

Ahmedabad University  
School of Engineering and Applied Science



Data Structures and Algorithms - Lab  
Project Report 1

Project Title: Dictionary Implementation Using TRIE

Group Number:  
04

Group Members:

Ashna Jain (201501008)  
Dhruti Chandarana (201501015)  
Charmi Chokshi (201501021)  
Manasi Dubey (201501051)

## Table of Contents

### 1. Brief Description of Project

- Problem Statement

### 2. List of Data Structure to be used

- TRIE
- Why did we use Trie?

### 3. Operations to be Performed on each Data Structures

- Block Diagram and Brief Description to Explain all Operations on each Data Structures
- Requirements to Processing on Data Structures

### 4. Flowchart/ Pseudo Code of each Logical Process

- Flowchart to Describe Whole process

### 5. Pseudo Code for all Logical Process

### 6. List of Programs

- File Name of each Program
- Source Code of each Program
- Input Data
- Output Generated
- List of Bugs and Defects

### 7. Test Results

### 8. References

- Problem Statement:

Dictionary is the most used application nowadays. So we thought to implement a dictionary with many more features added.

- The main feature is obviously to search for a meaning of a particular word. We are not just showing the meaning of the word entered but also its full definition, example, synonyms and antonyms.
- In many of the dictionaries when we enter a wrong word it shows “no results found” but in our case we have implemented a function which displays many words similar to the entered word.
- We have also added a function which shows some of the words recently searched. It helps the user to memorize the words as sometimes it happens that you forgot the words you searched for before but you want to search the meaning again because you forgot it.
- Another feature added is word of the day. Here we print a new word generated randomly from memory and its description everyday so that user can learn a new word each day and improve their vocabulary.
- Translator is one of more features included. It translates the English word into French or Spanish as per the users’ choice.

- List of Data Structures to be used with design Logic:

TRIE :

Trie stores keys that are usually strings. It is basically a *k*-ary position tree.

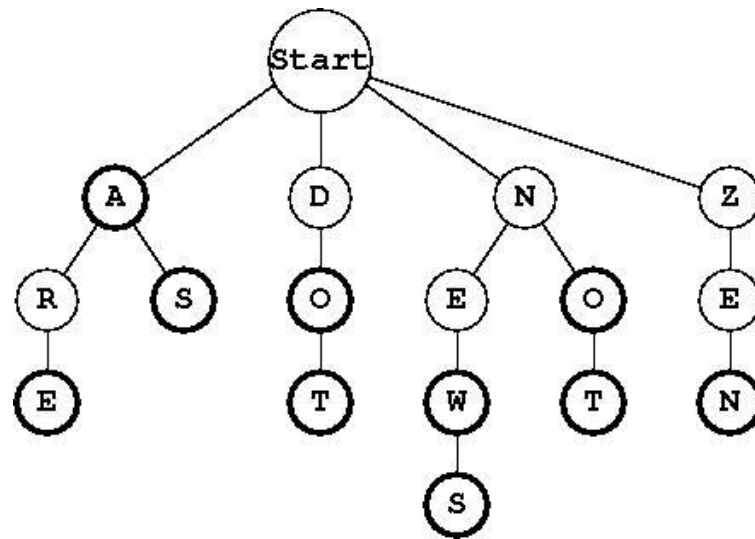
In contrast to binary search trees, nodes in a trie do not store the keys associated with them. Rather, a node's position in the tree represents the key associated with that node. All the descendants of a node have a common prefix of the string associated with that node, and the root is associated with the empty string. Figure below shows a trie.

Why did we use TRIE ?

A Dictionary can also be implemented using other concepts (e.g. HASH TABLES).

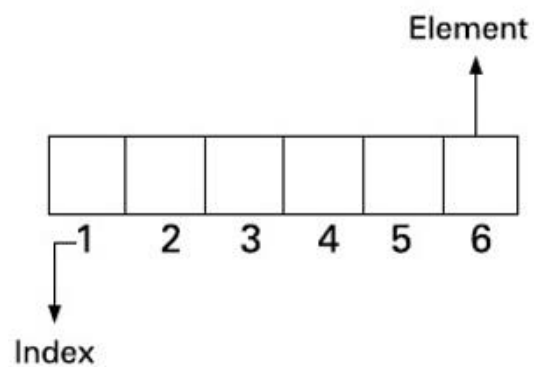
1. Searching for data in a trie is faster in the worst case,  $O(m)$  time, compared to an imperfect hash table, which may have numerous key collisions.
2. Trie is free from collision of keys problem.
3. Unlike a hash table, there is no need to choose a hash function or to change it when more keys are added to a trie.
4. A trie can sort the keys using a predetermined alphabetical ordering
5. TRIES can also be used to find words with common prefix
6. Memory is also used efficiently as words are not stored independently (they are stored as characters and furthermore words with common prefix are not allocated separate memory).

