balance > 0; m++)**

**{**

**interest = balance \* rate;**

**principal = monthPay - interest;**

**balance = balance - principal;**

**cout << setw(W) << right << m;**

**cout << setw(W) << right << balance;**

**cout << setw(W) << right << monthPay;**

**cout << setw(W) << right << interest;**

**cout << setw(W) << right << principal << endl;**

**month = m;**

**}**

**cout << "\nMonths to repay loan: " << month << endl;**

**// Show application close**

**cout << "\nEnd of Lizard Lenders" << endl << endl;**

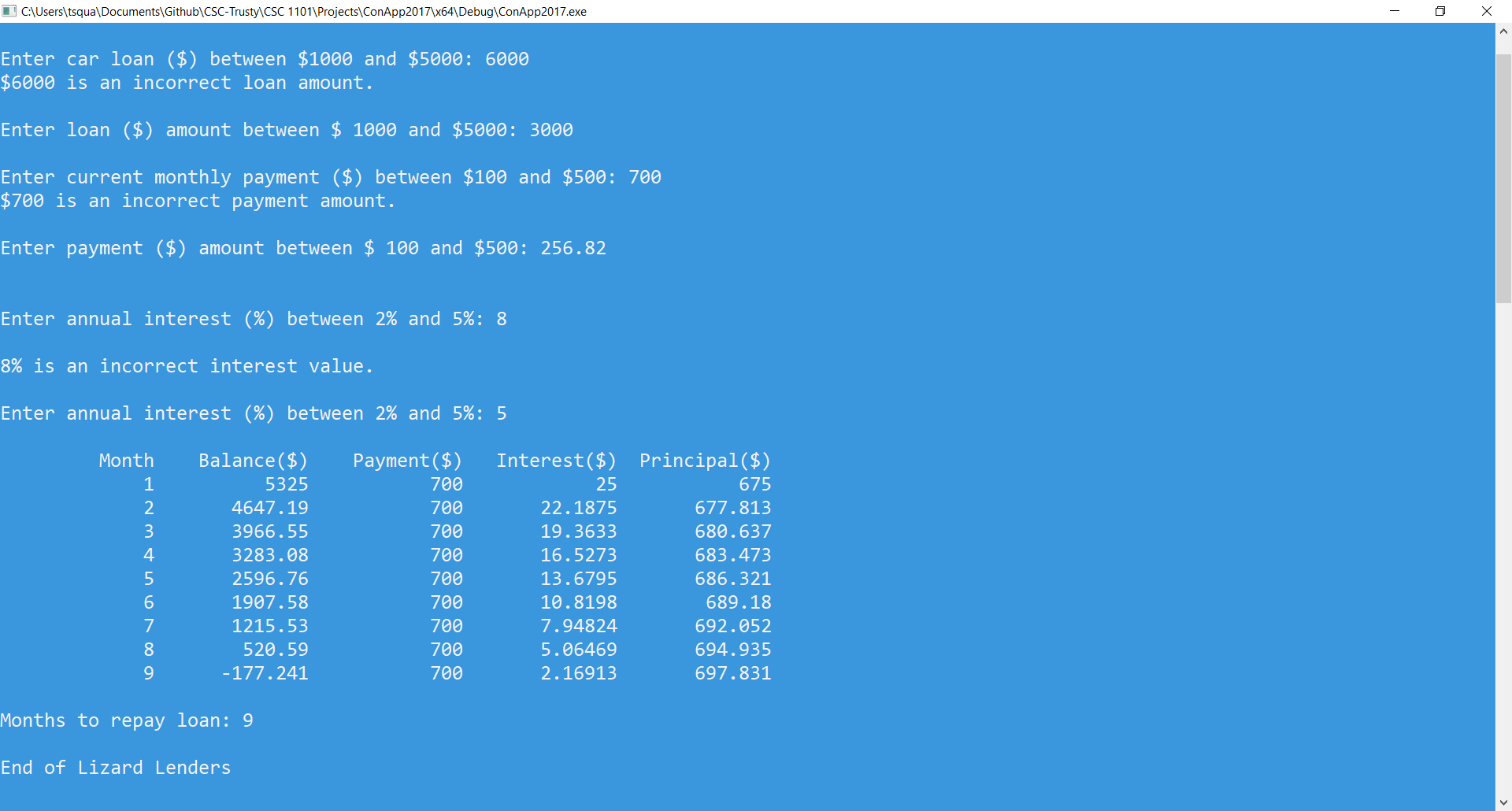
**// Pause before application window closes**

