|  |  |
| --- | --- |
| GROUP MEMBERS | TASKS |
| Dejah Taylor | Loading, Sorting, Displaying, Exception Handling |
| Phillip Code | Sentence Validation for both structures |
| Kristofer Campbell | Time Calculation, Case Validations |
| Tahjae Campbell | GUI, Error-Checking, Sentence Validation, Comments |
|  |  |

Group Report

Data Structures Time Comparison

File – 260 words (All figures are in seconds)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Data Structure | Load/Parse | Sort | Add | Search | Print | Sentence Validation |
| Stack | 0.25 | 0.641 | 0  (illogical after sorting) | 0 | 0.297  (sorted) | varies on response time and input |
| Tree | 0.249 | 0.421 | 0(illogical) | 0.016 | 0.421  (sorted) | varies on response time and input |

Note: Results vary each time on program run

File – 260 words + 520 words (All figures are in seconds)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Data Structure | Load/Parse | Sort | Add | Search | Print | Sentence Validation |
| Stack | 0.532 | 2.016 | 0  (illogical after sorting) | 0 | 0.927  (sorted) | varies on response time and input |
| Tree | 0.516 | 1.281 | 0  (illogical) | 0 | 1.281  (sorted) | varies on response time and input |

Note: Results vary each time on program run

File – 260 words + 520 words + 1040 words (All figures are in seconds)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Data Structure | Load/Parse | Sort | Add | Search | Print | Sentence Validation |
| Stack | 1.062 | 1.97+ | 0 (illogical after sorting) | 0 | 2.251  (not sorted) | varies on response time and input |
| Tree | 1.062 | 3.016 | 0  (illogical) | 0.016 | 3.048  (sorted) | varies on response time and input |

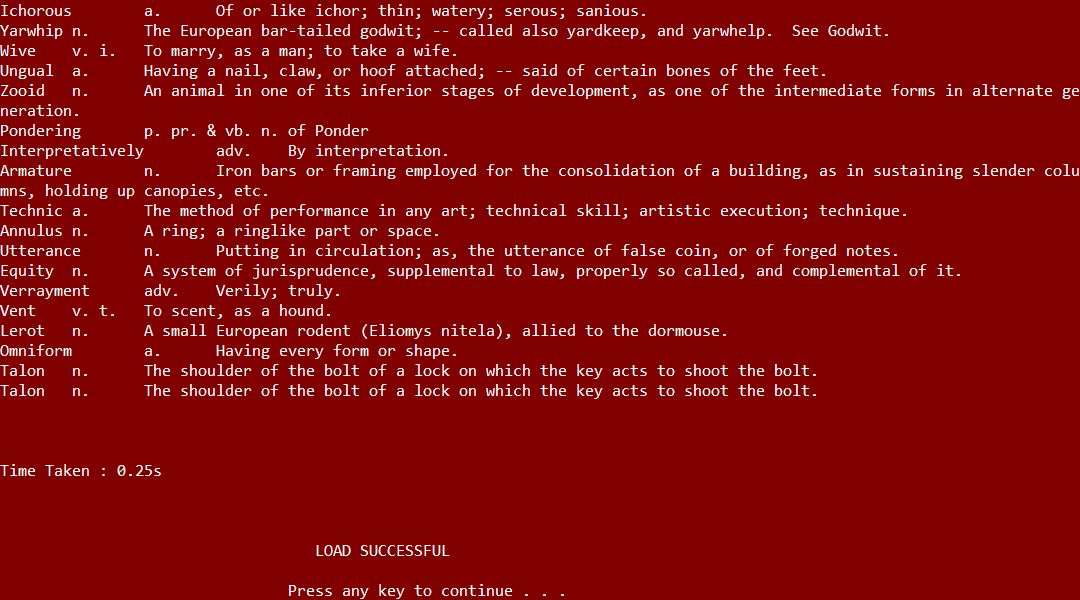
Note: Results vary each time on program run