- TRIE

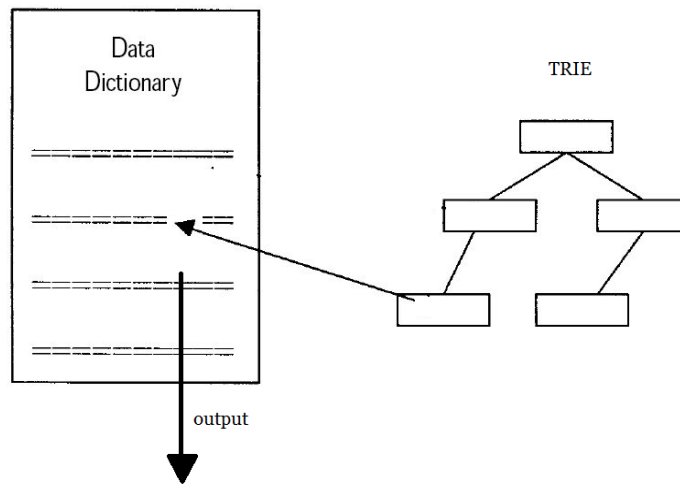


- ARRAY

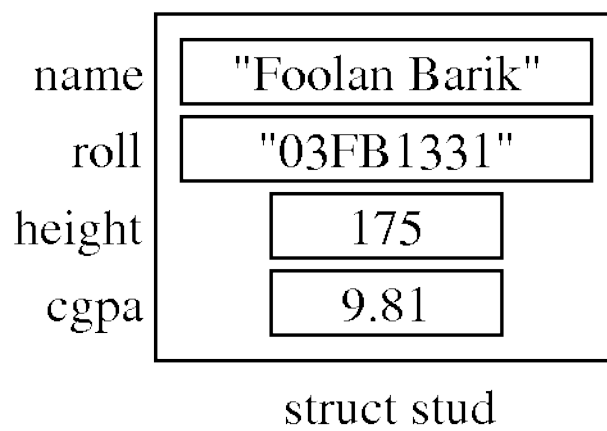
Array



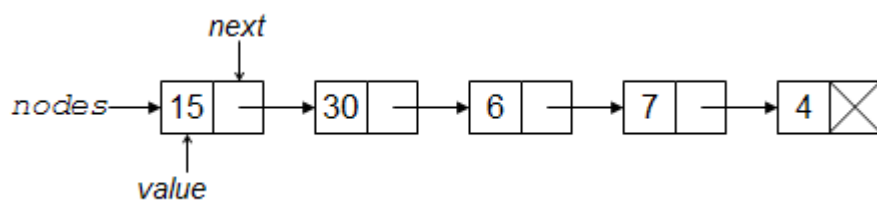
- FILE



- STRUCTURE

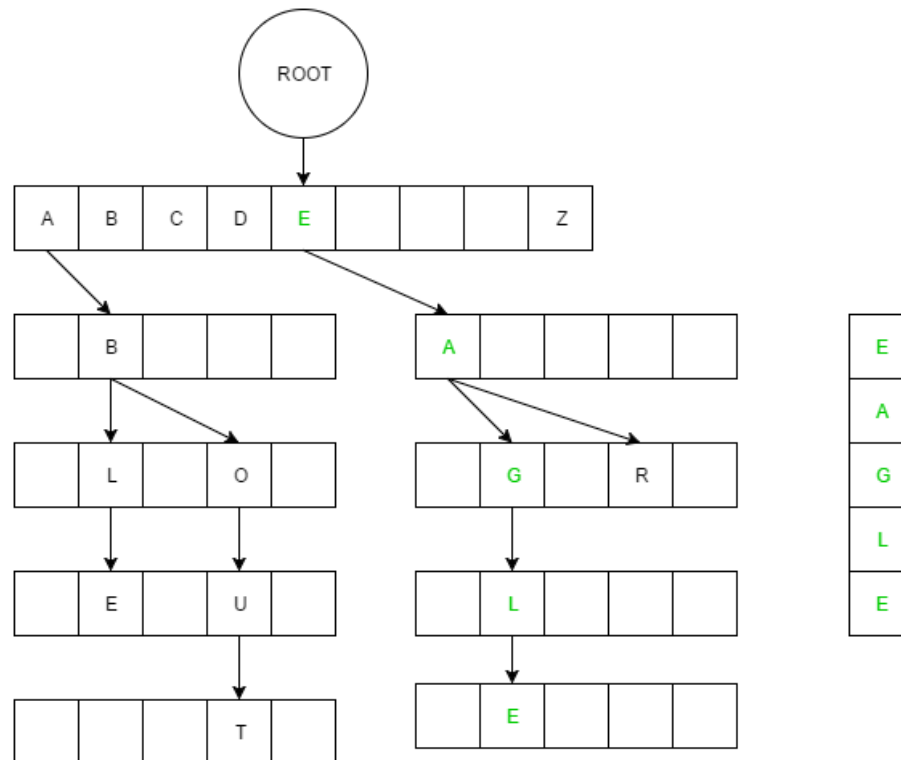


- LINKED LIST



- Operations to be performed on each data structures:

