**Context/Question:**  *Item 3 – Document Search – Run a performance test that does 2M searches with random search terms, and measures execution time. Which approach is fastest? Why?*

As per the requirement, three different approaches were used to implement the Searches, as follows:

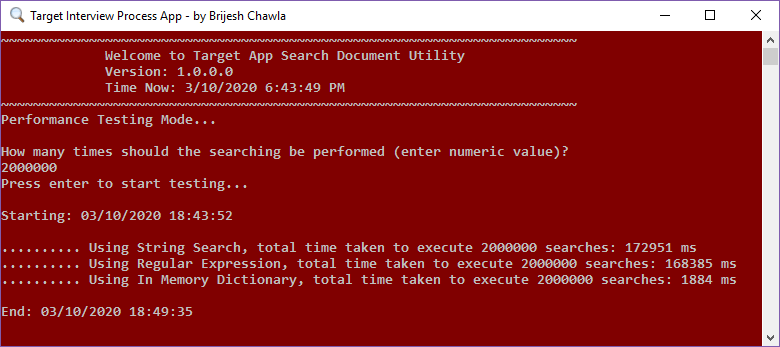
1. String Match
2. Regular Expression
3. Preprocessed content – this approach could be done through database and indexing the data, however, for this comparison, the data is preprocessed and loaded into the Dictionary (Key-Value pair) and utilized

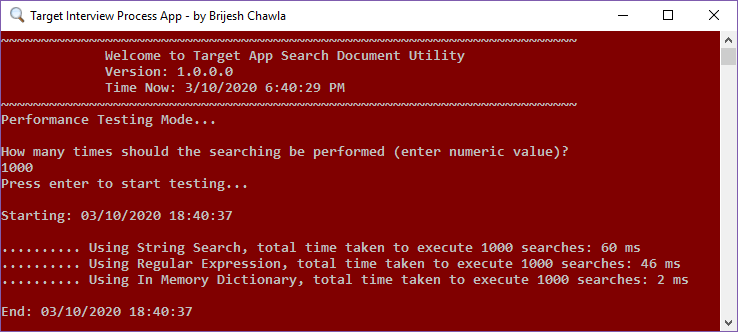
**3rd approach (the preprocessed data) results in fastest performance (as can be seen by the execution times – screen shots below).**

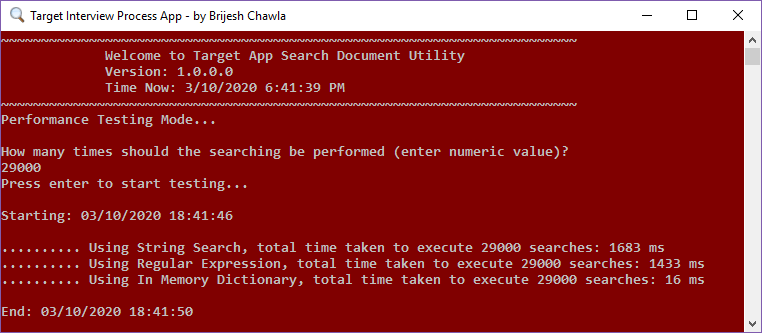
Reason for that is the in-memory execution and key lookup. Data is already processed and all the words and their existence counts are already identified and loaded into the Dictionary (Hashtable) object. When users query the existence of the word, the execution is fastest because it’s just a matter of looking up in the Dictionary if the value exists or not; whereas, the string match and the regular expression approaches have to internally traverse through the string/text to find the matching text and hence it takes more time comparatively.

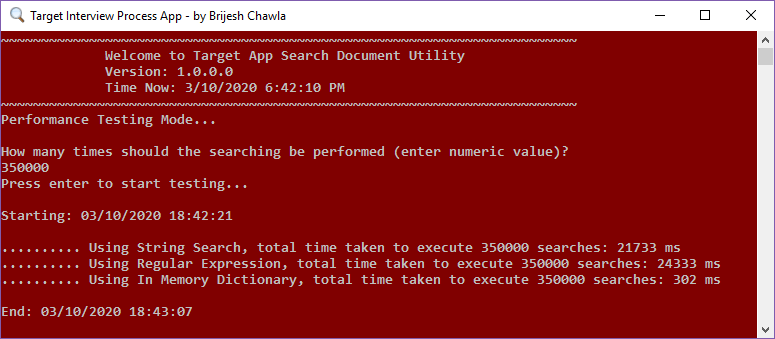
**Additional Notes: *The solution (logic/code) I have provided is based on the assumption of exact word match. The searching could be extend to stem search easily, such as when as user search for word “dream”, the stem search could search for “dreams”, “dreaming”, etc.***

**Following are Screenshots Results from Performance Testing**









**Following are Screenshots Results from System Testing – Individual Method Search Results**

*(This page is left intentionally blank so that all three method call result screenshots could fit in single page for better understanding – screenshots starts from next page)*

