Hong Kong Diploma of Secondary EducationExamination

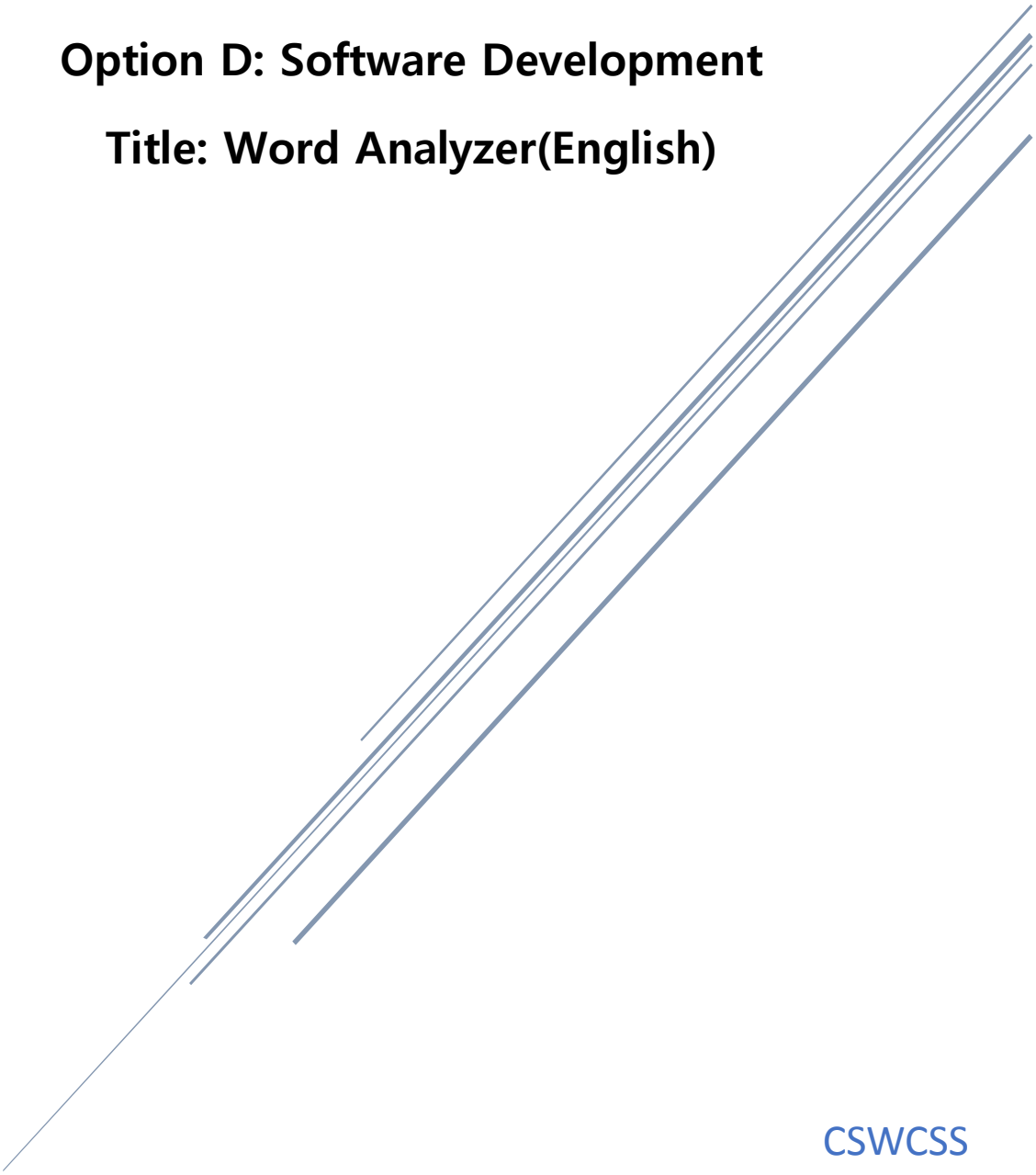
201x

Information and Communication Technology

(Coursework)

Option D: Software Development

Title: Word Analyzer(English)



CSWCSS

Index

Phase 1: Preliminary Investigation

Background

English is an important language for Hong Kong students. In HKDSE, over 20% school uses English as primary teaching language in all subjects except Chinese language and Chinese History. Yet, English language are usually weighted heavier than other subjects in the calculation of university admission score. These facts show us that students have to work hard in English.

However, how can we improve our English? There were plenty of ways that we could improve such as gramma, use of words and sentence structure. In my point of view, the most efficient way is to read more in our daily life. By reading more, we could learn more from others work in order to improve ourselves. Also , students need tools to help checking their own writing before handing it in. Therefore , the main goal of my study is to help students study others work and correct mistakes for their writing.

Necessity

When I first told the idea of my program to some of my classmates, they think it suit their need because in the existing application (PC/IOS/Android) has only a few similar to mine .It integrated the dictionary and word analyze application , so it is unique .

Moreover, for me, as a students , this program is beneficial to my study , with the help of this program , I can get useful words from online newspaper such as SCMP . Also , I can check my writing for my homework and documents (SLP , OEA for my JUPAS application)

Hence, it is worthwhile to develop this program.

Investigation

As The program is mainly design for students, I have interview some classmate before developing the software, here are some requirements/expectation and suggestion from them:

-Checking spelling mistakes

-Individual Vocab Book

-Extract useful words from the text

Objectives

In this project , I am going to develop a word analyze program which can analyze work piece and save individual vocab books for each users .

Phase 2: System Design

Data Output

- Maximum number of users
- Maximum number of words in file
- Number of words in dictionary
- Analyze report
- List of wrong word
- List of correct word

This Chapter will show the designing steps of the program

Function

The system support for following functions:

For **Users**:

Login and registration system	In the interface , user can 1.login 2.login as visitor 3.register This ensure only permitted students and teachers can access the software by registered username and password.
Read file	Allow user to upload text file for analysis.
Generate Analyze report (Report will be displayed on screen)	Generate report including : -Frequency of all words -Total number of words -Total number of paragraphs -Number of vocab
Check wrong word	Check out wrong word from the essay by comparing with the word bank

	Correct word will be written to CorrectWord.txt while wrong word will be written to wrongword.txt
Edit users' own vocab book	Display / Edit user's own
Exit	Execute the program

For **Administrator**:

Edit Dictionary	Add word into word bank

User-friendly functions

-File which need to be read are not dedicated

-Key press is used as a way to replace entering number (To convince user)

- Considered that there are more than one user or the program maybe installed in public computer (maybe a computer in class) ,The program is designed as for more than one user in different time .Every user has their own username and password to protect their privacy.