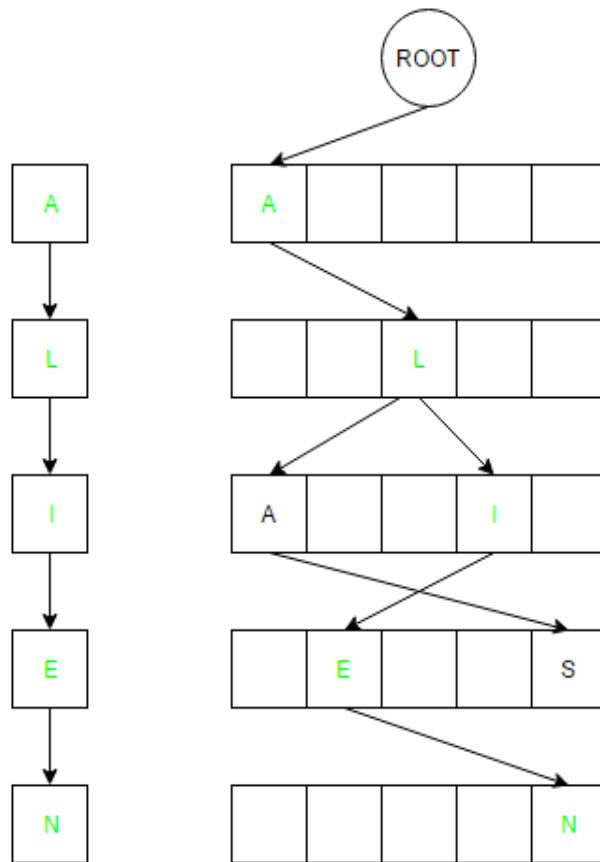
- Insertion Operation in TRIE data structure:



Insertion of "EAGLE" in TRIE data structure  
where other elements already exist

- If user want to enter new word in dictionary then we are using this function
- Insertion of an element is shown in diagram where we are inserting single character of word “EAGLE” at on time in TRIE tree.
- Finding E from root node and then inserting A; A is already there so next character G; now two link from A, one to R and another to G and then similarly inserting all elements.
- Finally we have two words starting from E in TRIE: EAGLE and EAR.

- Searching Operation in TRIE data structure:

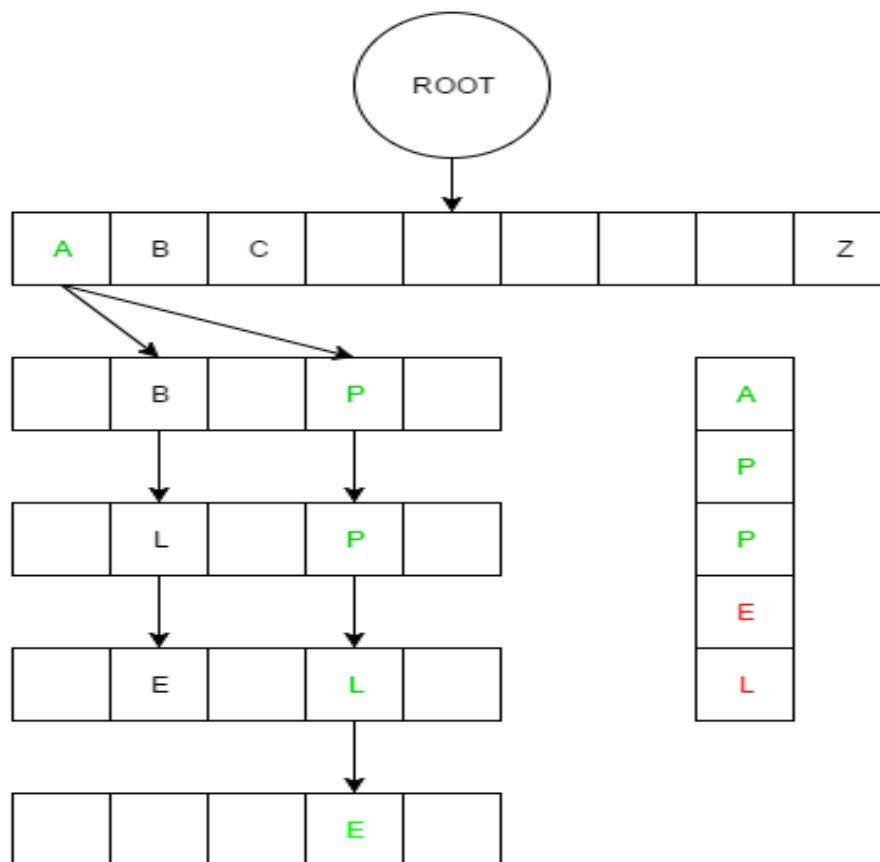


Searching for “ALIEN” in the trie.

- User will provide a word(here “ALIEN”) to be searched in the tree and then its meaning will be displayed.
- In our case first “a” is searched from root in the first array of letters.
- Similarly, “l”, “i”, “e”, “n” are searched from the 2<sup>nd</sup>, 3<sup>rd</sup>, 4<sup>th</sup>, and 5<sup>th</sup> arrays respectively.
- While searching for the word alien, “alas” was also formed but as the respective letters of alas were not similar to the individual letters of alien, so it’s meaning will not be displayed.



- Finding Similar Words from TRIE data structure:



Given word is "APPEL" which is not right So  
function will return "APPLE"

- If any word is not in dictionary/ TRIE we are printing similar words to the entered word.
- Comparing characters of given word in TRIE; if there is no such word, then finding possible valid combination of the given word.
- Here given word is “APPEL”, so in TRIE “APP” is correct combination that is much similar to the given word and then printing all words starting from prefix “APP”; here it is “APPLE” and similarly finding other valid prefix and printing other similar words.

