Assignment Report for Assignment 3

Course and Section	CSC 340-01 TIC
Assignment Name	Assignment 3
Due Date and Time	07-09-2023 at 11:55 PM
First Name and Last Name	Juan Segura Rico
SFSU Email Account	jsegurarico@sfsu.edu
First Name and Last Name of Teammate	Jonathan Curimao
SFSU Email Account of Teammate	jcurimao1@sfsu.edu

PART A

Question Description and Analysis:

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.

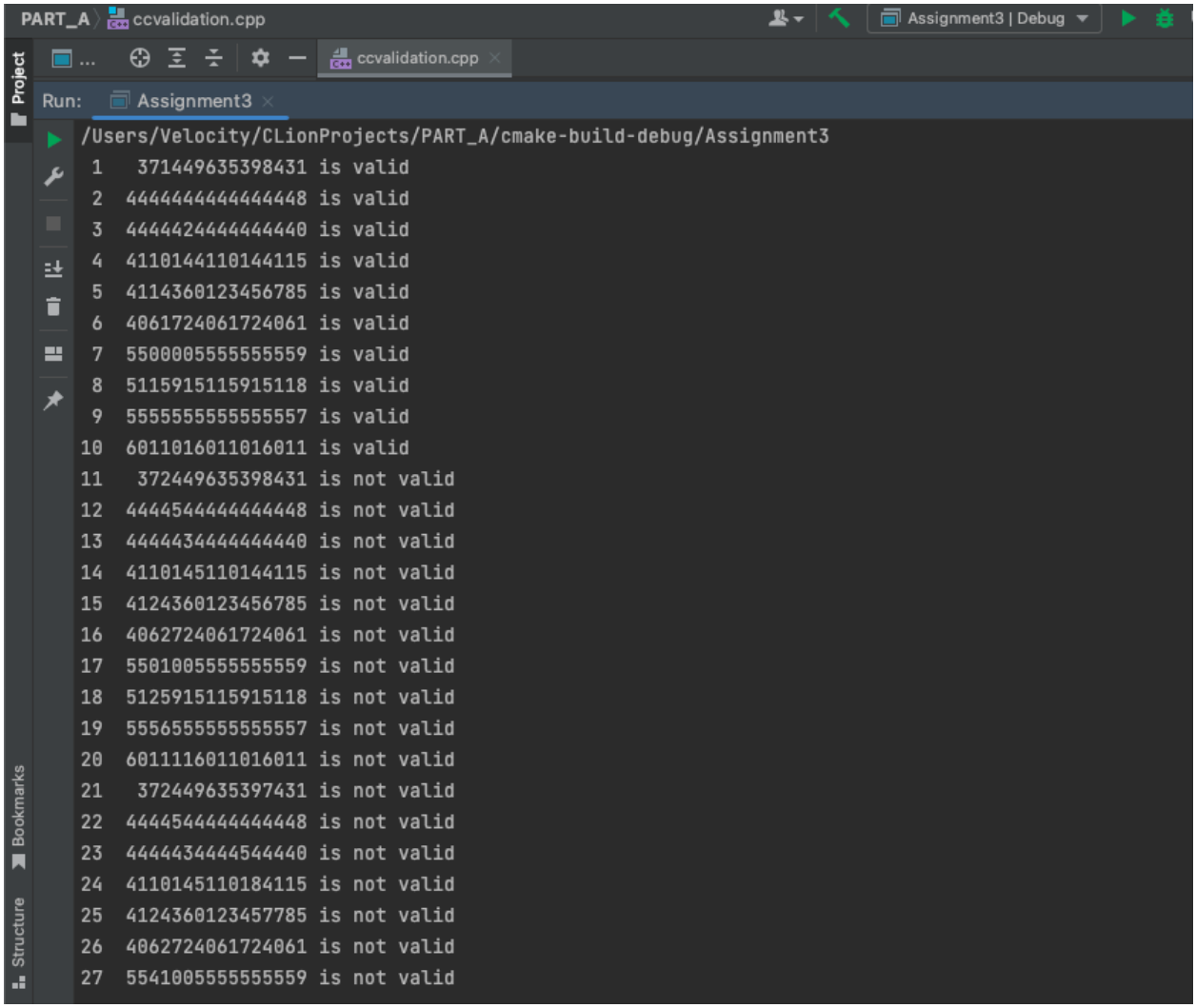
2. Our program must produce identical output