Menu Design

1.login

-1.1Essay Analyze

-1.2Check VocabBook

-1.2.1 Essay Analyze

-1.2.2 Exit

2.Register

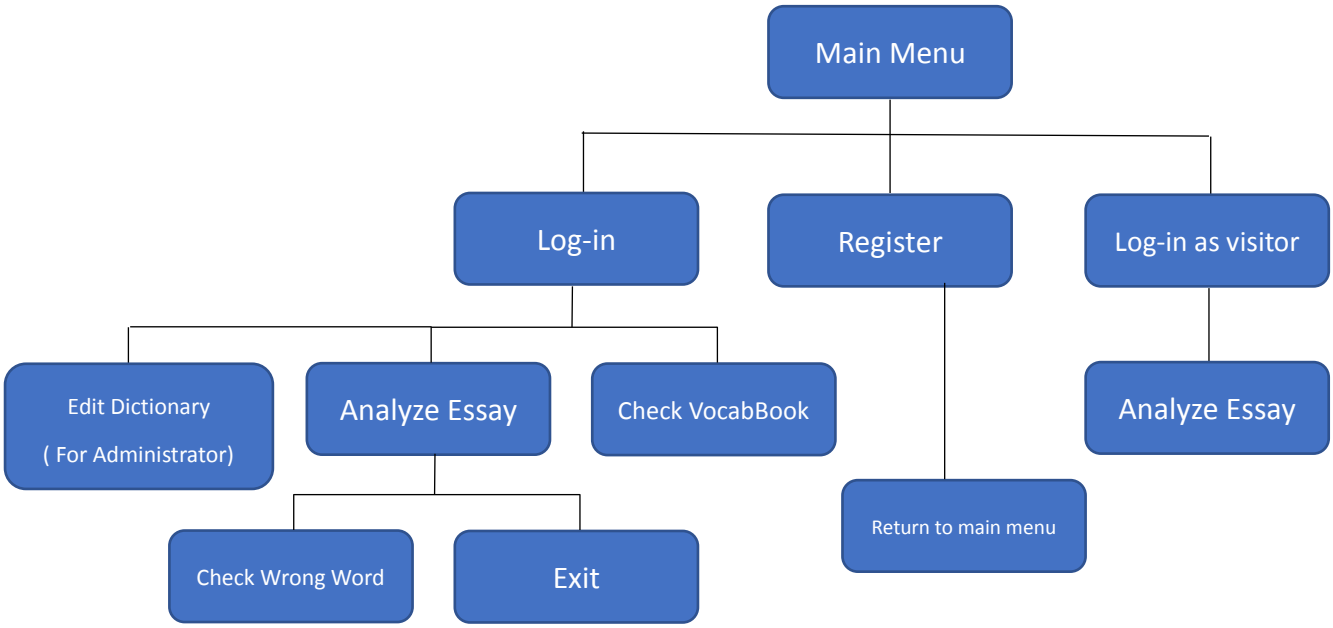
Enter username and password

Confirm information

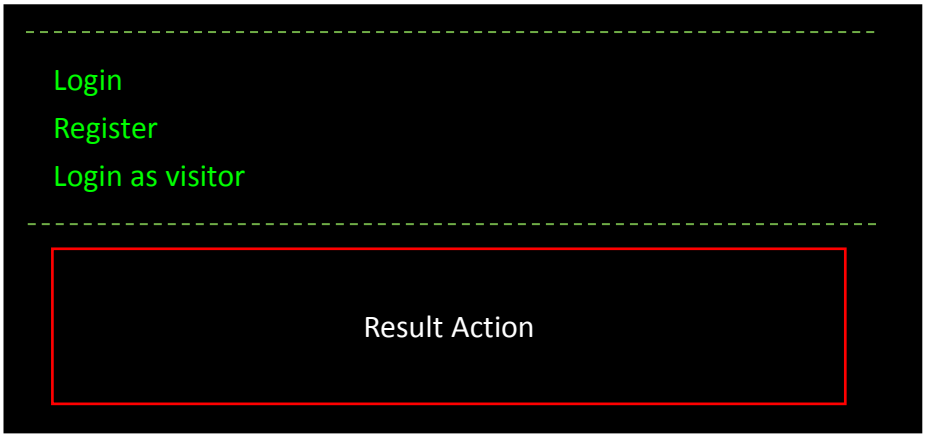
-2.1

3.Login as visitor

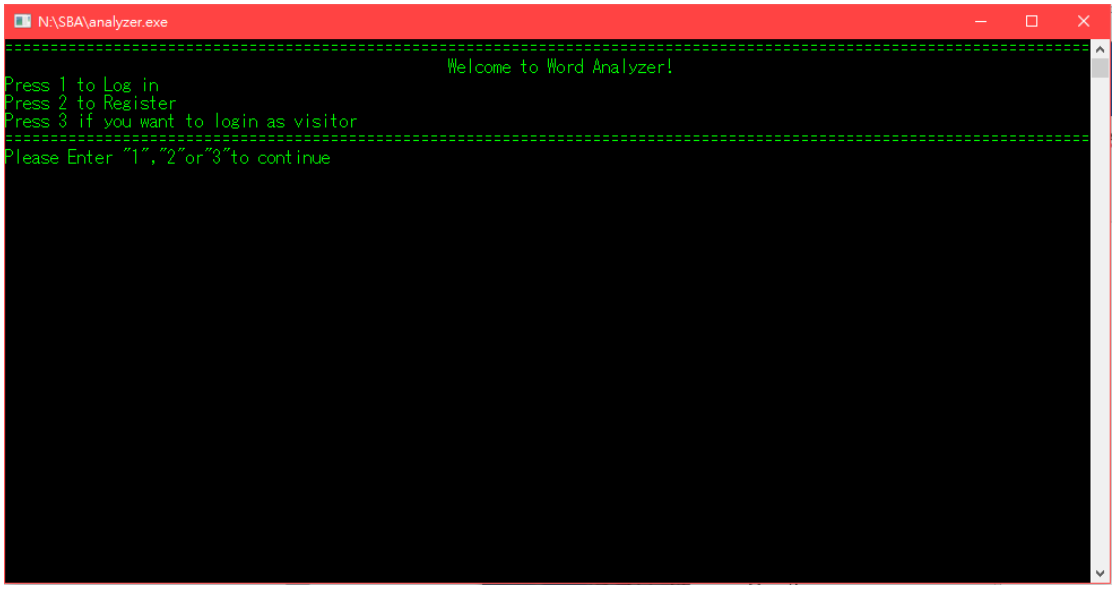
Menu Structure



UI Design



Menu UI

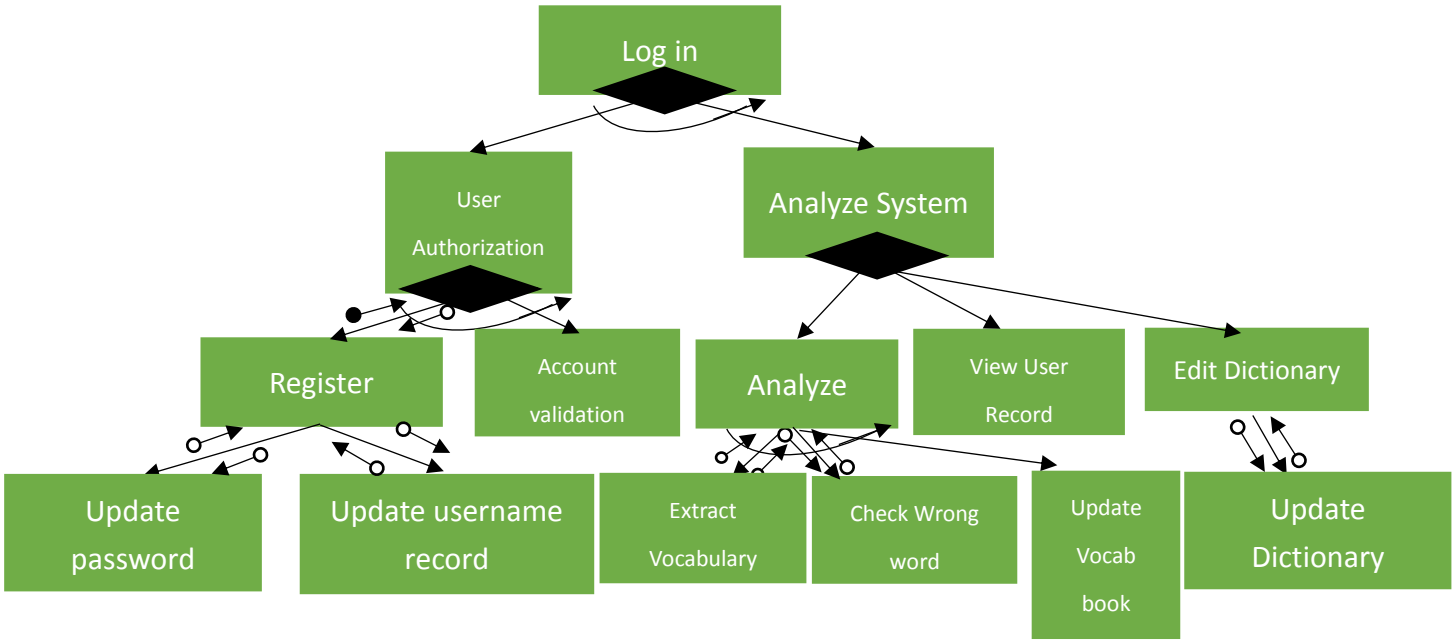


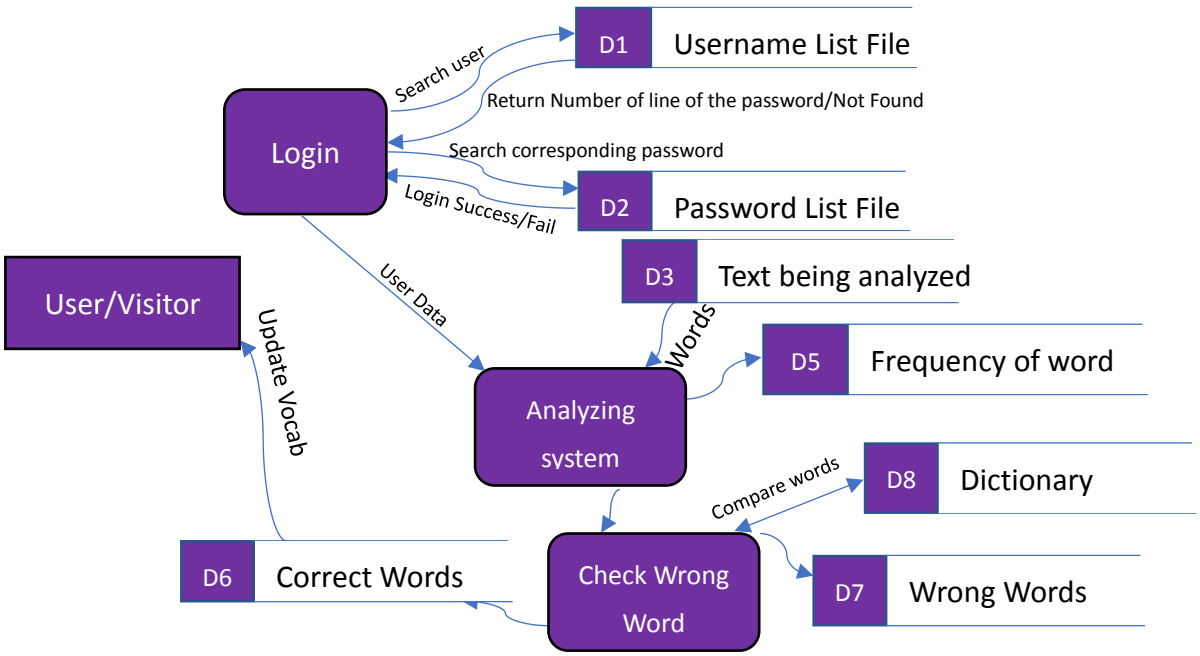
System Flowchart

Data File stored on disk in the graph	Brief description	Actual text files
	The data file stores the list of registered username	/SBA/UserName.txt
	The data file stores the list of registered user's password	/SBA/Password.txt
	The data file stores the text file that user want to analyze , the file name depends on user.	Under /SBA ,Depends on user
	The data file stores the log-ined user's vocabulary	/SBA/username+Vocabbook.txt
	The data file temporarily stores words in the text (sorted)	/SBA/Vocabulary.txt
	The data file stores the list of suspected wrong words from the text	/SBA/WrongWord.txt
	The data file stores the list of suspected wrong words from the text	/SBA/CorrctWord.txt

	The data file stores the list correct word copied from online dictionary . Administrator can edit the file in the software	/SBA/Dictionary.txt