- Requirement to process in data structure:

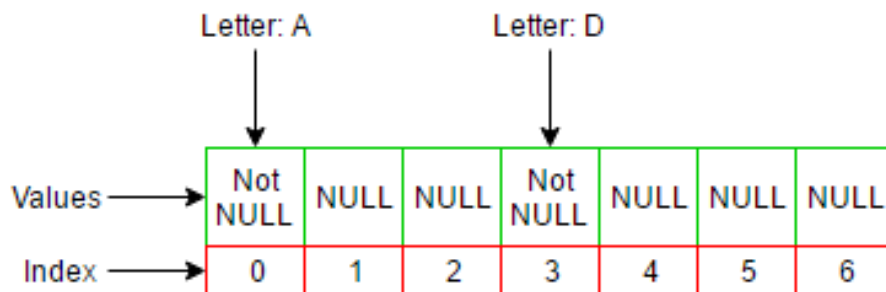
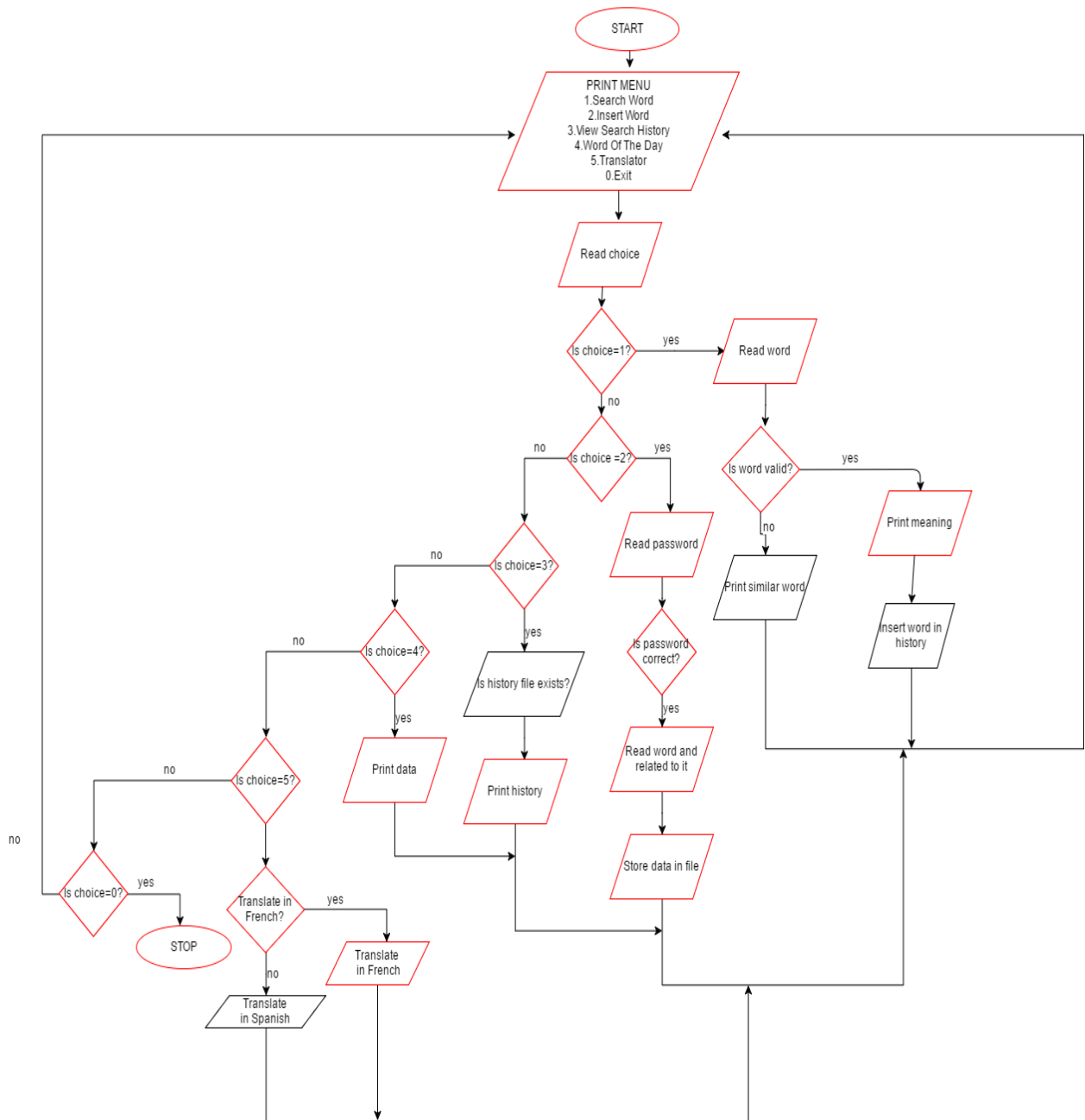


Diagram shows how array elements virtually sorted; Not actual data but some details that represent it.

- It is require that data stored in array which is part of structure of TRIE data structure us sorted.
- Initially values at all indexes are NULL.
- For every alphabet there is a key value for it and that key is its index in array.
- Here key is assign as given below.
  - For letter: key is ASCII value(letter) – ASCII value('A').
  - For letter: 'A' key= 0
  - For letter: 'B' key= 1 and so on.
- This key is index for array that represent letter in array; so for 'A', 0<sup>th</sup> index will be not NULL, for B 1<sup>st</sup> index and so on.

