Course: CSC340.05

Student: Kyle Harvey, SFSU ID: 915139815

Teammate:

Assignment Number: 03

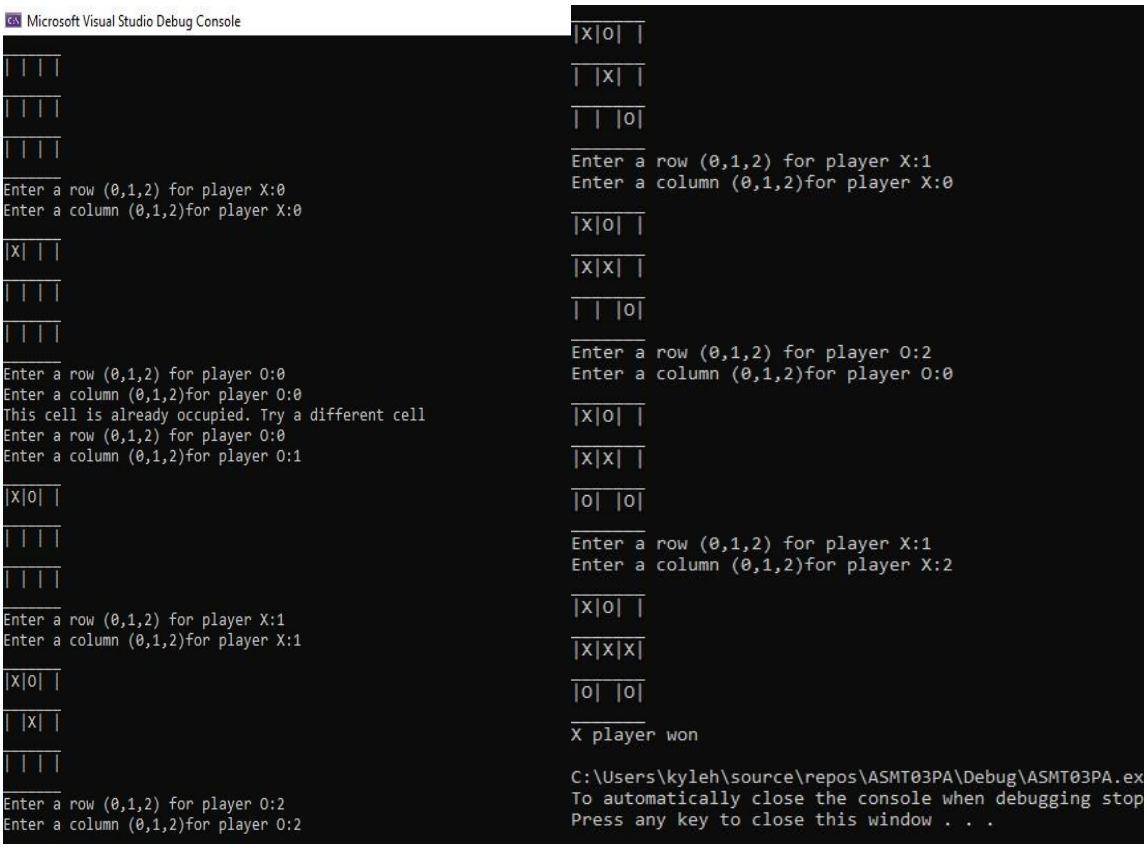
Assignment Due Date & Time: 10-13-2020 at 11:55 PM

Question **Part A**: Please implement a basic version of Tic Tac Toe:

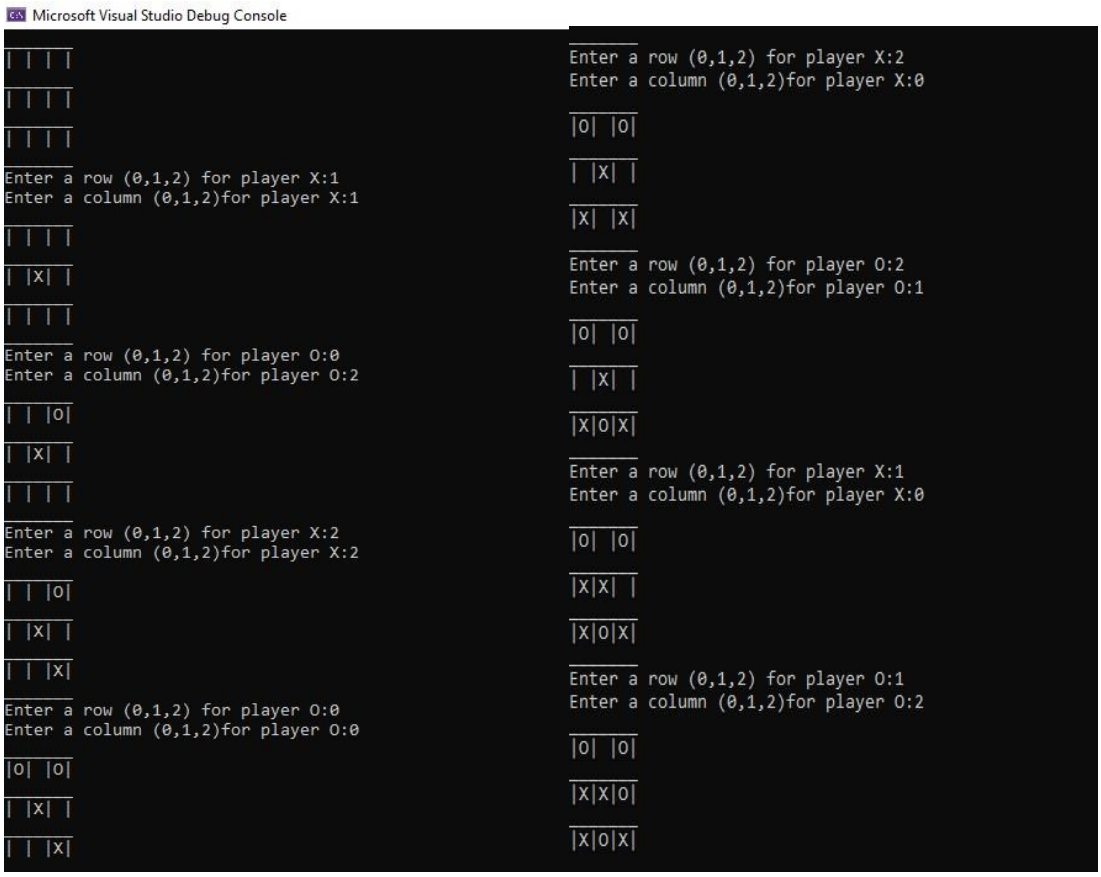
1. Function main and function headers are provided. Please implement the functions and do not change the main.

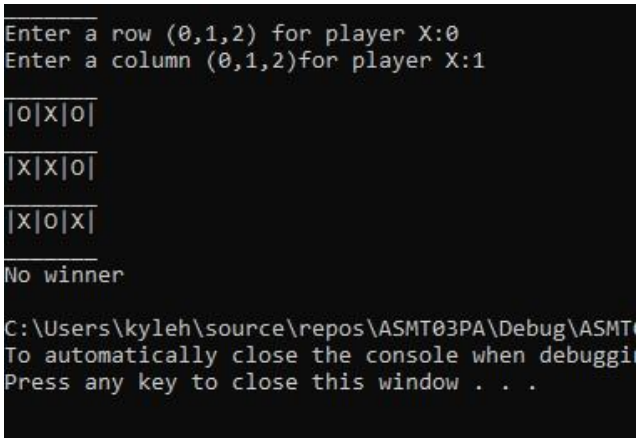
2. Our program must produce identical output: Assignment-03_PA_Run1.txt and Assignment-03_PA_Run2.txt

- Answer/Summary/Discussion: **Run1 Output:**



Run2 Output:





Question **Part B** #1: Credit card numbers follow certain patterns. A credit card number must have between 13

and 16 digits. The starting numbers are: 4 for Visa cards, 5 for MasterCard cards, 37 for American Express cards, and 6 for Discover cards.

Example: Validating 4 3 8 8 5 7 6 0 1 8 4 0 2 6 2 6

a) Double every second digit from right to left. If doubling of a digit results in a two-digit number, add the two digits to get a single digit number.

b) Now add all single-digit numbers from Step a:

4 + 4 + 8 + 2 + 3 + 1 + 7 + 8 = 37

c) Add all digits in the odd places from right to left in the card number:

6 + 6 + 0 + 8 + 0 + 7 + 8 + 3 = 38

d) Sum the results from Step b and Step c:

37 + 38 = 75

e) If the result from Step d is divisible by 10, the card number is valid; otherwise, it is invalid.