Structure Chart

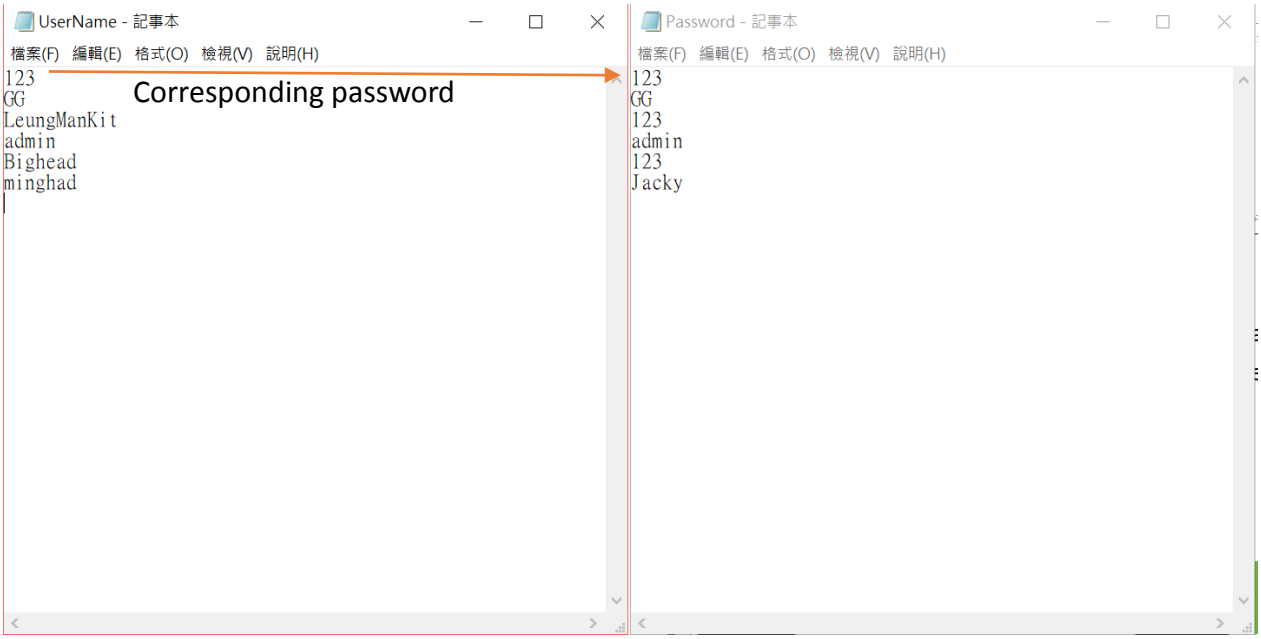




Data Flow Diagram

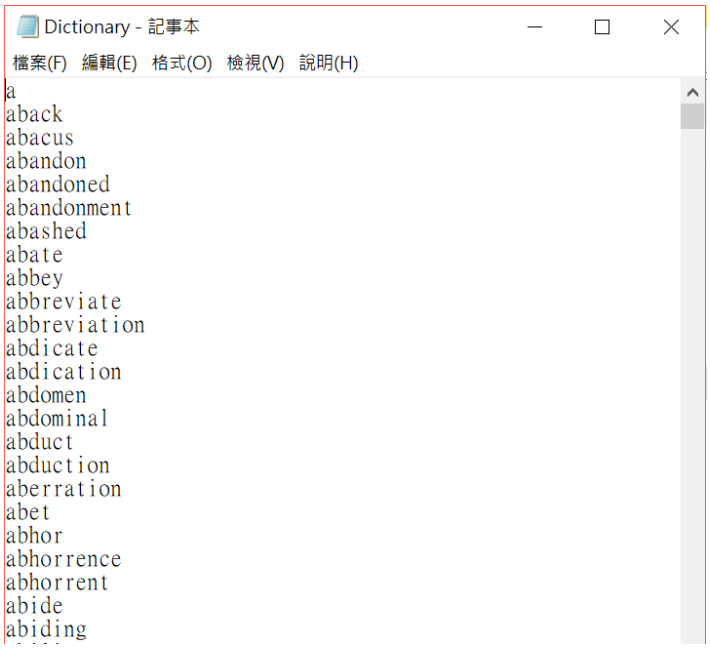
Input Data Format

In order to reduce risk of read/write error, data of the program will be separately stored in to different text files. Also, the data will be store individually on each line .In doing so , information will be well-organised



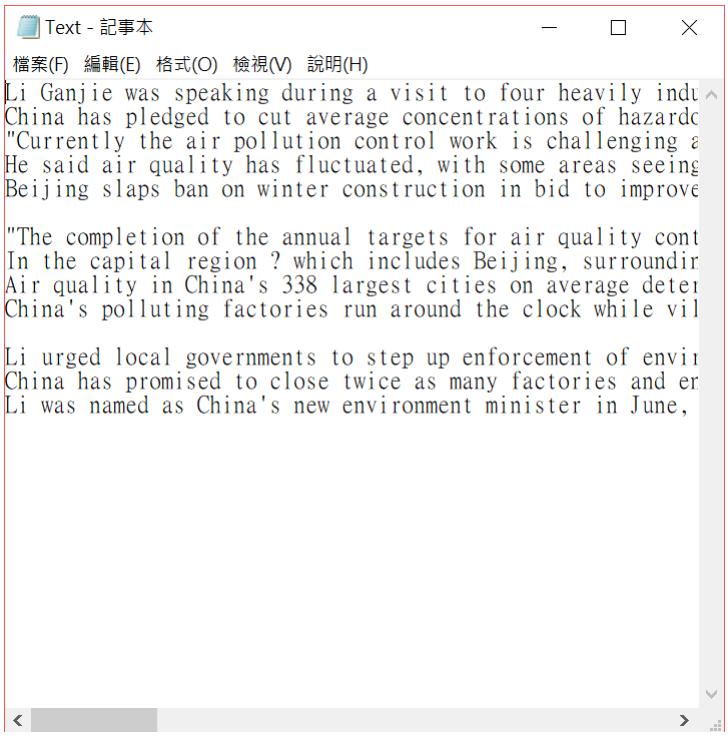
In the screenshot above ,we could see that username and password are saved in two text files. When log in operation is ran , the program will first search the user name in ‘username.txt’ , if the username was found successfully , program will mark the

position of the username in find the corresponding password in the same row in ‘password.txt’



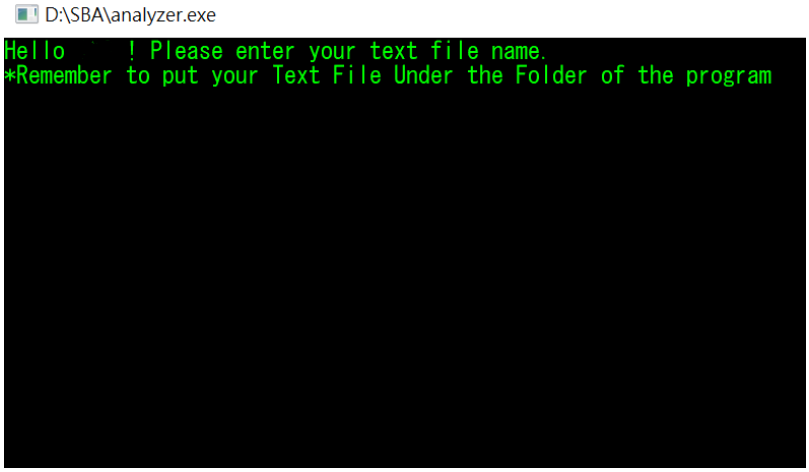
Data files are saved as ‘one line one word’ so that data can easily be read by using readln function.

TextFile that need to be analyzed



The file being analyze have to be stored in the .txt file . File name can be defined by

user freely , they have just need to enter the file name of the text



Phase 3: System implementation

3.1 Brief Description

In this chapter, I will discuss the implementation of the Education Software program. I will

I. Describe the data structures that will be used in the program
II. Describe and explain the algorithms of the functions that will be performed by each procedure in the program
III. display some of the source code
IV. display the user interface

Data Structure

Text file(.txt)

Text file is widely used in this program to store the processed data of the program . It is because that the program is form by a lot of procedures , in order to prevent too many global variable in the program , modulation of the program and convince the access of data , text file is used as a clear and effective way to organize the data, these data include :

- Dictionary
- Users' Vocab book
- username and password
- the files being read
- Wrong / Correct words
- Words that unnecessarily to be written into the Vocabulary

Array

One dimensional array

One dimensional array is the major kind of array used in the software

Array name	Array of	usage
Sentence	String	store each line of the text
Word	String	store each word of the text
TempUserNameList	String	store user name list
TempPasswordList	String	store password list
CorrectWord	String	store correct spelling words
WrongWord	String	store wrong spelling words
Frequency of word	Integer	store frequency of each word

Variables

Variable name	Variable type	usage
Logined user	String	To save login-ed username
VocabBook()	text	To save user vocab book name
counter	integer	Use as counter in insertion sort
x	integer	Used as counter in for loop
y	integer	Used as counter in for loop
z	integer	Used as counter in for loop
Dictionary()	text	To Dictionary file name(Dictionary.txt)

Parallel array

Username (String) **and** User Password (String)

Words (string) and the corresponding frequencies (integer)

Procedure Name	**Function Details**
Readfile	Function: It is used to load the text file into the array of words Program Flow(Algorithm) 1.Input Name of text file → 2.Store each sentence into Sentence [] Array → 3.Chop the sentence into words and save them into Word[] → 4.Get Total Number Of Words, Total

	Sentences, Total Paragraph
GenerateReport	Generate report include -Total number of paragraphs - Total number of words - Total number of sentences -Frequency of each word
CheckWrongWord	Check out if there are any wrong word in the text , correct words will be saved in 'CorrectWord.txt' while other will be saved in 'WrongWord.txt'
SortWord	Lower case all words and sort them by alphabetical order
SortWordForUser	Lower case all words and sort them by alphabetical order. Afterwards , user will be asked if save the new vocab into their vocab book
EditUserVocabBook	Display user's vocab book
LoginSystem	Login and go to Main body of the program