- Algorithms / Pseudo code / Flowcharts for each logical process:

- Flowchart:



- Pseudo Code of each Function:

#### 1.getnode()

This function will generate a New Node and returns the address of that node.

1.Set temp:=NULL(temp is a variable of node type)

2.Allocate memory to temp.

3.If ( temp is not NULL) then:

Set fname[temp]:=NULL,i:=0.

Repeat the following step while i<26.

Set the alphabets field at i to NULL.

Set i:=i+1.

End of loop

End of if statement.

4.return temp.

#### 2.insert(root, key)

This function takes root and key as input and inserts a key in the TRIE data structure using getnode() function.

1.Set temp := length of the string key

Set temp := root,level:=0.

2. Repeat the following steps while temp<length:

Find the index of character at key[level](index is found by subtracting Ascii value of the character by 65) and store it in index.

If(!temp->alphabets[index]) then:

Set temp->alphabets[index] :=Call getnode().

End of if statement.

Set temp:=temp->alphabets[index].

Set level:=level+1.

End of loop.

3.Set fname[temp]:= key.

3.search(root, key)

This function searches for a particular key in the TRIE data structure.If the word is found then it returns true otherwise false.

1.Set length := length of the string key,temp:=root,level:=0.

2.Repeat while level<26 :

Find the index of character at key[level](index is found by subtracting Ascii value of the character by 65) and store it in index.

If(!temp->alphabets[index]) then:

return false.

End of if statement.

Set temp to temp->alphabets[index].

End of loop.

3.return (temp!=NULL and fname[temp]:=NULL).

4. addHistory(key)

This function adds a word which are frequently used and adds them in the history file.

1.Set fp: = NULL(fp is a file pointer).

2.Open the “history.txt” file and allocate fp to it.

3.if(fp=NULL) then:

Write : An error occurred.

Else:

Write the string in the file.

End of if structure.

4.Close the file.

4.void search\_similar( root,key)

This function takes root and key as input and searches for words which are similar to key if words found then print them else prints no similar words found.

1.Store the length of the key in variable length.

2.Declare i,j,flag and len as integers.

3.Settemp:=root,temp1:=root,free:=root,flag:=0,cnt:=0,level:=0,i:=0.

4.Declare character c.

5. If(length=1) then:

Write: No similar words found then Return.

6. Repeat while level<length-2:

Find the index of character at key[level](index is found by subtracting Ascii value of the character by 65) and store it in index.

If(temp->alphabets[index]) then:

Set temp to temp->alphabets[index].

Else:

Set flag:=1.

End of if structure.

End of loop.

7. Find the index of character at key[level](index is found by subtracting ASCII value of the character by 65) and store it in index.

    If(temp->alphabets[index]) then:

        Set temp to temp->alphabets[index].

    Else:

        Set flag:=1.

End of if structure.

8.If(flag!=1) then:

    Repeat while i<26:

        If(temp->alphabets[i]) then:

            Set len:=0.

        Repeat while len<length-2:

            Print key[len].

        End of loop.

        Set c := i+(char a).

        Print the character which has the Ascii value c.

            Set free to temp->alphabets[i].

Repeat while(free and cnt=0):

    Set j:=0.

    Repeat while j<26:

        If(free->alphabets[j])

            Print the character with Ascii value

        j+(char a).

        If(free->alphabets[j])

            Set free to free->alphabets[j].

    Else if(!free->alphabets[j])

        Set cnt:=1.

End of if structure.

End of loop.

End of if structure.

Set cnt:=0.

9. If(flag=1) then:

Write:SIMILAR WORDS NOT FOUND.

5. printData(key)

This function prints the data contained in the file key.

1.fptr := NULL(fptr is a file pointer).

2.Set ext,:=".txt".

3.Concat key and ext.

4. If(pointer of the file name is not NULL, open file in read mode) then:

Write>Error opening file.

Else:

Store the first character of the file in ch.

Repeat the steps while (ch != EOF):

Print ch.

Store the next character in ch.

End of loop

End of if structure.

5.Close the file.



## 6. insert\_word(key)

This function inserts a new word in the dictionary by storing the description in the file and adding the word in the word file.

1. Set fptr:=NULL,fp:=NULL and declare a structure record.
2. Open the "words.txt" file and allocate pointer fp to it.
3. If(fp =NULL) then:

Write: An error occurred.

Else:

Write the word in the file "words.txt".

End of if structure.

4. Close the file.

5. Set ext:=".txt" and record.word:=key.

6. Concat key and ext.

7. Open the key file and allocate pointer fptr to it and open the "translation.dat" file and allocate pointer file to it.

8. If(fptr=NULL) then:

Write: An error occurred.

Else:

ask the user to enter the meaning, description,  
synonyms, antonyms and example of word and  
store them to their relative character arrays.

End of if structure.

9. store the above contents of each character arrays in file of that word

Write the record in "translation.dat" file.

10. Close the file

## 7.wordOfTheDay(root)

This function takes root as input and prints the word of the day data which is a random generated by system each day.

