Dillon Li – 1642853

Project 1: Tries

Submitted: 10/23/16

Assignment Description

Implement search completion using the trie data structure. The dictionary is provided, and all possible results for a given input must be displayed.

Logic

I implemented several necessary functions: insert, search, and print_leaves.

Insert takes in a string and inserts all the nodes (including intermediate ones) necessary to create that string. If a node already exists, it traverses that node instead of inserting a new one (i.e. inserting ‘aad’ and then ‘ab’, when inserting ‘ab’, node containing ‘a’ will not be inserted twice).

Search takes in a string and returns a pointer to the node that corresponds to that string. If the string is not found, it will return a NULL pointer.

Print_leaves uses the search function to take in the node of the string that a user has input. If that node is a leaf, it will print that node, if not, it will print out all leaf node children beneath it (every word in the dictionary inserted that begins with those letters).

Main() is structured such that if a NULL pointer returned by search (the string is not found), then print_leaves will not run, but a message will be returned to the user instead.

Output Examples

Returns queries:

```
Please type search queries:
aa
Your options are:
aaas
aactive
aadvantage
aaker
aap
aapg
aaron
aarp
aas
aau
Press any key to continue . . .
```

```
Please type search queries:
bran
Your options are:
branch
branches
brand
branded
brandenburg
brandi
branding
brandname
brandnew
brandon
branson
brant
Press any key to continue . . .
```

Leaf node:

```
Please type search queries:
aactive
Your options are:
aactive
Press any key to continue . . .
```

Query not found:

```
Please type search queries:
shivlshailghewilarudslufias
There are no results that match your query
Press any key to continue . . .
```