Procedure Readfile

Source code

```
Procedure ReadFile ;
var
FileName:string;
Word:array[1..1000]of string;
Sentence:array[1..30]of string;
CheckWord,NumberOfLine:integer;
wordfile:text;
GeneratedVocabList:text;
{integer for Sentence Array}
CountNoOfLn,LinePosition,NumberOfWord,SentenceLength:integer;
{integer for Sentence Array}
WordInitiallize:integer;
temp:string;
Skip:boolean;

begin
    {initialize}
    Skip:=false;
```

```
TotalSentences:=0;
TotalNumberOfParagraphs:=1;
{initialize}
clrscr;
{Load Text into Variables}
for WordInitiallize:=1 to 1000 do
      Word[WordInitiallize]:='';
{Load Text into Variables}

{Get the file}

writeln('Hello ',LoginedUser,' ! Please enter your text file name.');
writeln('*Remember to put your Text File Under the Folder of the program');
readln(FileName);
assign(wordfile,FileName+'.txt');
reset(wordfile);
{Get the file}

{Load Text into Array_Sentence}
NumberOfLine:=0;
while not EOF(wordfile)do
begin
      NumberOfLine:=NumberOfLine+1 ;
      readln(wordfile,Sentence[NumberOfLine]);
      if Sentence[NumberOfLine]='' then
          TotalNumberOfParagraphs:=TotalNumberOfParagraphs+1;
end;
close(wordfile);
{Load Text into Array_Sentence}

{Chop Sentence into word}
NumberOfWord:=1;
for CountNoOfLn:=1 to NumberOfLine do
begin
      SentenceLength:=length(Sentence[CountNoOfLn]);
      for LinePosition:=1 to SentenceLength do
      begin
            temp:=copy(Sentence[CountNoOfLn],LinePosition,1);
```

```
            if (temp <> ' ') and (temp <> '.') and (temp <> ',') and (temp <> '"') then
            begin
                Skip:=false;
                if Word[NumberOfWord]=' ' then
                Word[NumberOfWord]:=temp
                else
                    Word[NumberOfWord]:= concat(Word[NumberOfWord],temp)
            end
            else
            begin
                if temp='.' then
                    if copy(Sentence[CountNoOfLn],LinePosition+1,1)='' then
                        TotalSentences:=TotalSentences+1;
                    if Skip=false then
                    begin
                        Skip:=true;
                        NumberOfWord:=NumberOfWord+1;
                        TotalNumberOfWords:=NumberOfWord-1;
                    end;
            end;
        end;
end;

{Chop Sentence into word}


{Generate Vocab List}
Assign(GeneratedVocabList,'Vocabulary.txt');
rewrite(GeneratedVocabList);
CheckWord:=1;
while Word[CheckWord]<>'' do
begin
        writeln(GeneratedVocabList,Word[CheckWord]);
        CheckWord:=CheckWord+1;
end;
//writeln('Please Check Your Vocab in E:\Program\SBA\Vocabulary.txt');
CloseFile(GeneratedVocabList);
{Generate Vocab List}
```

end;

Explain

This function will be processed in below steps:

1.First the text file will be read line by line with the Pascal function readln and saved into an array of string called sentence

2.After that , sentence will be chopped into words , the procedure will extract the word one by one , if the string is ‘’, . ,, or blank , the system will recognize the end of a word and save the next character into new string variable

3.At the same time , there is a counter TotalNumberOfWords is used as count the number of

4.When there is a blank line , system will define it as a new paragraph , so the counter TotalNumberOfParagraph will +1

5. When there is a ‘.’, system will regard it as a new sentence so the counter TotalNumberOfSentence will +1

6. At last , all words in array Word[]will be saved into a text file 'Vocabulary.txt'.

Procedure GenerateReport

Source code

```
Procedure GenerateReport;
var
x,y:integer;
WordFrequcyFile:text;
WordFrequency:array[1..1000]of integer;
Word:array[1..1000]of string;

begin
        x:=0;
        y:=0;

        assign(WordFrequcyFile,'WordFrequcyFile.txt');

        reset(WordFrequcyFile);
        while not eof(WordFrequcyFile) do
        begin
                x:=x+1;
                readln(WordFrequcyFile,Word[x]);
                readln(WordFrequcyFile,WordFrequency[x]);
        end;
```

```
        close(WordFrequcyFile);
        writeln('Finish Analyze process ,press enter to generate Analyze report');
        readln();
        writeln('Analyze Report');
        writeln('Total words:',TotalNumberOfWords);
        writeln('Total Sentences:',TotalSentences);
        writeln('Total Paragraph:',TotalNumberOfParagraphs);
        writeln('Number of vocab :',NumberOfVocab);
        writeln('Frequency of all words');
        for y:= 1 to x do
        writeln(Word[y]:15,' : ',WordFrequency[y]);

        writeln('Please Find your vocabulary "Vocabulary.txt" at SBA/');
end;
```

Explanation

The procedure will read the word file 'WordFrequcyFile.txt' to get the Frequency of each word processed in procedure Sortword and the Total Number Of Words Sentences Paragraphs and vocab save in global variables.

Procedure CheckWrongWord

Source code

```
Procedure CheckWrongWord(TxtName:String);
var
DictionaryWord:array[1..100000]of string;
UncheckedWord:array[1..1000]of string;
WrongWord:array[1..1000]of string;
CorrectWord:array[1..1000]of string;
x,y,PointerW,PointerC,top,mid,bottom,position,count,q,NoOfWord:integer;
temp:string;
Found:boolean;
Dictionary,TxtFile,WrongWordFile,CorrectWordFile:text;
begin
        x:=0;
        y:=0;
        PointerW:=0;
        PointerC:=0;
        for x:=1 to 100000 do
```

```
        DictionaryWord[x]:='';
for x:=1 to 1000 do
begin
        UncheckedWord[x]:='';
        WrongWord[x]:='';
        CorrectWord[x]:='' ;
end;
assign(TxtFile,TxtName+'.txt');
assign(Dictionary,'Dictionary.txt');
assign(WrongWordFile,'WrongWord.txt');
assign(CorrectWordFile,'CorrectWord.txt');
reset(TxtFile);
x:=0;
while not eof(TxtFile) do
begin
        x:=x+1;
        readln(TxtFile,UncheckedWord[x])
end;
close(TxtFile);

NoOfWord:=x;
x:=0;
for x:=0 to NoOfWord-1 do
begin
        temp:=lowercase(UncheckedWord[x+1]);
        position:=1;
        for y:=1 to x do
        if temp > UncheckedWord[y] then
            position:=position+1;
        if UncheckedWord[position-1]<>temp then
        begin
                for count:=x downto position do
                UncheckedWord[count+1]:=UncheckedWord[count];
                UncheckedWord[position]:=temp;
        end
end;

y:=0;
```

```
reset(Dictionary);
while not eof(Dictionary) do
begin
        y:=y+1;
        readln(Dictionary,DictionaryWord[y]);

end;
close(Dictionary);

PointerW:=0;
for x:=1 to NoOfWord do
begin

        top:=1;
        bottom:=y;
        mid:=(top+bottom)div 2;
        found:=false;
        temp:=UncheckedWord[x];
        {
        for q:=1 to 200 do
        begin
                writeln(DictionaryWord[q]) ;
                if y mod 100 =0 then
                    readln;
        end;
        }
        repeat
                 begin
                 mid:=(top+bottom)div 2;
                 if temp<DictionaryWord[mid] then
                 begin
                     bottom:=mid-1;
                     //writeln(temp,'<',DictionaryWord[mid] ,' ',1);
                 end
                 else if temp>DictionaryWord[mid] then
                 begin
                     top:=mid+1;
                     //writeln(temp,'>',DictionaryWord[mid] ,' ',2);
```

```
                end
                else
                    begin
                    found:=true;
                    //writeln('found');
                    end;
                {
                writeln(DictionaryWord[mid]);
                writeln(top,' ',mid,' ',bottom,' ',temp,found);
                readln();
                }
                end;
            until found or (top>bottom);

             if found then
             begin
                    PointerC:=PointerC+1;
                    CorrectWord[PointerC]:=temp;
             end
             else
             begin
                    PointerW:=PointerW+1;
                    WrongWord[PointerW]:=temp;
             end;
        end;
        rewrite(WrongWordFile);
        for x:=1 to PointerW do
            writeln(WrongWordFile,WrongWord[x]);
        writeln('Please Check the wrong word in "WrongWord.txt"');
        close(WrongWordFile);

        rewrite(CorrectWordFile);
        for x:=1 to PointerC do
            writeln(CorrectWordFile,CorrectWord[x]);

        close(CorrectWordFile);

end;
```

Explanation

1.Sorted words stored in Vocabulary.txt will be read and stored into UncheckedWord array

2.Words will be compared with the words stored in dictionary.txt by Binary search.

3.Correct word will be saved into Correctword.txt , Wrong word will be saved into Wrongword.txt

Procedure SortWord

```
Procedure SortWord(TxtName:string);
var
Word,TempWordList:array[1..1000]of string;
WordFrequcyFile:text;
x,y,NoOfWord,position,pointer:integer;
count,MaxFrequencyOrder,MaxFrequency:integer;
TxtFile,Report,VocabularyBook:text;
Fchar,temp,LastSorted:string;
stop:boolean;
WordFrequencyOrder:integer;
NoRepeatWord:array[1..1000]of string;
TempNoRepeatWord:string;
TempFrequencyOfGivenWord:integer;
CountRepeat:integer;
Button:Char;
p,q,r:integer;
ok:string;

begin
        x:=0;
        y:=0;
        assign(TxtFile,TxtName+'.txt');
        reset(TxtFile);
        while not eof(TxtFile) do
        begin
                x:=x+1;
                readln(TxtFile,Word[x])
        end;
        close(TxtFile);
        NoOfWord:=x;
        //insertionsort
```

```
for x:=0 to NoOfWord-1 do
begin
        temp:=lowercase(Word[x+1]);
        position:=1;
        for y:=1 to x do
        if temp > word[y] then
             position:=position+1;
        if Word[position-1]<>temp then
        begin
                for count:=x downto position do
                Word[count+1]:=Word[count];
                Word[position]:=temp;
        end
end;
//insertionsort
q:=0;
r:=NoOfWord;

////store sorted list in to txtfile
rewrite(TxtFile);
for y:=1 to r do
      writeln(TxtFile,NoRepeatWord[y]);
close(TxtFile);
////store sorted list in to txtfile

//Get Frequency of each word
for p:=1 to 1000 do
   FrequencyOfGivenWord[p]:=1;
p:=0;
for p:=1 to NoOfWord do
begin
        temp:=Word[p];
        if (temp<>NoRepeatWord[q]) and (temp>'A')and (temp<'zzzzzzzzzz')then
        begin
                NoRepeatWord[q+1]:=temp;
                q:=q+1;
        end
        else
```

```
        begin
                r:=r-1;//words in list-1
                if temp=NoRepeatWord[q] then
                    FrequencyOfGivenWord[q]:=FrequencyOfGivenWord[q]+1;
        end;
end;

 MaxFrequency:=0;
 for y:= 1 to r do
 begin
         if FrequencyOfGivenWord[y]>MaxFrequency then
         begin
                 MaxFrequencyOrder:=y;
                 MaxFrequency:=FrequencyOfGivenWord[y];
         end;
 end;
 assign(WordFrequcyFile,'WordFrequcyFile.txt');
 rewrite(WordFrequcyFile);
 for x:=MaxFrequency downto 1 do
 begin
        y:=0;
        for y:=1 to r do
        begin
                if FrequencyOfGivenWord[y]=x then
                begin
                        writeln(WordFrequcyFile,NoRepeatWord[y]);
                        writeln(WordFrequcyFile,FrequencyOfGivenWord[y]);
                end;
        end;
 end;

 close(WordFrequcyFile);
 y:=0;
rewrite(TxtFile);
for y:=1 to r do
      writeln(TxtFile,NoRepeatWord[y]);
close(TxtFile);
NumberOfVocab:=r;
```

```
end;
```

Explanation:

This function is used to sort the word of the text file and remove repeated words.