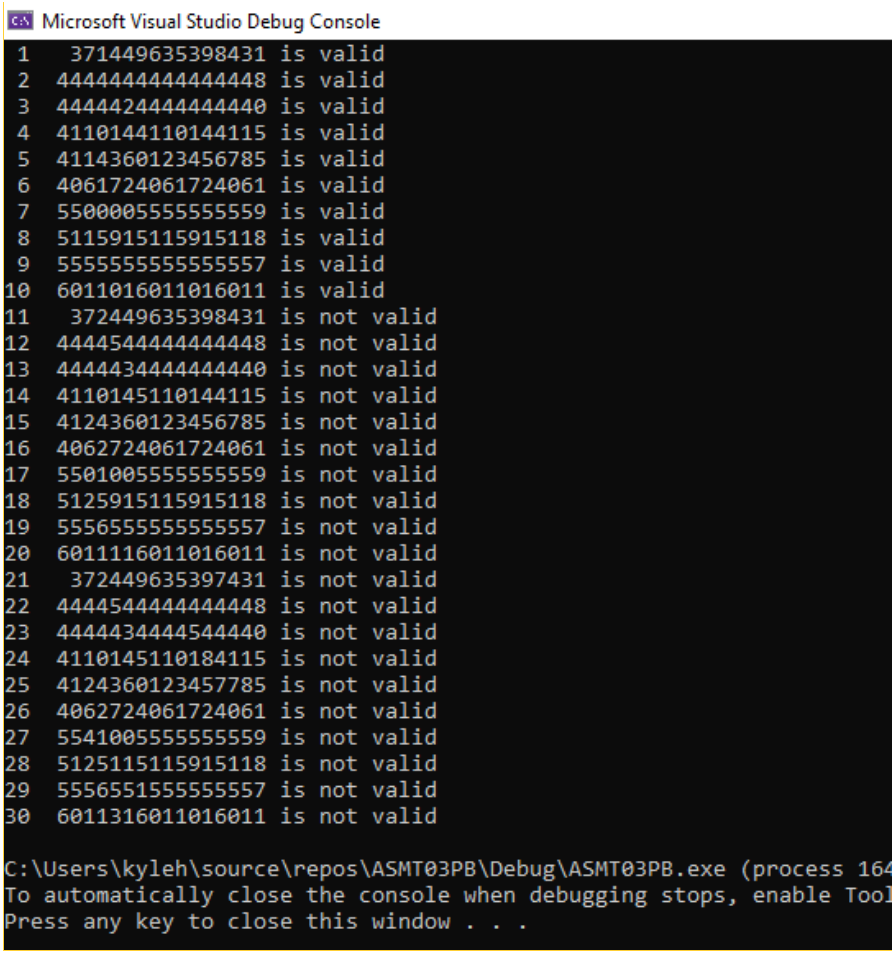
Please implement Credit Card Number Validation:

1. Function main is provided. Please implement isvalidcc and other functions which you may add to the program.

2. Please do not change function main

3. Your program must produce identical output: Assignment-03_PB_Run.pdf

- Answer/Summary/Discussion:



Question **Part C** #1: Our satisfied clients are back to ask us to implement another interactive dictionary. Our dictionary takes input from users and uses the input as search key to look up values associated with the key. Requirements: - Coding: No hard coding, https://en.wikipedia.org/wiki/Hard_coding. Please think about Dynamic and Scalable. - Data Source: A text file, Data.CS.SFSU.txt . Please think about Software Deployment and Usability. - Data Structure: Use existing data structure(s) or create new data structure(s) to store our dictionary's data. Each keyword, each part of speech, and each definition must be stored in a separate data field. Do not combine them such as storing three parts in one String. - Data Loading: When our program starts, it loads all the original data from the Data Source into our

dictionary's data structure. The data source file is opened once and closed once per run. It must be closed as soon as possible. It must be closed before our program starts interacting with users. - User Interface: A program interface allows users to input search keys. This interface then displays returned results. Our program searches the dictionary's data (not the data source text file) for values associated with the search keys. - Identical Output: Our program's output must be identical to the complete sample run's output: Assignment-03_PC_Run.pdf

Program Analysis to Program Design, 10 points. Please think about Interviews. In at least 1 full page, please focus on the differences/improvements you are making in this C++ version of the program in comparison to your previous Java version while explaining the following in detail: ▪ Your analysis of the provided information and the provided complete sample output. Please think about Clients and Sales. ▪ What problem you are solving. Please explain it clearly then define it concisely. Please think about Problem Solving and Interviews. ▪ How you store data in enum objects. And why. Please think about Data Structures and Data Design. ▪ Which data structures you use/create for your dictionary. And why. Please think about Data Structures and Data Design.