Data Structure Recommendation

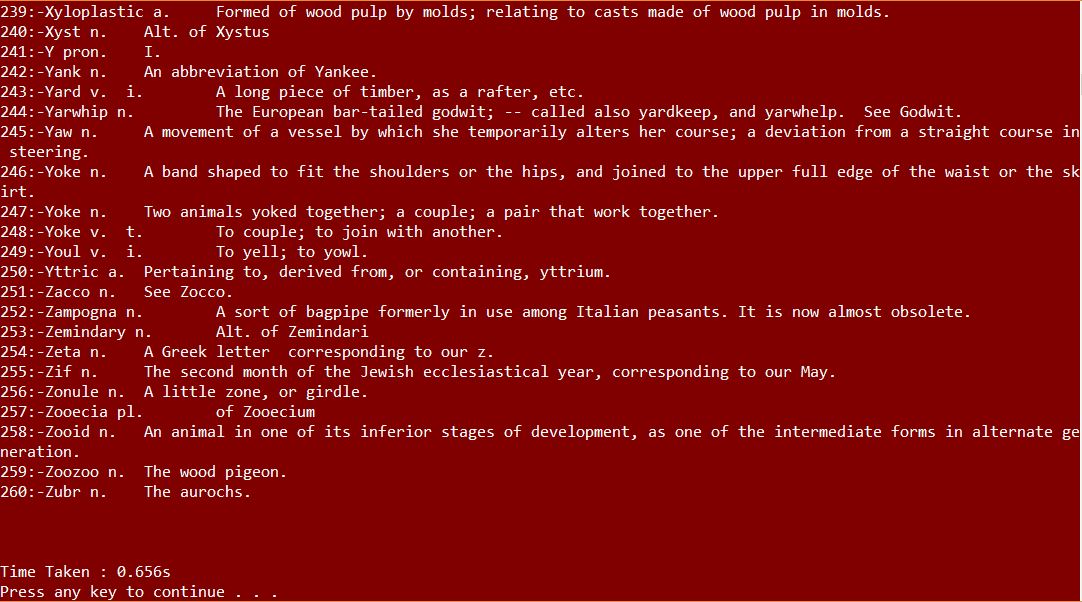
In our project, we used the tree and a stack as our data structures. The tree data structure proved easier to work with. The stack however, had some inconsistences. Sorting the stack was very difficult as it meant we had to pop elements from the stack to an array, run a sorting algorithm on the array and then push all the elements from the array back into the stack. In our project, when we added all the files to the stack, the sorting algorithm crashed. As such a stop watch was used give a close as possible time hence the ‘+’.

We would recommend using the linked list data structure instead of the stack. The linked list structure is a general list and as such features more features which would’ve been more efficient to use for this project.

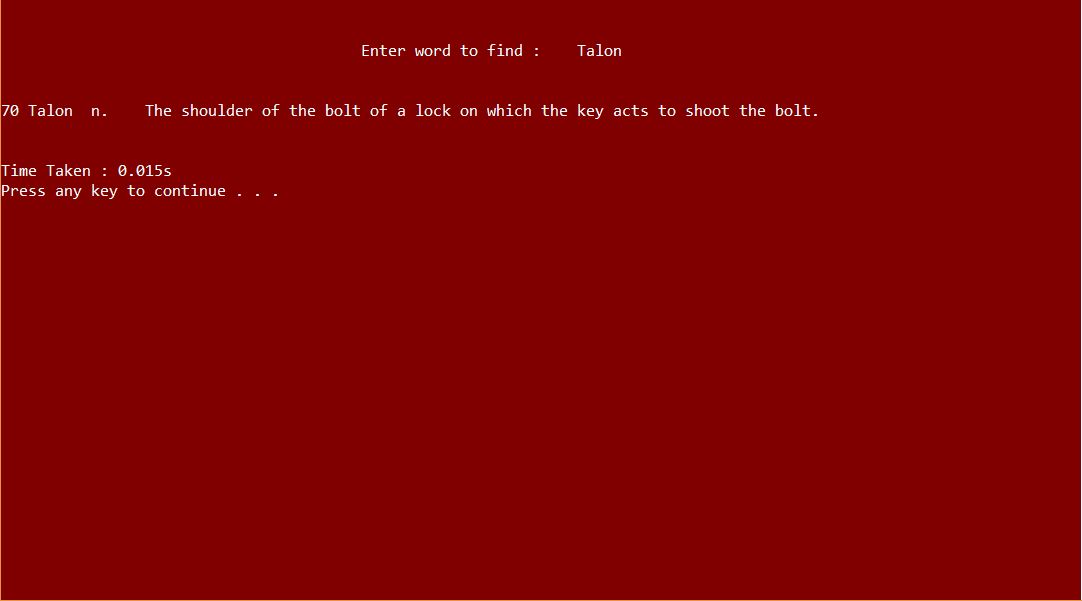
Stack Operation Screen Shots (260 word file)