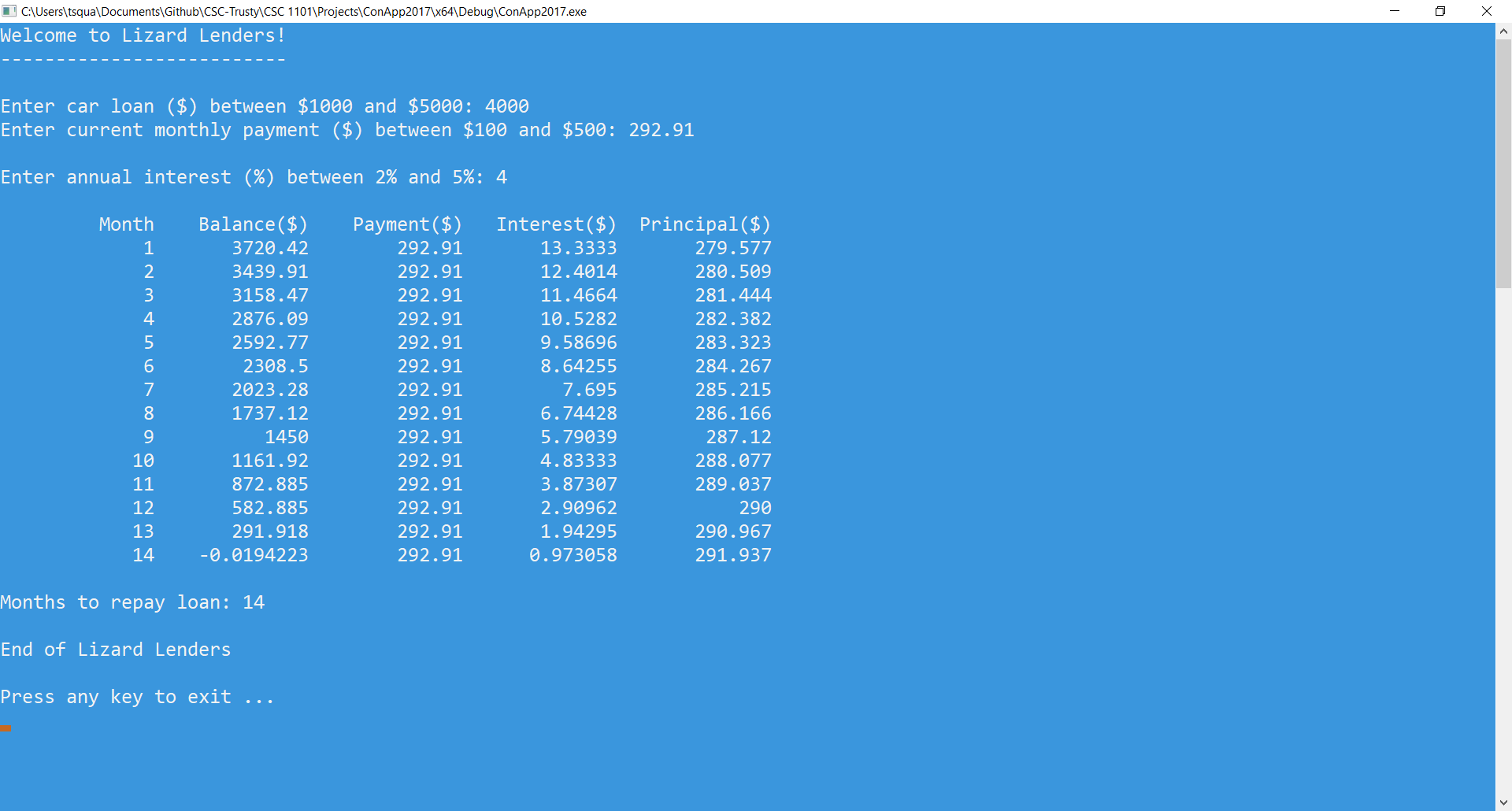
**cout << "Press any key to exit ..." << endl;**