- 1.Declare time\_t t.
- 2.Do time(&t).
- 3.Set fptr := NULL,fp:=NULL,i:=0.
- 4.Set char com[15],c[15],cc[12],con[15],ch[15].
- 5.Set flag:=0,i.
6. .Open “t\_date.txt” file and allocate pointer fptr to it.
- 7.Open “t\_date.txt” file and allocate pointer fp to it.
8. If(fptr=NULL) then:  
Write:ERROR OPENING FILE.  
Return.  
Else:  
Get the first 12 characters from file pointed by pointer ptr and store it in ch.  
End of if structure.
- 9.Repeat while i<11:  
Set cc[i]:=ctime(&t)[i].  
End of loop.
- 10.Set i:=0.
- 11.Repeat while i<11:  
If(ch[i]!=cc[i]) then:  
Set flag:=1.  
End of if structure.

End of loop.

12.If(flag=0) then:

Repeat while(!feof(fp)):

    Read the string in the file.

    End of loop.

13.Write:The word of the day is:

14 Print ch.

15.Call printData(c).

8.printHistory()

    This function prints the contents of history file.

    1.Setfptr := NULL,fn:= "history",ext:= ".txt".

    2.Open the file fn and assign it to fptr.

    3. Concat fn and ext.

        4. If(fptr!=NULL) then:

        Write: Error opening file.

        Else:

            Store the first character of file fn in ch.

        Repeat while (ch != End of file) :

            Print ch.

        Store the next character in ch.

        End of loop

    End of if structure.

    5.Close the file.

## 9.wordRead(fileName, choice)

This function gives the French or the Spanish translation of the word based on user's choice.

1.Set fptr:= NULL.

2.Set cnt:=0.

3.Declare a character word.

4.Declare a instance of structure of meaning and store it in record.

5.Open the file named filename in read binary mode and allote it to fptr.

6.If(fptr=NULL) then:

Write: File doesn't exist.

Else:

Write: Enter a word.

Store the word entered by the user in the word field.

Read the first record from the file named filename and store it in record field.

Repeat while it is not end of file:

If record.word and word are equal then:

Print the given word.

If(choice = 1) then:

Print record.french.

Else:

Print record.spanish.

End of if structure.

Set cnt:=cnt+1.

Read the next record and store it in record field.

End of loop.

7. If(cnt=0) then:

Write:Unable to find translation.

10.main()

This is the main function of the program.

1.Declare fptr(file pointer),fp(file pointer),fr(file pointer).

2.Declare char ch,newWord,wordSearch,a,wor,cc,ch.

3.Declare choice,password,j,r.

4.Set i:=0,f:=0,cnt:=0,checkPassword:=12345.;

5. Set root:=getnode().

6. If(pointer of file is NULL(open file in read mode)) then:

Write:ERROR OPENING FILE.

Else:

Repeat while(It is not End of the file)

store each successive character in c

Call insert(root,c).

End of loop.

7.Close the file.

8.If(pointer of file is NULL(open file t\_date in write mode))

then:

Write:ERROR OPENING FILE.

Else:

1.Declare time\_t t;

2.Do time(&t).

3. Open t\_date in read mode and allocate pointer fp to it.
4. Repeat while i<11:
  - Set cc[i]:=ctime(&t)[i].
  - End of loop.
5. Store the first 12 characters of file "t\_date.txt" in ch.
6. Close the file.
7. Set i:=0.
8. Repeat while i<11:
  - If(cc[i]!=ch[i]) then:
    - Set f:=1.
    - End of if structure.
  - End of loop.
9. If(f=1) then:
  1. open file t\_date in write mode
    2. Write the date and time in file "t\_date.txt".
    3. Set r:= (rand() %50 )+1. (generating random word using inbuilt function)
  4. If(pointer of file word.txt is null (open file in read mode)) then:
    - Write:Error opening file.
    - Else:
      - Repeat while(scan and store successive character in c till end of file and i<r)
      - Set i:=i+1.
      - End of loop.

5.Store the word after the ith position present in the file “words.txt” in variable c.

6.Close the file.

9.Write: WELCOME TO THE 'MACD' DICTIONARY.

10.Repeat the following steps while ch!=0:

1.Write: MENU.

Write:Enter your choice:

2.Read the value of choice.

3.If choice=1 then:

Set cnt := 0.

Write:ENTER THE WORD YOU WANT TO SEARCH:

Store the word in wordSearch.

If((pointer of history file is null (open it in read mode))  
then:

Write: Error opening file.

Else:

Repeat while(scan successive character of file till end of file  
and store them in character c):

If(compare wordSearch and c, if they are equal) then:

Write:word found directly from history file.

Call printData(wordSearch).

Set cnt:=cnt+1;

Close the file.

If(cnt = 0) then:

If(search(root, wordSearch))

then:

Write:WORD FOUND FROM TRIE.

Call addHistory(wordSearch).

Call printData(wordSearch).

Else:

Write :WORD NOT FOUND.

Call search\_similar(root,wordSearch).

break;

Else if choice = 2 then:

Write:Enter Password to add New Word into  
Dictionary:

Store the password in the variable  
password.

If(password = checkPassword(previously set password))  
then:

Write: Enter a new word.

Store the new word in the variable  
newWord.

Call insert(root, newWord).

Call insert\_word(newWord).

Else:

Write : Invalid word.

Break



Else if choice = 3 then:

Write: Enter a word.

Store the new word in a.

Call search\_similar(root,a).

Break.

Else if choice = 4 then:

Write:SEARCH HISTORY.

Call printHistory()

Break.