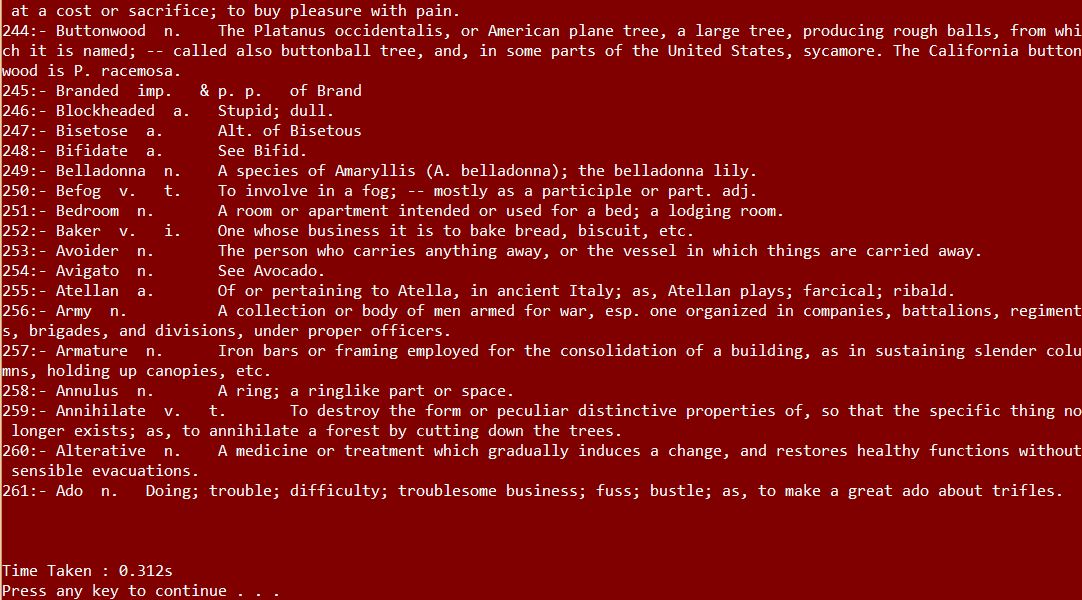
Load

Sort



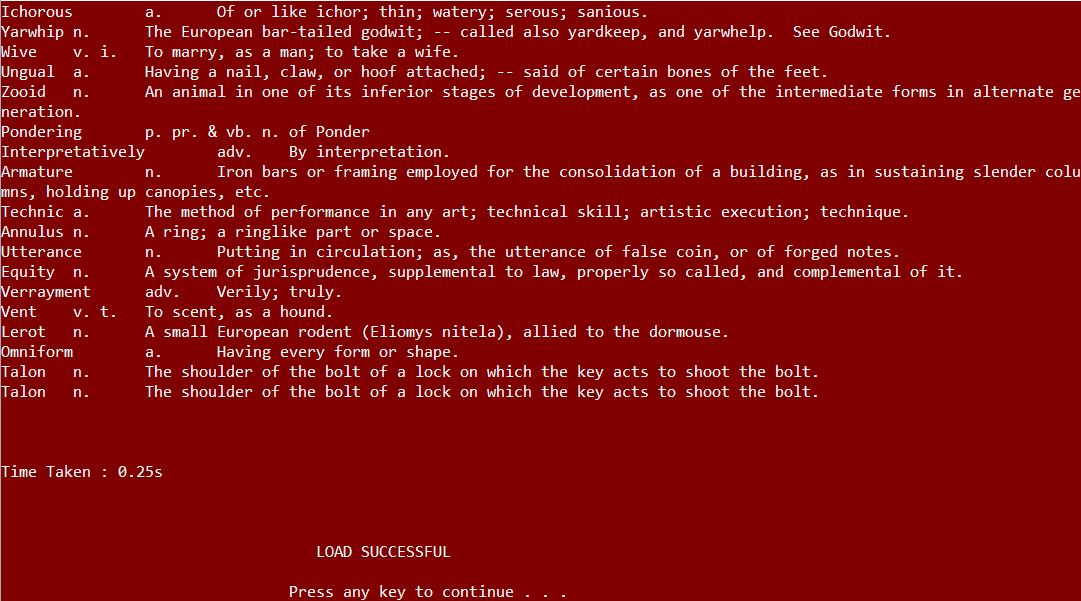
Add Word

 Search for Word