I. lower case all words in 'vocabulary.txt' so that all words in the array can be sorted properly

II. load the words from text file to array

III. Delete repeated words and count the frequency of the word

IV. Update 'vocabulary.txt' with sorted word again

V. Save frequency of words into 'FrequencyOfWord.txt'

Procedure SortWordForUser

```
Procedure SortWordForUser(TxtName:string);
var
Word,TempWordList:array[1..1000]of string;

x,y,NoOfWord,position,pointer:integer;
count,tempO,MaxFrequency:integer;
TxtFile,Report,VocabularyBook:text;
Fchar,temp,LastSorted:string;
stop:boolean;
WordFrequencyOrder:array[1..1000]of integer;
NoRepeatWord:array[1..1000]of string;
CountRepeat:integer;
Button:Char;
p,q,r:integer;

ok:string;

begin
    x:=0;
    y:=0;
    assign(TxtFile,TxtName+'.txt');
    reset(TxtFile);
    while not eof(TxtFile) do
    begin
        x:=x+1;
        readln(TxtFile,Word[x])
    end;
    close(TxtFile);
```

```
NoOfWord:=x;
//insertionsort
for x:=0 to NoOfWord-1 do
begin
        temp:=lowercase(Word[x+1]);
        position:=1;
        for y:=1 to x do
        if temp > word[y] then
             position:=position+1;
        if Word[position-1]<>temp then
        begin
                for count:=x downto position do
                Word[count+1]:=Word[count];
                Word[position]:=temp;
        end
end;
//insertionsort
q:=0;
r:=NoOfWord;

////store sorted list in to txtfile
rewrite(TxtFile);
for y:=1 to r do
       writeln(TxtFile,NoRepeatWord[y]);
close(TxtFile);
////store sorted list in to txtfile

//Get Frequency of each word
for p:=1 to 1000 do
   FrequencyOfGivenWord[p]:=1;
for p:=1 to NoOfWord do
begin
        temp:=Word[p];

        if (temp<>NoRepeatWord[q]) and (temp>'A')and (temp<'zzzzzzzzzz')then
        begin
                NoRepeatWord[q+1]:=temp;
                q:=q+1;
```

```
            end
            else
            begin
                r:=r-1;//words in list-1
                if temp=NoRepeatWord[q] then
                    FrequencyOfGivenWord[q]:=FrequencyOfGivenWord[q]+1;
            end;
        end;

        //Get Frequency of each word
        for x:=1 to 1000 do
            WordFrequencyOrder[x]:=0;

        for x:=1 to 20 do
        begin
            MaxFrequency:=0;
            for y:= 1 to r do
            begin
                tempO:=FrequencyOfGivenWord[y];
                {
                if x=1 then
                begin
                    if tempO>MaxFrequency then
                        MaxFrequency:=y;
                end
                else
                }
                begin
                    if (tempO>MaxFrequency) and
(tempO<FrequencyOfGivenWord[WordFrequencyOrder[x-1]]) then
                        MaxFrequency:=y;
                end;
            end;
            WordFrequencyOrder[x]:=MaxFrequency;
        end;


        rewrite(TxtFile);
```

```
        for y:=1 to r do
              writeln(TxtFile,NoRepeatWord[y]);
        close(TxtFile);
        NumberOfVocab:=r;
        writeln('Finish Analyze process ,press enter to generate Analyze report');
        readln();
        writeln('Analyze Report');
        writeln('Total words:',TotalNumberOfWords);
        writeln('Total Sentences:',TotalSentences);
        writeln('Total Paragraph:',TotalNumberOfParagraphs);
        {
        for x:=1 to 10 do
              writeln(x,':',NoRepeatWord[WordFrequencyOrder[x]],'
',FrequencyOfGivenWord[WordFrequencyOrder[x]],' Times');
        }


        repeat
        begin
              writeln('Need to add vocabulary into your vocab book?(Y/N)');
              Button:=ReadKey;
        end
        until(Button='Y') or (Button='N');

        if Button='Y'then
        begin
              writeln;
              writeln('Number of vocab :',r);
              writeln('Please follow steps below:');
              writeln('1.Find "Vocabulary.txt" at SBA/');
              writeln('2.Open the file and delete unwanted words.');
              writeln('3.Save the file and close it.');
              writeln('*Please confirm that there is only one word on each line and no black lines are
added*');
              writeln('After finish above steps ,please press any button to continue.');
              while ok<>'ok' do
              begin
                    writeln('Please enter "ok" to continue.');
```

```
        readln(ok);
end;
for x:=1 to 1000 do
      Word[x]:='';
reset(TxtFile);
x:=0;
while not eof(TxtFile) do
begin
        x:=x+1;
        readln(TxtFile,Word[x])
end;
close(Txtfile);
//Merge vocab
assign(UserNameList,'UserName.txt');
reset(UserNameList);
pointer:=0;
count:=0;
while not eof(UserNameList) do
begin
        pointer:=pointer+1;
        readln(UserNameList,TempUserNameList[pointer]);
end;
close(UserNameList);
assign(VocabBook[pointer],LoginedUser+'VocabBook.txt');
reset(VocabBook[pointer]);
while not eof(VocabBook[pointer]) do
begin
        count:=count+1;
        readln(VocabBook[pointer],TempWordList[count]);
end;
close(VocabBook[pointer]);

            //count= word in original vocab book
for y:=count+1 to count+x do
      TempWordList[y]:=Word[y-count];
readln();

//insertionsort
```

```
readln();
for x:=0 to y do
begin
        temp:=TempWordList[x+1];
        position:=1;
        for r:=1 to x do
        if temp > TempWordList[r] then
             position:=position+1;
        if TempWordList[position-1]<>temp then
        begin
                for count:=x downto position do
                TempWordList[count+1]:=TempWordList[count];
                TempWordList[position]:=temp;
        end
end;

for x:=1 to 1000 do
      NoRepeatWord[x]:='';
r:=y;
q:=0;
for x:=1 to y do
begin
        temp:=TempWordList[x];
        if (temp<>NoRepeatWord[q])then
        begin
                NoRepeatWord[q+1]:=temp;
                q:=q+1;
        end
        else
        begin
                r:=r-1;//words in list-1
                if temp=NoRepeatWord[q] then
        end;
end;
rewrite(VocabBook[pointer]);
for x:=1 to r do
      writeln(VocabBook[pointer],NoRepeatWord[x]);
close(VocabBook[pointer]);
```

```
        repeat
        begin
                writeln('Thanks for using our service ,any things else?');
                writeln('1)Analyze next file');
                writeln('2)Check wrong words');
                writeln('3)Exit');
                Button:=ReadKey;
        end
        until(Button='1') or (Button='2') or (Button='3');
        if Button='1' then
        begin
                ReadFile;
                SortWordForUser('Vocabulary');
        end


        //insertionsort
        //Merge vocab
        {
        assign(VocabularyBook,'VocabBook.txt');
        rewrite(VocabularyBook);
        for y:=1 to x do
              writeln(VocabularyBook,Word[y]);
        close(VocabularyBook);
        }
end
else
begin
        repeat
        begin
                writeln('Thanks for using our service ,any things else?');
                writeln('1)Analyze next file');
                writeln('2)Check wrong words');
                writeln('3)Exit');
                Button:=ReadKey;
        end
        until(Button='1') or (Button='2') or (Button='3');
        if Button='1' then
```

```
            begin
                ReadFile;
                SortWordForUser('Vocabulary');
            end
            else if    Button='3' then

        end;
end;
```

Explanation

Similar function with Sortword

I. Call SortWord Function
II. Ask if the login-ed user to Check wrong words (call function)
III. Ask if user save new word into vocab book
IV. Ask next operation from user to quit the procedure or do it again(recursion)

