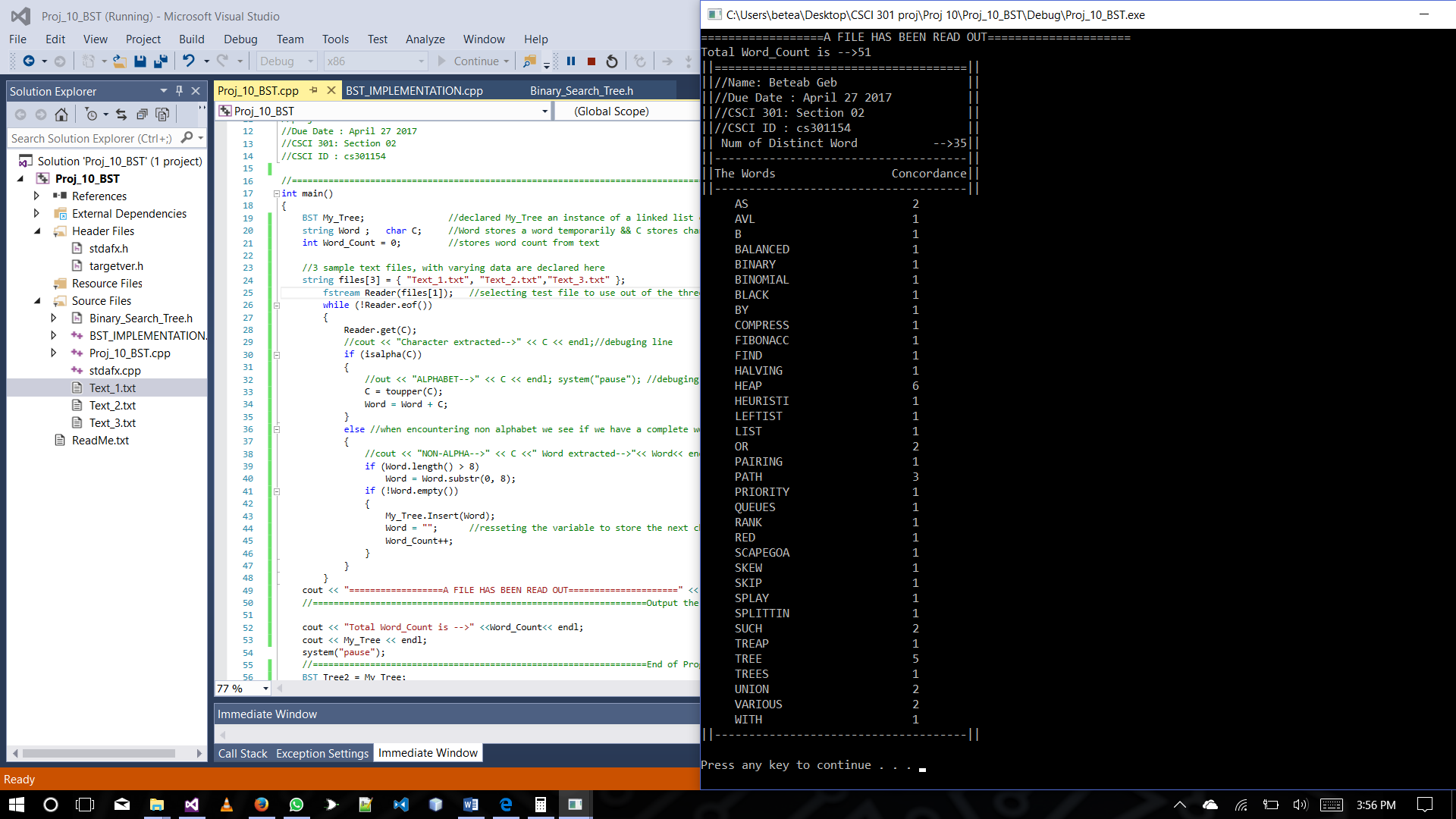
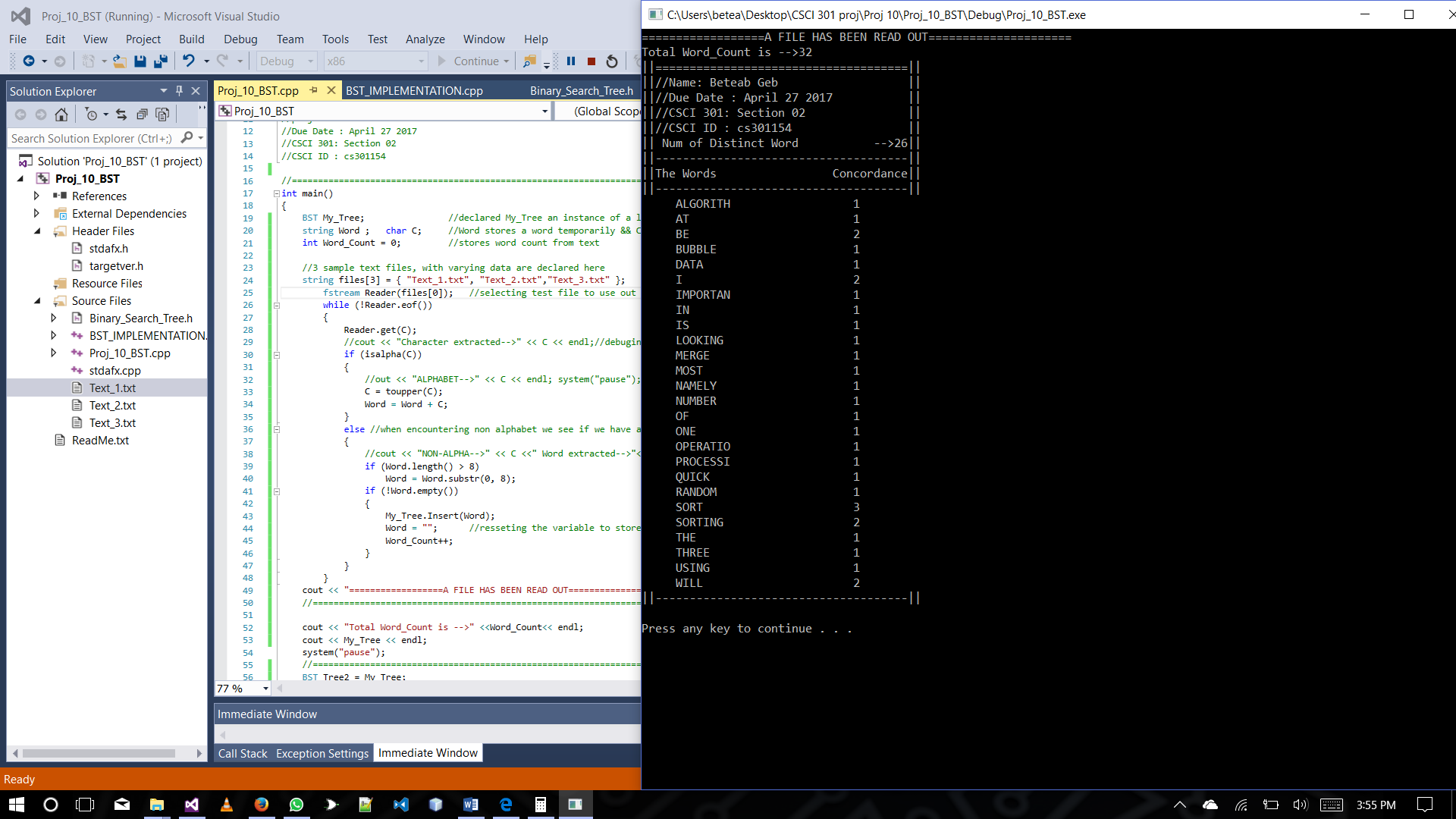
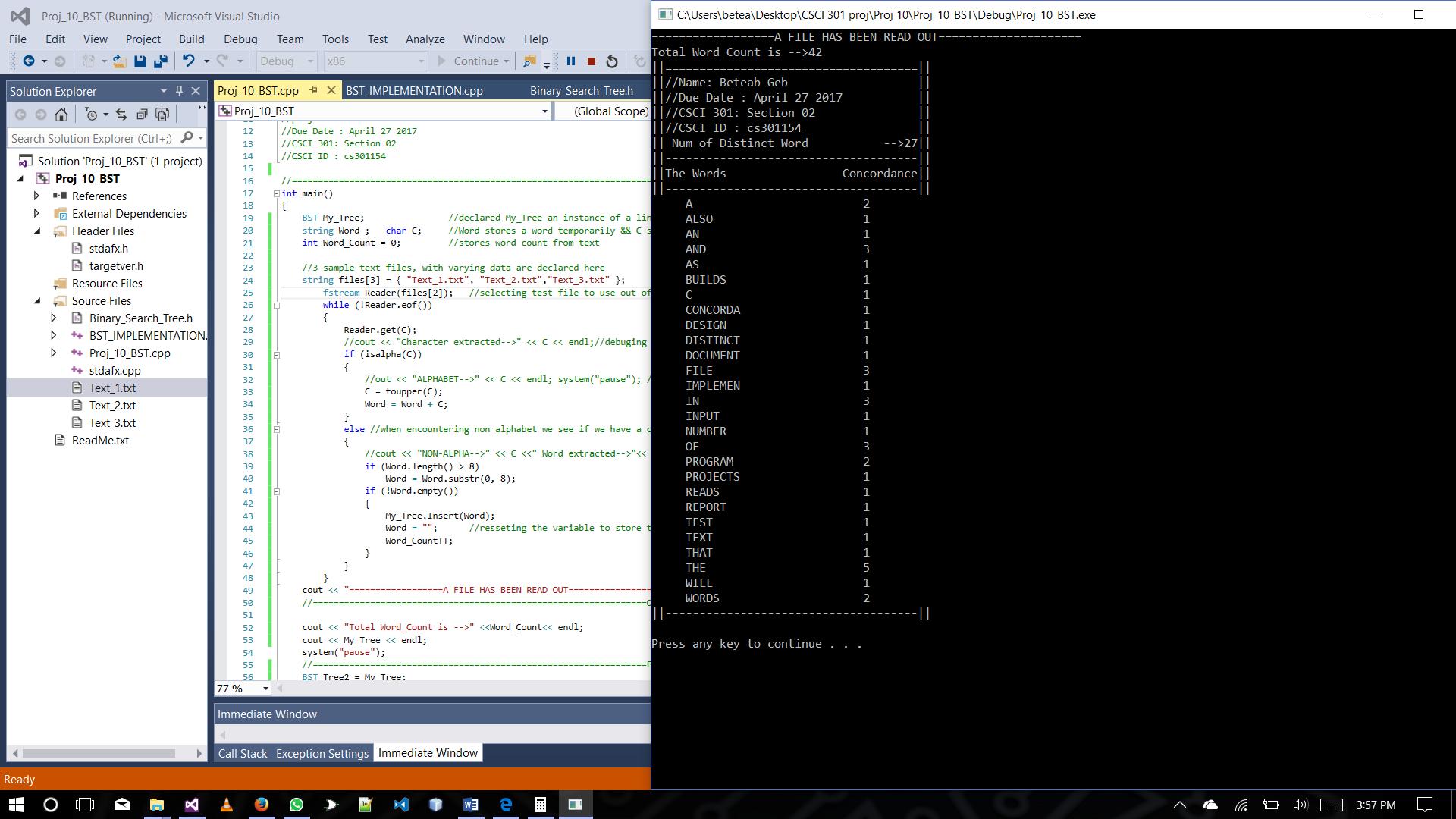
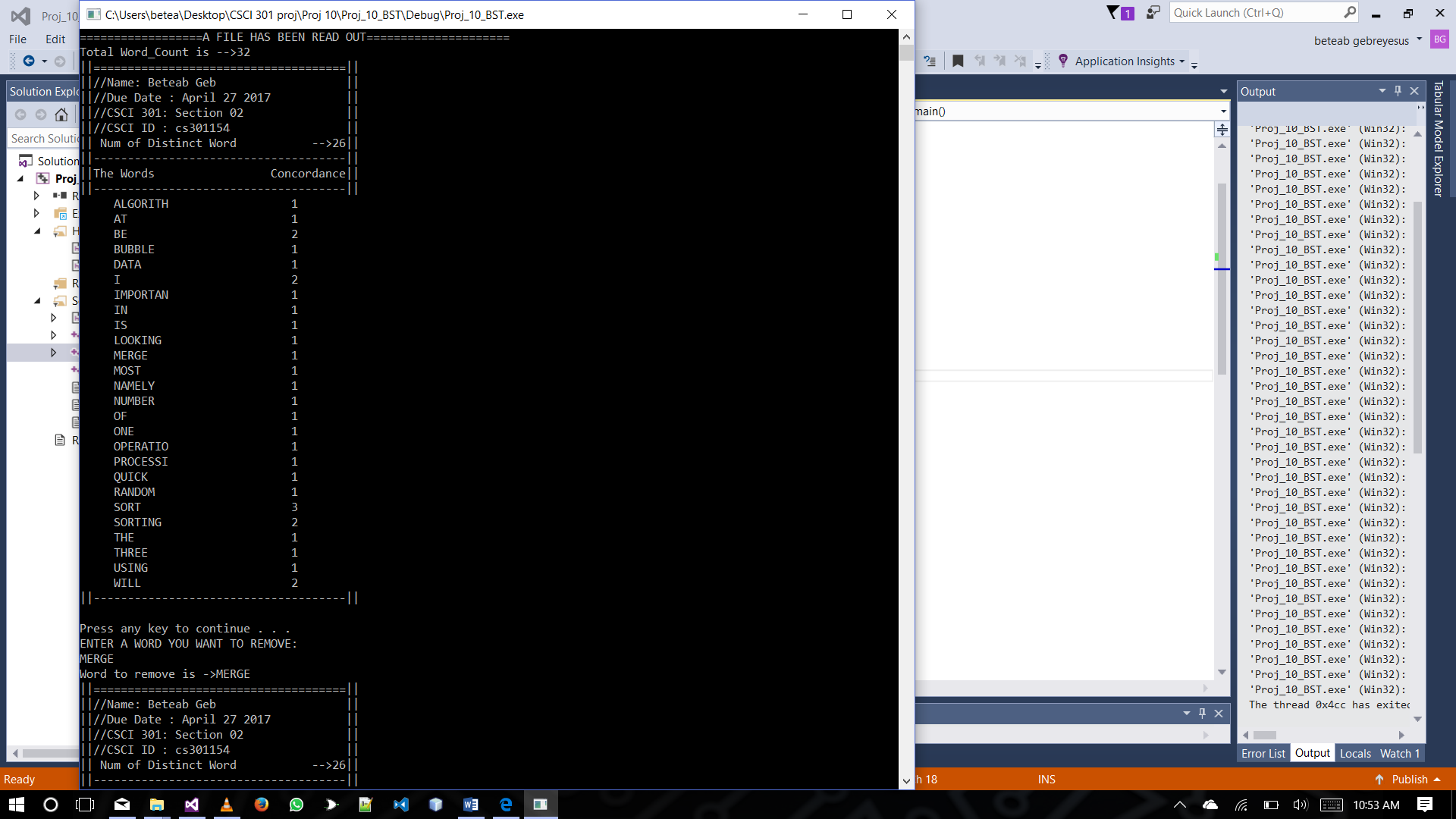
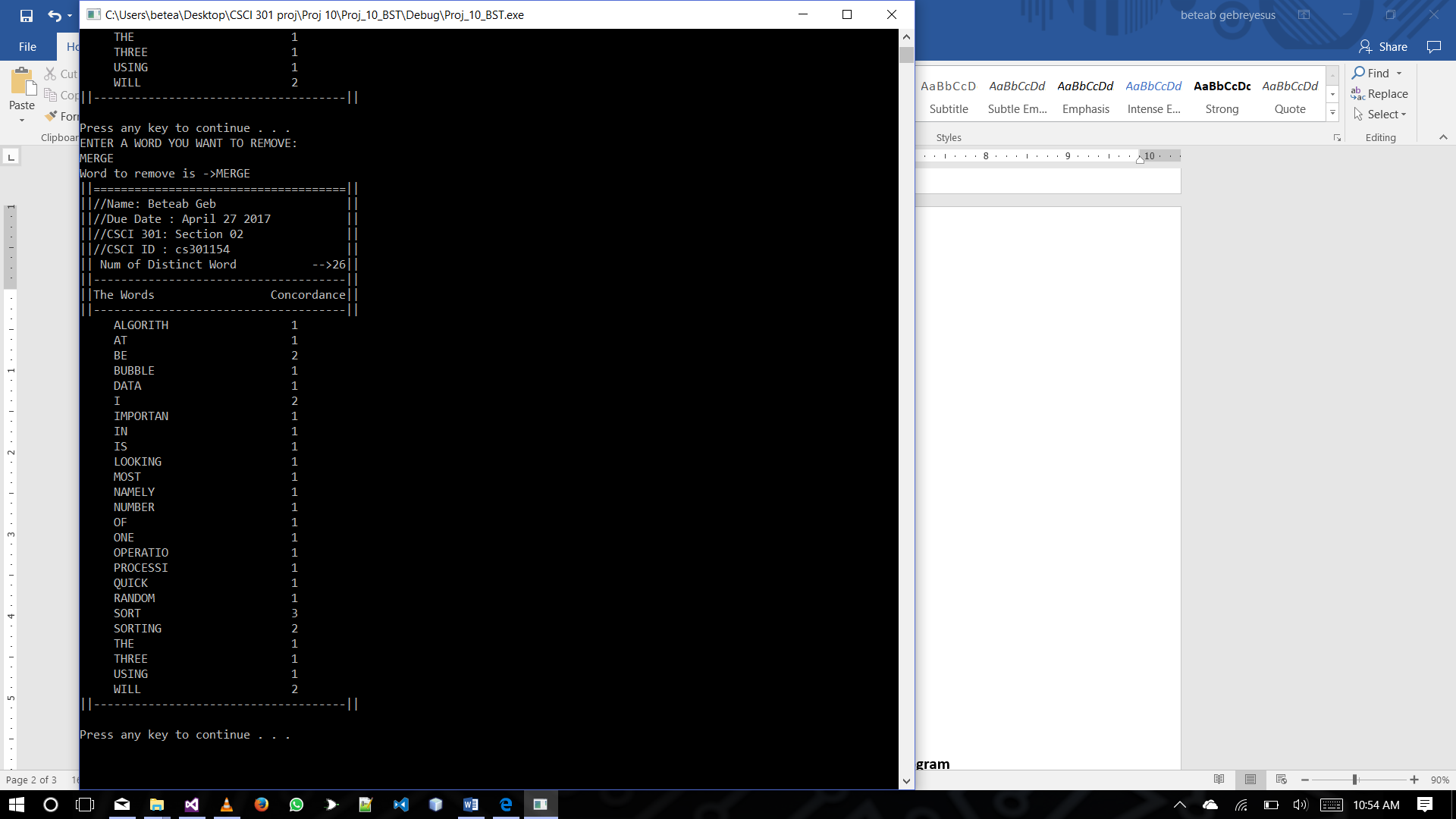
Text\_1.txt Text\_2.txt Text\_3.txt

TESTING REMOVAL **BEFORE AFTER REMOVAL OF** MERGE

**Program Tests and screen shots of the three runs of the program**

|  |  |
| --- | --- |
| Input for testing | Program Output  Concordance Information |
| **“Text\_1.txt”**  Sorting is one of the most important  operations in data processing. I will be  looking at three sorting algorithms,  namely Merge\_Sort, Quick\_Sort, Bubble\_Sort.  I will be using random number | Word\_Count ->32  Distinct Words ->26  Max-Word-Appearance-> 3(SORT) |
| **“Text\_2.txt”**  various balanced trees such as AVL tree, red-black tree,  B-tree, splay tree, treap, skip list, or scapegoat tree,  priority queues such as binary heap, leftist heap,  skew heap, binomial heap, Fibonacci heap,  or pairing heap, union find with various heuristics  (union by rank, path compression, path halving, path splitting). | Word\_Count ->51  Distinct Words ->35  Max-Word-Appearance-> 6(HEAP) |
| **“Text\_3.txt”**  design, implement, test, and document a  C++ program that reads an input file of  text and builds a concordance of the  words in the file, as in Projects 4  and 6. The program will also report the  number of distinct words in the file. | Word\_Count ->42  Distinct Words ->27  Max-Word-Appearance -> 5(THE) |