Answer:

(Provided in screenshot section)

Screenshots of Outputs and Explanation:

These screenshots show what I accomplished…


PART_A
ccvalidation.cpp
Assignment3 | Debug
Project
ccvalidation.cpp
Run:
Assignment3
/Users/Velocity/CLionProjects/PART_A/cmake-build-debug/Assignment3
1 371449635398431 is valid
2 4444444444444448 is valid
3 4444424444444440 is valid
4 4110144110144115 is valid
5 4114360123456785 is valid
6 4061724061724061 is valid
7 5500005555555559 is valid
8 5115915115915118 is valid
9 5555555555555557 is valid
10 6011016011016011 is valid
11 372449635398431 is not valid
12 4444544444444448 is not valid
13 4444434444444440 is not valid
14 4110145110144115 is not valid
15 4124360123456785 is not valid
16 4062724061724061 is not valid
17 5501005555555559 is not valid
18 5125915115915118 is not valid
19 5556555555555557 is not valid
20 6011116011016011 is not valid
21 372449635397431 is not valid
22 4444544444444448 is not valid
23 4444434444544440 is not valid
24 4110145110184115 is not valid
25 4124360123457785 is not valid
26 4062724061724061 is not valid
27 5541005555555559 is not valid
Bookmarks
Structure

(My output)

PART B

Question Description and Analysis:

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

Answer:

(Provided in screenshot section)

Screenshots of Outputs and Explanation:

These screenshots show what I accomplished…

```
-------------
| O | X | X |
-------------
|   | O | O |
-------------
|   | X | X |
-------------
Enter a row (0, 1, 2) for player O   : 0
Enter a column (0, 1, 2) for player O: 1
This cell is already occupied.  Try a different cell.
Enter a row (0, 1, 2) for player O   : 0
Enter a column (0, 1, 2) for player O: 2
This cell is already occupied.  Try a different cell.
Enter a row (0, 1, 2) for player O   : 1
Enter a column (0, 1, 2) for player O: 1
This cell is already occupied.  Try a different cell.
Enter a row (0, 1, 2) for player O   : 1
Enter a column (0, 1, 2) for player O: 0
-------------
| O | X | X |
-------------
| O | O | O |
-------------
|   | X | X |
-------------
O player won

Process finished with exit code 0
```

(Sample game I ran)

PART C

Question Description and Analysis:

Our satisfied clients are back to ask us to implement another interactive dictionary. Our dictionary takes input from users and uses the input as a search key to look up values associated with the key.

1. 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 from the external text file. 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.

2.

▪ 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?