Procedure EditUserVocabBook

Source code

```
var
Word:array[1..1000]of string;
VocabBook:text;
x,y:integer;
Button:char;
begin
        x:=0;
        assign(VocabBook,LoginedUser+'VocabBook.txt');
        reset(VocabBook);
        while not eof(VocabBook) do
        begin
                x:=x+1;
                readln(VocabBook,Word[x]);
        end;
        close(VocabBook);
        writeln('                Your Vocab List');
        for y:=1 to x do
              writeln(Word[y]);

        writeln('Press any button to continue.');
        readln();
```

```
        clrscr;
        repeat
        begin
                writeln('1)Essay Analyze(Vocab)');
                writeln('2)Exit');
                writeln('Press "1"or "2"to continue');
                Button:=ReadKey;
        end
        until(Button='1') or (Button='2');
        if Button='1' then
        begin
                ReadFile;
                SortWordForUser('Vocabulary');
        end
        else

end;

procedure AdministratingSystem ;
var
UserNameFile,InputFile,Dictionary:text;
x,y,z,r,position,count:integer;
filename,temp:string;
UserName:array[1..100]of string;
DictionaryWord:array[1..50000]of string;
InputWord:array[1..50000]of string;
Button:Char;
begin
        clrscr;
        writeln('Welcome back administrator,what do you want to do?');
        repeat
        begin
                writeln('1)View user list');
                writeln('2)Update Dictionary');
                writeln('Press "1"or "2"to continue');
                Button:=ReadKey;
        end
        until(Button='1') or (Button='2');
```

```
if Button='1' then
begin
        assign(UserNameFile,'UserName.txt');
        reset(UserNameFile);
        x:=0;
        while not eof (UserNameFile) do
        begin
                x:=x+1;
                readln(UserNameFile,UserName[x]);
        end;
        close(UserNameFile);
        y:=0;
        for y:=1 to x do
        writeln(UserName[y]);
        writeln('Anythings else?');
        repeat
        begin
                writeln('1)Return to main menu');
                writeln('2)Exit');
                writeln('Press "1"or "2"to continue');
                Button:=ReadKey;
        end
        until (Button='1') or (Button='2');
        if Button='1' then
        AdministratingSystem;
end

else if Button='2' then
begin
        writeln('Please enter the name of the word file');
        readln(filename);
        assign(InputFile,filename+'.txt');
        reset(Inputfile);
        readln();
        x:=0;
        while not eof (InputFile) do
        begin
                x:=x+1;
```

```
        readln(InputFile,InputWord[x]);
end;
close(InputFile);

y:=x;

assign(Dictionary,'Dictionary.txt');
reset(Dictionary);
x:=0;
while not eof (Dictionary) do
begin
        x:=x+1;
        readln(Dictionary,DictionaryWord[x]);
end;
close(Dictionary);

for z:=x+1 to x+y do
DictionaryWord[z]:=InputWord[z-x];
writeln(z);
readln();
y:=0;

for y:=x to z-1 do
begin
        writeln(DictionaryWord[1]);
        writeln(DictionaryWord[2]);
        writeln(DictionaryWord[3]);
        temp:=DictionaryWord[x+1];
        writeln(temp);
        position:=1;
        for r:=1 to y do
        if temp > DictionaryWord[r] then
        begin
                writeln(DictionaryWord[r]);
                position:=position+1;
        end;
        writeln(position);
        readln();
```

```
                    if DictionaryWord[position-1]<>temp then
                    begin
                        for count:=z downto position do
                        DictionaryWord[count+1]:=DictionaryWord[count];
                        DictionaryWord[position]:=temp;
                    end;
                end;
                rewrite(Dictionary);
                for y:= 1 to z do
                    writeln(Dictionary,DictionaryWord[y]);
                close(Dictionary);
                writeln('Inserted.');
                readln();
            end;

end;
```

Explanation

I. Display the vocabulary list

II. Choose end the program or analyze text file

procedure AdministratingSystem

```
var
UserNameFile,InputFile,Dictionary:text;
x,y,z,r,position,count:integer;
filename,temp:string;
UserName:array[1..100]of string;
DictionaryWord:array[1..50000]of string;
InputWord:array[1..50000]of string;
Button:Char;
begin
    clrscr;
    writeln('Welcome back administrator,what do you want to do?');
    repeat
    begin
        writeln('1)View user list');
        writeln('2)Update Dictionary');
        writeln('Press "1"or "2"to continue');
        Button:=ReadKey;
    end
```

```
until(Button='1') or (Button='2');
if Button='1' then
begin
        assign(UserNameFile,'UserName.txt');
        reset(UserNameFile);
        x:=0;
        while not eof (UserNameFile) do
        begin
                x:=x+1;
                readln(UserNameFile,UserName[x]);
        end;
        close(UserNameFile);
        y:=0;
        for y:=1 to x do
        writeln(UserName[y]);
        writeln('Anythings else?');
        repeat
        begin
                writeln('1)Return to main menu');
                writeln('2)Exit');
                writeln('Press "1"or "2"to continue');
                Button:=ReadKey;
        end
        until (Button='1') or (Button='2');
        if Button='1' then
        AdministratingSystem;
end

else if Button='2' then
begin
        writeln('Please enter the name of the word file');
        readln(filename);
        assign(InputFile,filename+'.txt');
        reset(Inputfile);
        readln();
        x:=0;
        while not eof (InputFile) do
        begin
```

```
        x:=x+1;
        readln(InputFile,InputWord[x]);
end;
close(InputFile);

y:=x;

assign(Dictionary,'Dictionary.txt');
reset(Dictionary);
x:=0;
while not eof (Dictionary) do
begin
        x:=x+1;
        readln(Dictionary,DictionaryWord[x]);
end;
close(Dictionary);

for z:=x+1 to x+y do
DictionaryWord[z]:=InputWord[z-x];
writeln(z);
readln();
y:=0;

for y:=x to z-1 do
begin
        writeln(DictionaryWord[1]);
        writeln(DictionaryWord[2]);
        writeln(DictionaryWord[3]);
        temp:=DictionaryWord[x+1];
        writeln(temp);
        position:=1;
        for r:=1 to y do
        if temp > DictionaryWord[r] then
        begin
                writeln(DictionaryWord[r]);
                position:=position+1;
        end;
        writeln(position);
```

```
                readln();
                if DictionaryWord[position-1]<>temp then
                begin
                        for count:=z downto position do
                        DictionaryWord[count+1]:=DictionaryWord[count];
                        DictionaryWord[position]:=temp;
                end;
        end;
        rewrite(Dictionary);
        for y:= 1 to z do
              writeln(Dictionary,DictionaryWord[y]);
        close(Dictionary);
        writeln('Inserted.');
        readln();
    end;

end;
```

Explanation

It is a procedure design for administrator . It can view user list and Update the dictionary

For View user list function:

I. Display all user name on the screen.

For Update Dictionary function:

I. Read the Words in dictionary

II. Read the word file that you need to merge with Dictionary into array Dictionary word

III. Start insertion sort with the starting point at the new word (as the original dictionary word list is sorted , it is no need to start from beginning)

IV.

Procedure LoginSystem

Source code

```
var
FoundUserName,FoundPassword,FoundUser,SameUserName:boolean;
RegisterUserName,RegisterPassword:string;
UserName,Password:string;
```

```
Pointer,x,y:integer;
Button:char;



begin
        clrscr;
        for x:=1 to 100 do
        begin
                TempUserNameList[x]:='';
                TempPasswordList[x]:='';
        end;
        TextColor(LightGreen);
        for x:=1 to ConsoleWidth do
        write('=');
        GotoXY(50,2);
        writeln('Welcome to Word Analyzer!');
        writeln('Press 1 to Log in');
        writeln('Press 2 to Register');
        writeln('Press 3 if you want to login as visitor');
        for x:=1 to ConsoleWidth do
        write('=');
        repeat
        begin
                writeln('Please Enter "1","2"or"3"to continue');
                Button:=ReadKey
        end;
        until (Button='1') or (Button='2') or(Button='3');
        if Button='3' then//Visitor
            begin
                    ReadFile;
                    SortWord('Vocabulary');
                    GenerateReport;
                    CheckWrongWord('Vocabulary');
                    //SortWord('Vocabulary');
                    writeln('Any thing else?');
                    repeat
                    begin
```

```
                writeln('1)Return to main menu');
                writeln('2)Exit');
                Button:=ReadKey
            end
            until (Button='1') or (Button='2') ;
            if Button='1' then
                LoginSystem;
        end
    else if Button='2' then//register
    begin
        SameUserName:=false;
        write('Please Enter Your username:');
        readln(UserName);
        write('Please Enter Your password:');
        readln(Password);
        writeln('Please confirm your account information:');
        writeln('UserName:',UserName);
        writeln('Password:',Password);
        repeat
        begin
            writeln('Press "C" to continue ,Press "R" to correct the
information .');
            Button:=ReadKey;
        end
        until(Button='C') or (Button='R');
        if Button='C' then
        begin
            assign(UserNameList,'UserName.txt');
            assign(PasswordList,'Password.txt');
            reset(UserNameList);
            pointer:=0;
            while not eof(UserNameList) do
            begin
                pointer:=pointer+1;
                readln(UserNameList,TempUserNameList[pointer]);
            end;
            close(UserNameList);
            pointer:=0;
```

```
                    reset(PasswordList);
                    while not eof(PasswordList) do
                    begin
                        pointer:=pointer+1;
                        readln(PasswordList,TempPasswordList[pointer]);
                    end;
                    close(PasswordList);
                    for x:=1 to pointer do
                    begin
                        if TempUserNameList[x]=UserName then
                            SameUserName:=true;
                    end;
                    if SameUserName=true then
                    begin
                        writeln('Same username is used , please use another
username');
                        writeln('Press enter to continue');
                        readln;
                        clrscr;
                        LoginSystem;
                    end
                    else
                    begin
                        pointer:=pointer+1;
                        TempUserNameList[pointer]:=UserName;
                        TempPasswordList[pointer]:=Password;
                        assign(VocabBook[pointer],UserName+'VocabBook.txt');
                        rewrite(VocabBook[pointer]);
                        close(VocabBook[pointer]);
                        rewrite(UserNameList);
                        for x:=1 to pointer do
                            writeln(UserNameList,TempUserNameList[x]);
                        close(UserNameList);
                        rewrite(PasswordList);
                        for x:=1 to pointer do
                            writeln(PasswordList,TempPasswordList[x]);
                        close(PasswordList);
                        writeln('Congratulations! You have singed up successfully');
```

```
                writeln('Please enter any button to return main menu');
                clrscr;
                LoginSystem;
            end;
        end
        else
        begin
        clrscr;
        LoginSystem;
        end;
end
else if Button='1' then
begin
        FoundUserName:=false;
        FoundPassword:=false;
        write('Please Enter Your username:');
        readln(UserName);
        write('Please Enter Your password:');
        readln(Password);
        assign(UserNameList,'UserName.txt');
        assign(PasswordList,'Password.txt');
        reset(UserNameList);
        pointer:=0;
        while not eof(UserNameList) do
        begin
            pointer:=pointer+1;
            readln(UserNameList,TempUserNameList[pointer]);
        end;
        close(UserNameList);
        for x:=1 to pointer do
            if TempUserNameList[x]=UserName then
            begin
                FoundUserName:=true;
                pointer:=x;
            end;
        if FoundUserName=false then
        begin
            writeln('Invalid username or password,Please enter correctly
```

```
registered username and corresponding password.');
                writeln('Enter to return main menu.');
                readln();
                clrscr;
                LoginSystem;
        end
        else
        begin
                reset(PasswordList);
                for y:=1 to pointer do
                        readln(PasswordList,TempPasswordList[y]);
                close(PasswordList);
                if TempPasswordList[pointer]<> Password then
                begin
                        writeln('Invalid username or password,Please enter
correctly registered username and corresponding password.');
                        writeln('Enter to return main menu.');
                        readln();
                        clrscr;
                        LoginSystem;
                end

                else if(UserName='admin') then
                begin
                        writeln('found admin');
                        readln();
                        AdministratingSystem;
                end

                else
                begin
                clrscr;
                LoginedUser:=UserName;

                writeln('Welcome Back ',LoginedUser);

                writeln('What do you want to do?');
                repeat
```

```
                    begin
                        writeln('1)Essay Analyze(Vocab)');
                        writeln('2)Check VocabBook');
                        writeln('Press "1"or "2"to continue');
                        Button:=ReadKey;
                    end
                    until(Button='1') or (Button='2');
                    if Button='1' then
                    begin
                      ReadFile;
                      SortWordForUser('Vocabulary');
                    end
                    else
                      EditUserVocabBook;

                    end;
                end;
            end;
end;
```

