1. Dictionary file is located in bin/debug folder. Its name is words.txt. This file has been downloaded from the internet and some contents have been trimmed while some have been added to make the file exhaustive but easy to navigate at the same time.
2. The contents of the file are in Uppercase
3. There are approximately 82,000 words in the text file
4. A timer is attached in the program which runs in a separate thread, so that it does not effect the performance of the process.
5. The board which consists of 16 letters representing a dice is randomly created. The dice is developed as per the given specification.
6. Except for “Q” every letter is represented as it is, “Q” is listed as “Qu”
7. As the timer expires the output is written into the console as well as in the file titled “latestOutput.txt” inside the bin/Debug folder

Brief Algorithm:

Step 1: Initialize the board (random function used) which is a two dimensional string array. Character array could not be taken as “Q” is represented as “Qu”

Step 2: Initialize a dictionary i.e. load the contents from words.txt file into the dictionary object

Step 4: Initialize a stringSequence into empty string

Step 5: Pick a letter in the board.

Step 6: add the letter into the stringsequence

Step 5: Check if there is anyword beginning from the string sequence picked in the dictionary. If yes continue and put it in a string sequence else go to step 4

Step 6: Move to the next neighbor, appending it to the stringSequence and push its index position (row and column) into parent list to keep the track of the pieces which has already been visited so that we do not visit that again and get into infinite loop.

Step 7: Check if there is anyword beginning from this new string sequence in dictionary. If yes assign current pointer to this new neighbor and repeat 6. If no remove its position from the parent list and move to next neighbor.

Step 8: if the string sequence in step 7 is a valid string in itself push it to the list of valid string. Dictionary object has been used for this purpose to avoid duplication of the words.

Step 8: Repeat through Step 5 until whole board has been scanned.

**Limitations**

English language has a very huge vocabulary and 80,000 words is a very small subset of the entire language.

It has been observed that the program takes approximately 2 minutes to go to find all possible words. If the dictionary is made larger, the program might itself not be able to find the list of all words within 3 minutes.

The interface is poor as the program is not developed for commercial purpose but just as a coding assignment given to the developer during an interview.

The program is only a 4X4 board which can be enlarged.

**Improvements**

Letters can be moved into database so that querying will be fast enough.

Multiple threads can be assigned to calculate the combination of letters to make the program faster.

Interface can be changed.