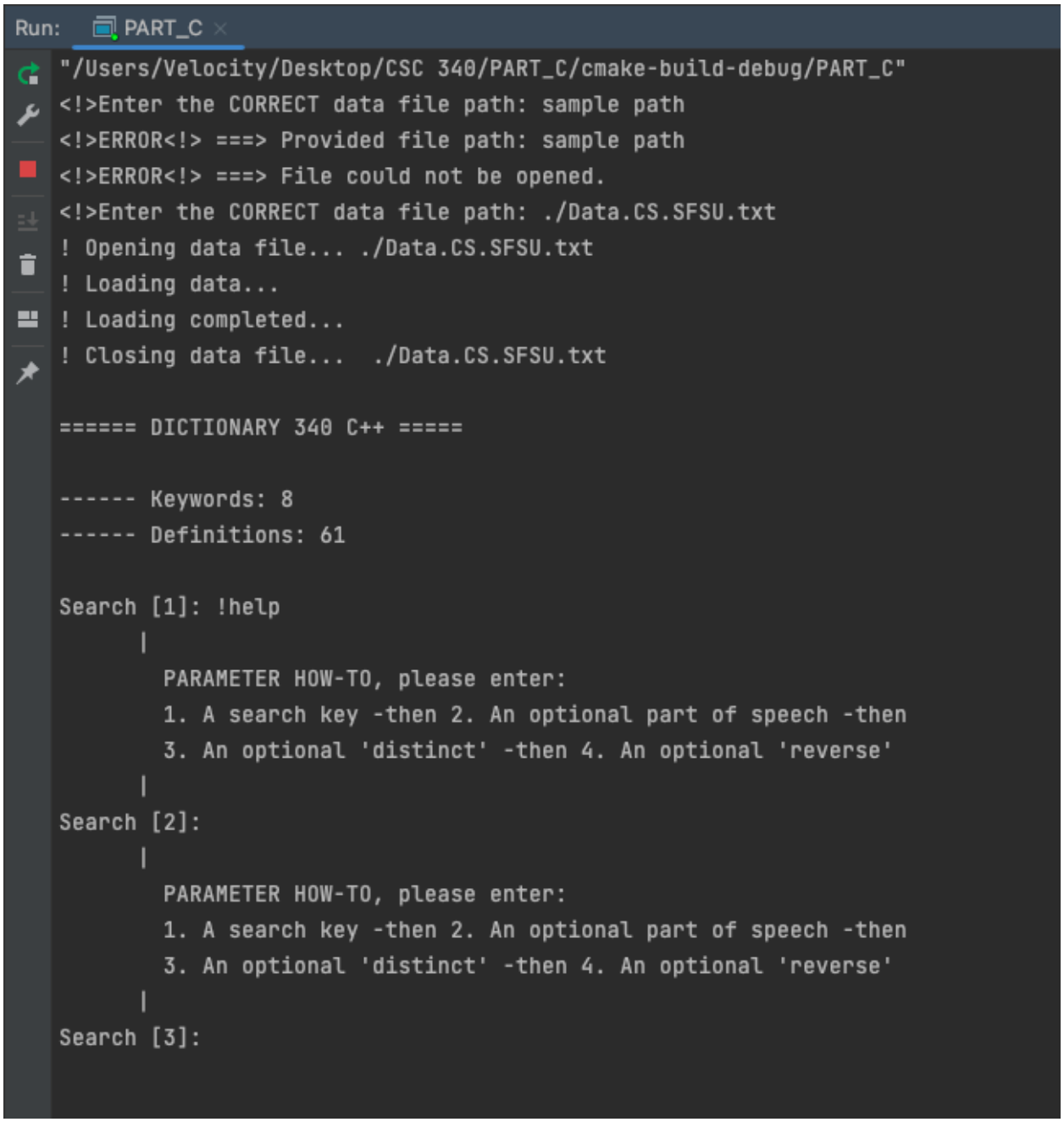
Answer: (Worked with Jonathan on this program)

1. In this program, we are asked to make an interactive Dictionary, like the one in Java, but this time we implement a txt file stored with our keywords. In order for our txt file to work/implement properly, I added it to my files directory… After that, the txt file loaded up without problems when running the program. Some data structures used include vectors, multimap, and list. We use these to store the keyword and its corresponding part-of-speech and definition pairs. A multimap is chosen because it allows multiple entries with the same keyword while preserving the order of insertion. This enables efficient lookup and retrieval of definitions based on the keyword. The list is used to store the matching definitions found during the search. A list is chosen because it allows efficient insertion and removal of elements at both ends. It also preserves the order of insertion, which is important when displaying the output.

2. After working on the program. It fully functions, and for the most part, it works properly. If there's any problems, they are very minor things. We tried fixing the order of some of the output, but it would mess up other parts of the code, so we weren't successful. Improvements we could make to the program, include that, and making our code more simple.

Screenshots of Outputs and Explanation:

Example output of our code:



```
Search [3]: aRRow
      |
      arrow [noun ] :  Here is one arrow: <IMG>
      |
Search [4]: adverb
      |
     adverb [noun ] :  Adverb is a word that adds more information about place, time, manner, cause or degree to
 a verb, an adjective, a phrase or another adverb.
      |
Search [5]: noun noun reverse
      |
noun [noun ] :  Noun is a word that refers to a person, (such as Ann or doctor), a place (such as Paris or city)
 or a thing, a quality or an activity (such as plant, sorrow or tennis).
      |
Search [6]: cSc220
```