Explanation

Main Body of the program user can register , login and login in as a visitor to continue their operation

For register

I. User are request to enter username and password
II. Load the existing user list text file into array
III. Compare the register user name with the user name list . If same username is found , system will not accept the registration
IV. Rewrite the user name list and pass word file (adding new user and password)
V. Create a new text file as a user's unique vocab book

For user

I. Log in the system
II. Load the user list and password into array separately
III. User authorization , if user name / password incorrect , user will be direct back to main menu . else get in to function
    i. Essay analyze(run procedure SortWordForUser)

ii. Check VocabBook(run procedure EditVocabBook);

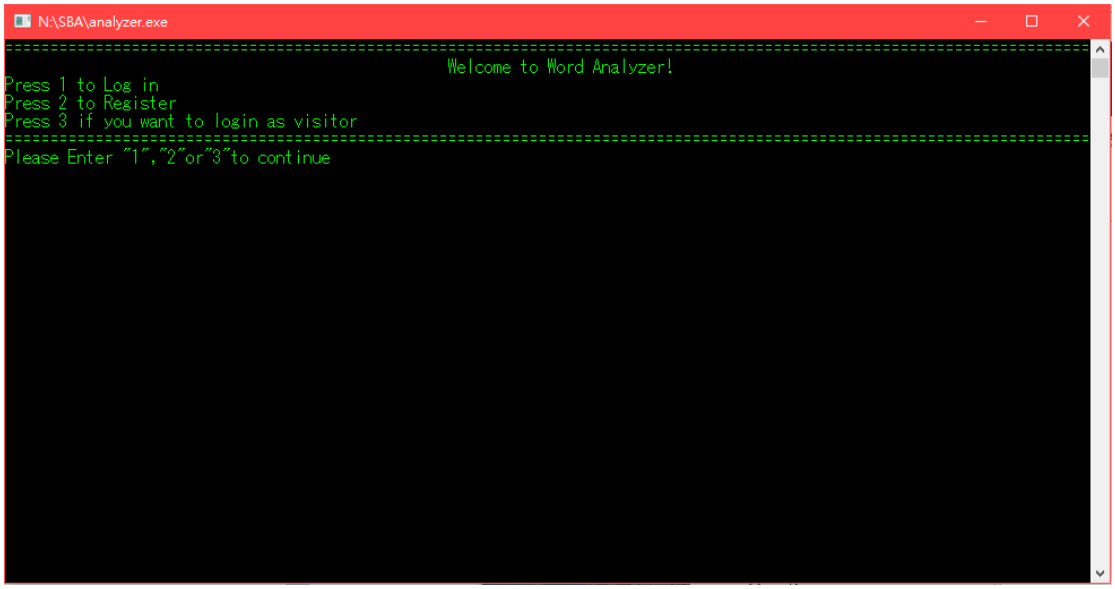
For login as visitor

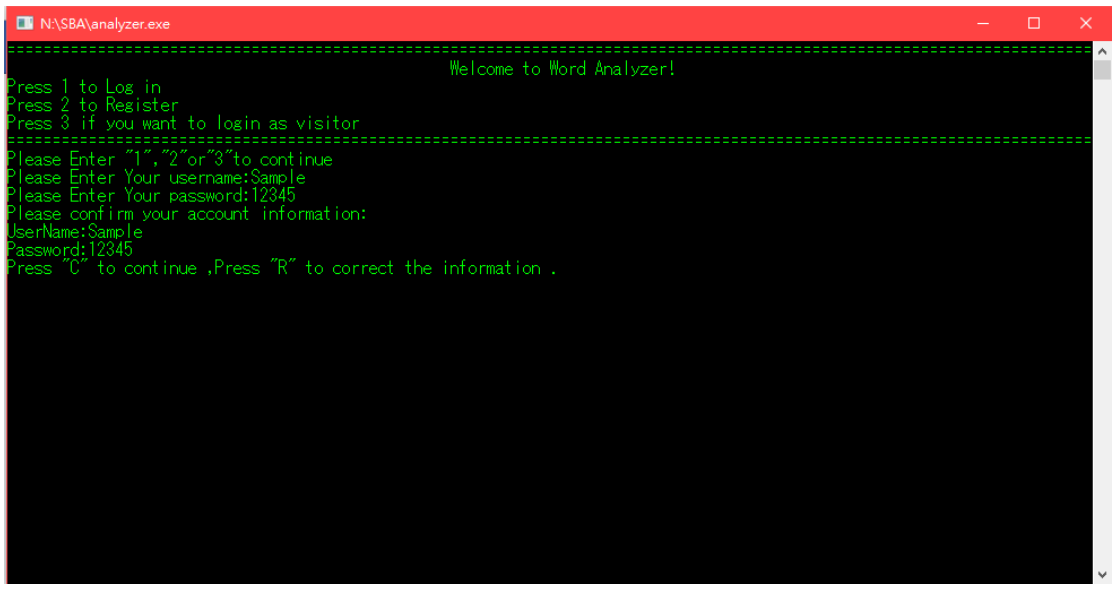
run procedure SortWord

Simple User Guide

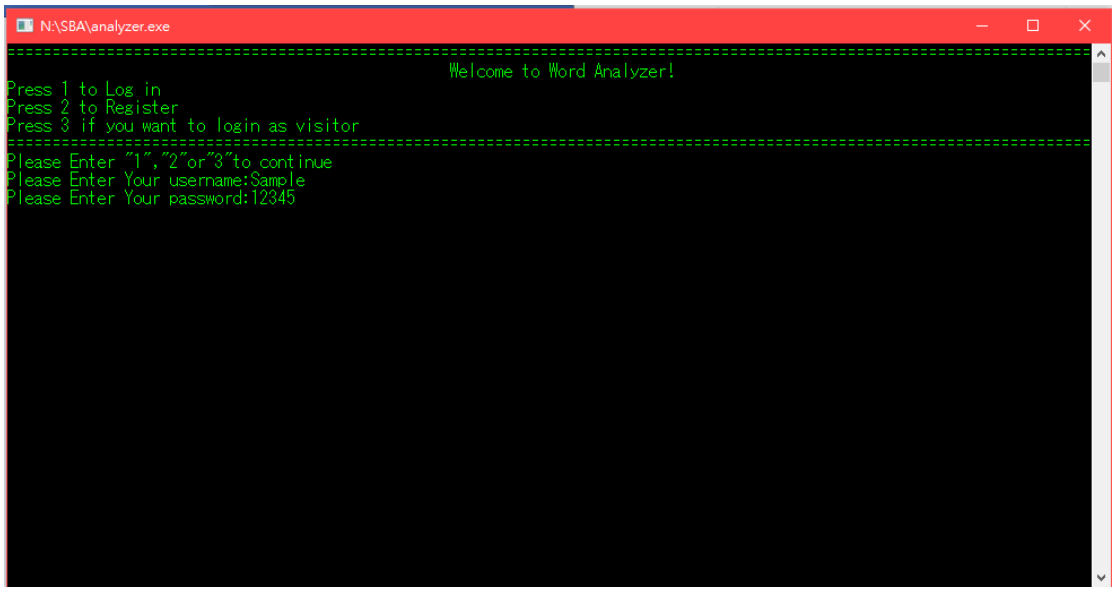
Here will show the normal operation of the program

