**Algorithm**

Define a node class Trie

Declare pointer variable children[26]

Declare Boolean end\_of\_word

In node constructor

For i=0 to 26:-

children[26]->NULL

end\_of\_word=false

Define node class trie

Declare pointer variable root

In trie constructor

Root pointer=new node

In Trie class

Insert\_word function(words)

Current pointer=root

For i=0 to i=length of word

If words[i]>=97 & words[i]<=122

Define index= words[i]-‘a’

If current->children[index]=NULL

Current->children[index]=new node

If i=length of word – 1

Current->children[index]->end\_of\_word=true

Current pointer to current->children[index]

Else if words[i]>=65 & words[i]<=90

words[i]=words[i]+32

Define index=words[i]-‘a’

If current->children[index]=NULL

Current->children index=new node

If i= length of word -1

Current->children[index]=true

Current=current->children[index]

Define bool search function(string key)

Node current pointer = root

For i=0 to i<length of key

Define index = key[i]-‘a’

If current->children[index]=null

Return false

If i< length of key -1 & current->children[index]- >end\_of\_word=false

Return false

Current=current->children[index]

Return true

declare function exploreSuggestions(node, prefix, suggestions)

If(node->end\_of\_word)

add prefix to suggestions

For each child node

If(node->children[i]!=NULL

recursively call exploreSuggestions with the latest updated prefix

Declare vector suggestCorrections(wrongString)

Initialize current node=root

Initialize suggestions and prefix as empty vector

For each character in wrongString

find out ASCII value of characters

If(current->children[index])==NULL

break loop

append character to prefix

move to next child node

If(current->end\_of\_word)

add prefix to suggestions

return suggestions

Declare function load\_dictionary(filename)while(file >> word)

push word in file

return dictionary

In main function

load dictionary into a trie

open input file

read each word from the input file

If(trie.search(word))

append it to coloured\_file and wrong\_file

else

append it to wrong\_word and coloured\_file with RED color

Correct paragraph using corrected\_word and using suggestions

print corrected paragraph