Else if choice = 5 then:

Call wordOfTheDay(root).

Break.

If choice=6 then:

Write:Choice Menu:

Write:1. French Transaltion

Write:2.. Spanish Translation.

Write: Enter your choice.

Store the choice in variable key.

Call wordRead("translation.dat", key).

Break.

Else if choice = 0 then:

Break.

Else:

Write: invalid choice.

Break.

- List of programs:

- Description of all files:
- getNodeFunction.h = header file that contains getNodeFunction.
- historyFunctions.h = header file that includes all function related to view history.
- insertFunctions.h = header file that contains function to make TRIE and to insert an element in dictionary.
- searchFunction.h = header file that contains function to search an element in dictionary.
- searchSimilarFunction.h = header file that contains function that search similar words of the given word.
- transFunction.h = header file that contains function to find translation of the given word.
- WODFuction.h = header file that contains function to find 'word of the day'.
- transNode.h = header file that contains structure which is used to store record of translation of word.

- trieNode.h = header file that contains structure which is used to make TRIE.
- Main.c = source file that includes main function.

- [Source Code](#)

The above link is provided to view all the functions of our program. The folder is shared with dsa15.seas@gmail.com.

#### Inputs of each Function:

- Search Meaning : A Word (character array) whose meaning User wants to Find
- Add New Word in the Dictionary: A New Word (character array) to add in the existing dictionary as well as its meaning, description, synonyms and antonyms as character array.
- Translate Word : A Word (character array) whose translation user wants to find

#### Output of each Feature:

- Search Meaning: If entered Word is a Valid Word: The meaning, description, synonyms, antonyms and example of searched word. Data will be fetched from File and will be

printed as character array.

If the entered word is Not a Valid Word: Words present in the Dictionary with Similar kind of Spelling will be printed as output.

- Add New Word in the Dictionary: A Success Statement if the Word is successfully added in the dictionary.
- Translate Word: The Translated Word (character array) either in French or in Spanish will be printed.
- View Search History: The Data of history file will be printed.
- View Word of the Day: Print the auto generated word with its whole description

Code analysis:

Time Complexity:-

$O(K)$  is the time complexity of TRIE data-structure in its worst case scenario, where  $K$  is the size of the string. Worst case scenario is when the input word has no common characters.

Advantages:

- All in One!!
  - We have made an application which not only a simple

dictionary which tells you the meaning of the entered word but also it tells you the whole description, synonyms, antonyms and practical use of that word in a sentence.

- Even this is not the only feature but we created another application of Word Translation in this only which gives us the translation of entered word in either French or in Spanish

- It provides Word of the Day which means every day our code will generate a random word from its memory and will give all the data related to it. One can learn at least a new word in a day and can increase his/her vocabulary using our application.

- Our application can correct a wrong word as well. So in spite of displaying “Word not Found” like in case of many other dictionaries ,ours gives you Suggestions

- Currently we have just 50-55 words in our dictionary but at any point of time one can Add a new Words in the Dictionary by first entering Password for the Safety.

- We are also providing List of Frequently Searched Words. So to decrease the time complexity if the searched word is one of the Frequently searched word then our code will directly print the data of that file and if not than first traverse the TRIE than print the data.

Limitations, Drawbacks and Bugs:

- Add New Word: While we do provide the functionality to add new words into the dictionary, though there is an inherent flaw in it - the person might enter any arbitrary sequence of characters (which the 'Trie ' identifies as a word).
  - For example - the person might enter 'asdfg' as a word into the file. Our program cannot distinguish between an arbitrary sequence of characters and a valid English word.
  - This basically implies that the software can be easily corrupted.
- Deletion of a word: It can only be done by explicitly opening the text file containing the data and then deleting it.
- Memory Consumption: Due to our lack of knowledge and time as well we have not implemented Word Scraping. So currently we are storing each data in Text File which consumes more memory but if we had used word scraping then our code might not be using any amount of memory.
- Less Words : Though at any point of time one can Add new Words in our Dictionary but currently we have just 50-55 words in our dictionary
- Search Incorrectly Spelled Word: We do have implemented an algorithm in which similar words to the incorrect word entered are shown to the user, but have not implemented anything which corrects the words which are spelled incorrectly.

- Test Results:
- Searching for a word in dictionary:

```
***WELCOME TO THE 'MACD' DICTIONARY***

~*~*~MENU~*~*~

1. Search Meaning
2. Add new Word in the Dictionary
3. View Search History
4. Word of the day
5. Translation of word
0. Exit

Enter your Choice: 1

      ENTER THE WORD YOU WANT TO SEARCH: apple

MEANING:
a round fruit with red, yellow, or green skin and firm white flesh
This ia a Noun.

FULL DEFINITION:
1: the fleshy usually rounded red, yellow, or green edible pome fruit of a
   usually cultivated tree (genus Malus) of the rose family;
   also : an apple tree ù compare crab apple.
2: a fruit (as a star apple) or other vegetative growth (as an oak apple)
   suggestive of an apple.

IDIOM:
apple of one's eye
one that is highly cherished <his daughter is the apple of his eye>
```

- Trying to add new word in the dictionary with wrong password:

```
~*~*~MENU~*~*~

1. Search Meaning
2. Add new Word in the Dictionary
3. View Search History
4. Word of the day
5. Translation of word
0. Exit

Enter your Choice: 2

      Enter Password to add New Word into Dictionary: 123

INVALID PASSWORD!!
```