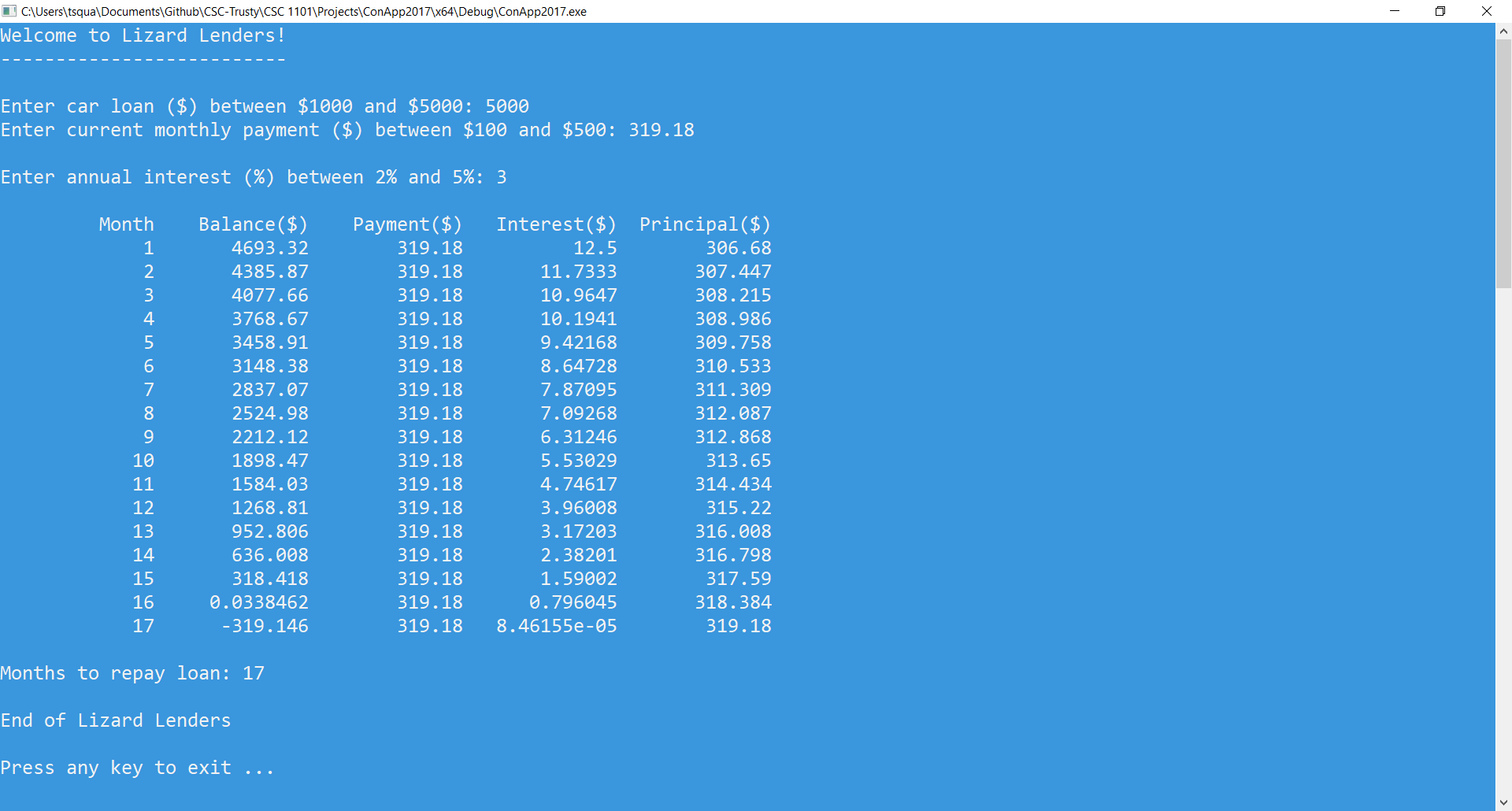
**\_getch();**

**}**

*[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**.