- Answer/Summary/Discussion:

For this program assignment, first I needed to understand what exactly were the requirements. This is important when it comes to the real world working with clients and selling a product, we need know what exactly what they want in order to provide the best service possible. For this program although we're using a different language for the output of the, the object for us to create an interactive dictionary by taking inputs from users and using the input as a search key to look up values associated with the key. There are some restrictions and requirements that we must follow as if a client have needs to meet. Those restrictions and requirements included, no

hard coding, the data source is stored this time in a text file already created for our program to access. It also requires an existing data structure or creating new data structures to store the dictionary's data. The program also starts and loads all the original data from the Data text file into our dictionary's data structure. The user interface needs to allow users to input search keys to find what they're looking for in the outputs. Last but not least there's an output that needs to be matched up as to meet the client's expectations. Once again when looking at my program, there are some issues that don't entirely meet the outputs as expected containing errors. The issues lie within any input after the third word, it's supposed show other outputs regarding to the key word but it doesn't process or recognize what it is and ignores it. Although I have my program being able to pick up all the keywords by itself, many of those keywords and what its suppose to output, it seems within the loops or arrays it doesn't recognize or store it. I was also missing implementing a keyword and definition word count. I left a placement for where I would have the program be able to count all the words and definition in the word file but I left it set as shown in the output example we were supposed to follow. There's mostly likely a way to improve my program to having it working to its extent of producing the correct output if I was able to spend more time on it but as of now there are some issues. Another way this program could have been improved or better is if the program was worked on with a partner or in a real world scenario in a team to go over the program and fill what was missing or how it could have optimized. The data structure that I used for creating this program were lists, multimap, vector, strings, algorithm and constructors. Overall this assignment was a good way to become familiar with C++ by recreating the same program output from the previous assignment still using data structure, data design, and all the available tools provided in C++.

Question **Part C** #2: Program Implementation, 80 points.

Please think about Interviews.

▪ Implement your program to meet all the requirements.

▪ In your assignment report, demonstrate your program to your grader/client.

▪ Does your program work properly?

▪ How will you improve your program?

- Answer/Summary/Discussion:

```
Search: book NOUN distinct reverse
 |
 <Please enter a search key (and a part of speech).>
 |
Search: placeHOLDER
 |
 placeholder [noun] : - > To be updated...
 placeholder [adjective] : - > To be updated...
 placeholder [adjective] : - > To be updated...
 placeholder ad[verb] : - > To be updated...
 placeholder conjunction - > To be updated...
 placeholder interjection - > To be updated...
 placeholder [noun] : - > To be updated...
 placeholder [noun] : - > To be updated...
 placeholder preposition - > To be updated...
 placeholder pro[noun] : - > To be updated...
 placeholder [verb] : - > To be updated...
 |
Search: placeholder distinct
 |
 <2nd argument must be a part of speech.>
 |
Search: placeHolder adjective reverse
 |
 <Please enter a search key (and a part of speech).>
 |
Search: distinCT DISTINot
 |
 <2nd argument must be a part of speech.>
 |
Search: distinctnoun distinct REVERSE
 |
 <Please enter a search key (and a part of speech).>
 |
Search: distinct noun distinct REVERSE
 |
 <Please enter a search key (and a part of speech).>
 |
Search: reverse reverse
 |
 <2nd argument must be a part of speech.>
 |
```

```
Search: reverse reverse
 |
 <2nd argument must be a part of speech.>
 |
Search: reverse
 |
 reverse [verb] : - > Change something to opposite.
 reverse [verb] : - > go back.
 reverse [verb] : - > turn something inside out.
 reverse [verb] : - > revoke ruling.
 reverse [adjective] : - > Opposite to usual or previous arrangement.
 reverse [adjective] : - > On back side.
 reverse [noun] : - > The opposite.
 reverse [noun] : - > Change to opposite direction.
 reverse [noun] : - > A dictionary program's parameter.
 reverse [noun] : - > To be updated...
 reverse [verb] : - > To be updated...
 reverse [noun] : - > To be updated...
 reverse [noun] : - > To be updated...
 reverse [verb] : - > To be updated...
 reverse [noun] : - > To be updated...
 |
Search: reverse distinct reVERSE
 |
 <Please enter a search key (and a part of speech).>
 |
Search: reverse ok
 |
 <2nd argument must be a part of speech.>
 |
Search: reverse noun ok
 |
 <Please enter a search key (and a part of speech).>
 |
Search: reverse noun ok distinct
 |
 <Please enter a search key (and a part of speech).>
 |
Search: reverse adverb
 |
 <2nd argument must be a part of speech.>
 |
Search: facebook
 |
```

```
 |
 <2nd argument must be a part of speech.>
 |
Search: facebook
 |
 <Not found.>
 |
Search: !Q

----- THANK YOU-----

C:\Users\kyleh\source\repos\ASMT03PC\Debug\ASMT03PC.exe (process 19996) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```