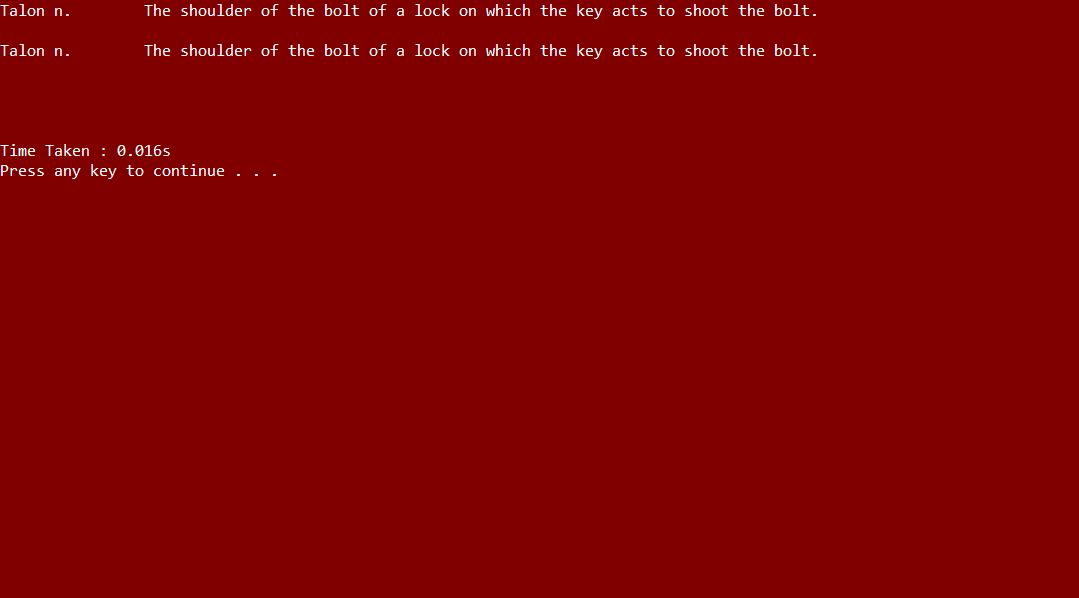


Print Word

Tree Operation Screen Shots (260 word file)

Load

Sort

Search

Add

Print