- Entering new word in the dictionary and then searching for it

```
Enter your Choice: 2

    Enter Password to add New Word into Dictionary: 12345

Enter a new Word:hello

Enter Meaning of the Word: it is a greeting

Enter which Part of Speech it belongs to: ---

Enter Full Definition of the Word: ---

Enter Example of the Word: hello, how are you?

Enter Synonyms of the Word: hii, wass up

Enter Antonyms of the Word: ---

Enter French Translation of Word: Bonjour

Enter Spanish Translation of Word: hola

~~~MENU~~~

1. Search Meaning
2. Add new Word in the Dictionary
3. View Search History
4. Word of the day
5. Translation of word
0. Exit

Enter your Choice: 1

    ENTER THE WORD YOU WANT TO SEARCH: hello

MEANING:
it is a greeting
This is ---.

FULL DEFINITION:
---

EXAMPLE:
hello, how are you?

SYNONYMS:
hii, wass up
```



- Viewing word of the Day

```
~~~MENU~~~
1. Search Meaning
2. Add new Word in the Dictionary
3. View Search History
4. Word of the day
5. Translation of word
0. Exit

Enter your Choice: 4

THE WORD OF THE DAY IS:dairy

MEANING:
a company that sells milk and foods made from milk (such as butter and cheese)
this is a noun

FULL DEFINITION:
1) a room, building, or establishment where milk is kept and butter or cheese is made

2) a : the department of farming or of a farm that is concerned with the production of milk, butter, and cheese
   b : a farm devoted to such production

3) an establishment for the sale or distribution chiefly of milk and milk products

EXAMPLE:
Tom's family has been involved in dairy farming for years.
```

- Finding French translation of apple

```
~~~MENU~~~
1. Search Meaning
2. Add new Word in the Dictionary
3. View Search History
4. Word of the day
5. Translation of word
0. Exit

Enter your Choice: 5

Choice Menu:
1. French Translation
2. Spanish Translation
Enter Your choice: 1

Enter Word: apple

Given word: apple
French Meaning: pomme
```

- Finding Spanish of apple

```
~~~~MENU~~~~  
1. Search Meaning  
2. Add new Word in the Dictionary  
3. View Search History  
4. Word of the day  
5. Translation of word  
0. Exit  
  
Enter your Choice: 5  
  
Choice Menu:  
1. French Translation  
2. Spanish Translation  
Enter Your choice: 2  
  
Enter Word: apple  
  
Given word: apple  
Spanish Meaning: manzana  
~~~~MENU~~~~
```

- Searching for a invalid translation

```
~~~~MENU~~~~  
1. Search Meaning  
2. Add new Word in the Dictionary  
3. View Search History  
4. Word of the day  
5. Translation of word  
0. Exit  
  
Enter your Choice: 5  
  
Choice Menu:  
1. French Translation  
2. Spanish Translation  
Enter Your choice: 1  
  
Enter Word: aaa  
  
Enable to find translation!
```

- Printing similar words when searching for a miss-spelled word

```
~~~~MENU~~~~  
  
1. Search Meaning  
2. Add new Word in the Dictionary  
3. View Search History  
4. Word of the day  
5. Translation of word  
0. Exit  
  
Enter your Choice: 1  
  
    ENTER THE WORD YOU WANT TO SEARCH: coo  
  
    ***WORD NOT FOUND***  
  
    SOME SUGGESTIONS FOR SIMILAR WORDS ARE:  
comb  
cow
```

```
~~~~MENU~~~~  
  
1. Search Meaning  
2. Add new Word in the Dictionary  
3. View Search History  
4. Word of the day  
5. Translation of word  
0. Exit  
  
Enter your Choice: 1  
  
    ENTER THE WORD YOU WANT TO SEARCH: appl  
  
    ***WORD NOT FOUND***  
  
    SOME SUGGESTIONS FOR SIMILAR WORDS ARE:  
apple
```

- Viewing History

```
~~~~MENU~~~~  
1. Search Meaning  
2. Add new Word in the Dictionary  
3. View Search History  
4. Word of the day  
5. Translation of word  
0. Exit  
  
Enter your Choice: 3  
  
SEARCH HISTORY  
  
apple  
hello
```

- Exiting from application

```
~~~~MENU~~~~  
1. Search Meaning  
2. Add new Word in the Dictionary  
3. View Search History  
4. Word of the day  
5. Translation of word  
0. Exit  
  
Enter your Choice: 0  
  
***THANK YOU FOR USING 'MACD' DICTIONARY***
```

## ● References:

- Merriam Webster
  - <https://www.merriam-webster.com/>
- Word net
  - <https://wordnet.princeton.edu/>
- Stack overflow
  - <https://www.merriam-webster.com/>
- Quora
  - <https://www.quora.com/What-are-the-best-ideas-for-Data-Structure-project>
- Wikipedia
  - <http://en.wikipedia.org/wiki/Trie>
- GeeksForGeeks
  - <http://www.geeksforgeeks.org/trie-insert-and-search/>
- Tutorialspoint
  - <https://www.tutorialspoint.com/>

## BOOKS:

- Data Structure By Seymour Lipschutz
- Data Structure Using C By Reema Thareja