Step 1 Register





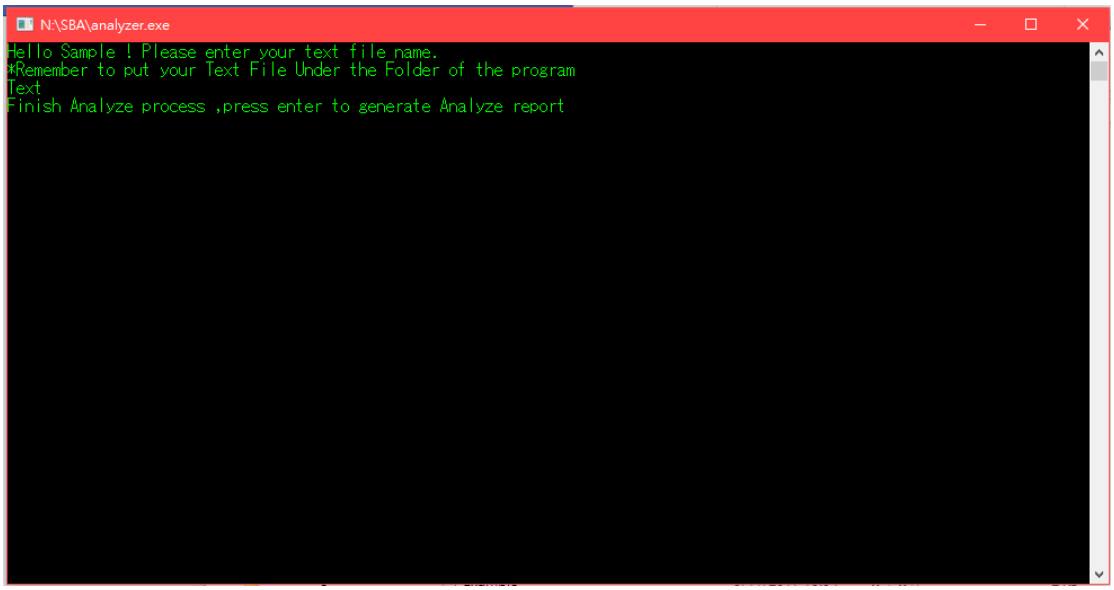
Step 2 Login

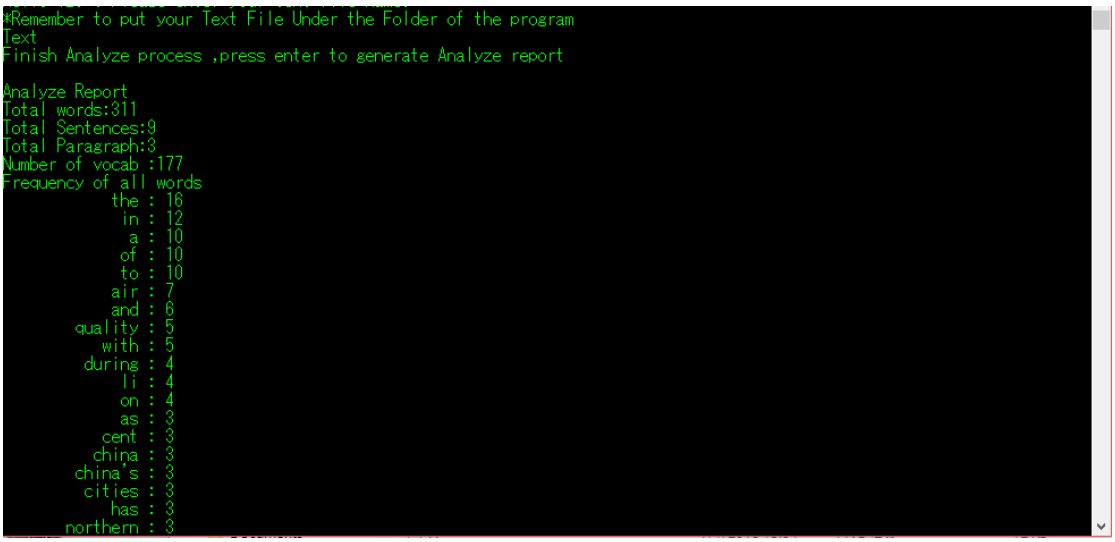


Step 3 Choose Analyze(Vocab)

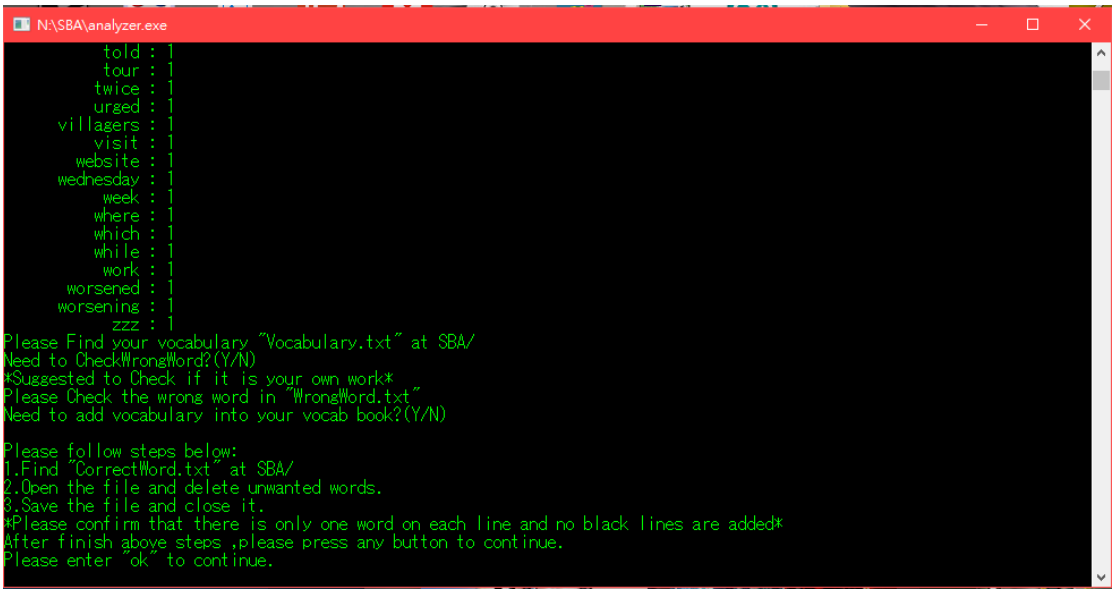
Step 4 Input File name

Step 5 Press Enter to generate report

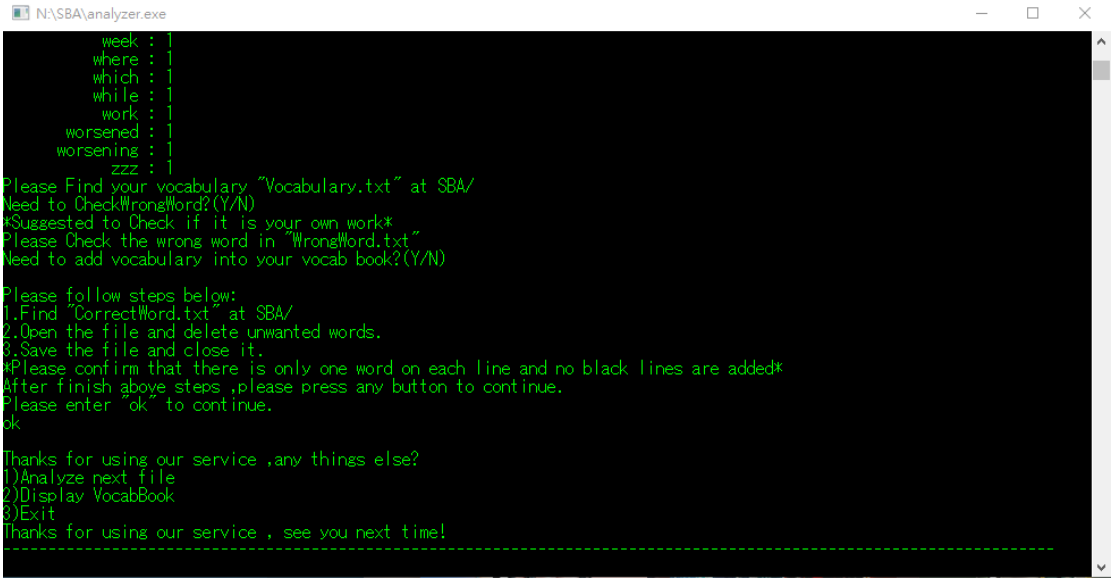




Step 6 Press 'Y' to check word



Step 7 Press '3' to exit



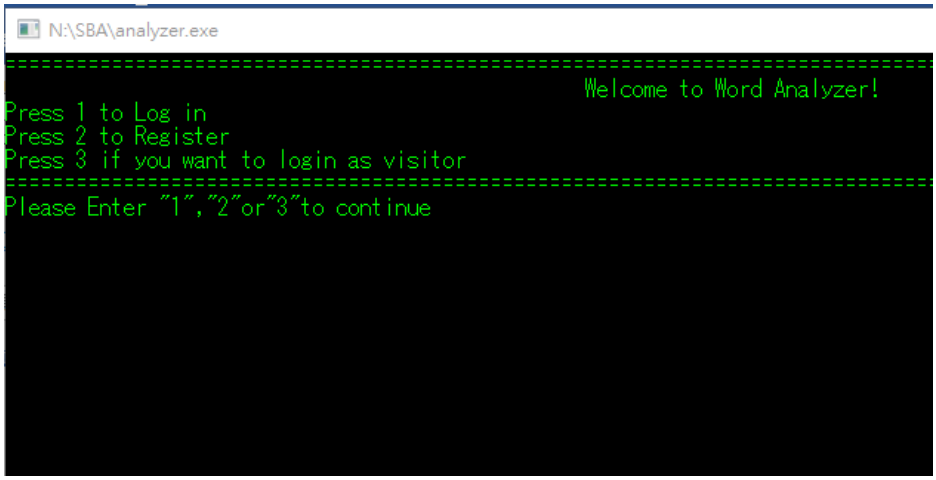
For if pressing Check VocabBook in step 3:

Phase 4: Testing and Evaluation

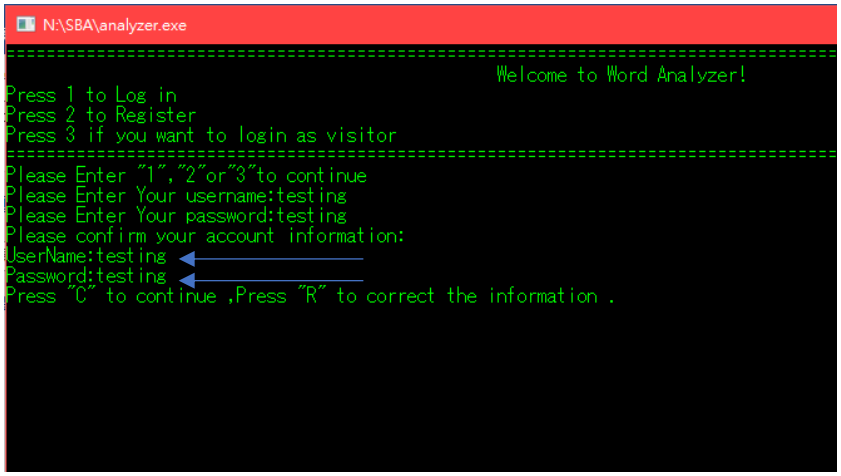
In this section I will test the procedure mentioned in Phase 3 one by one

Unit Test

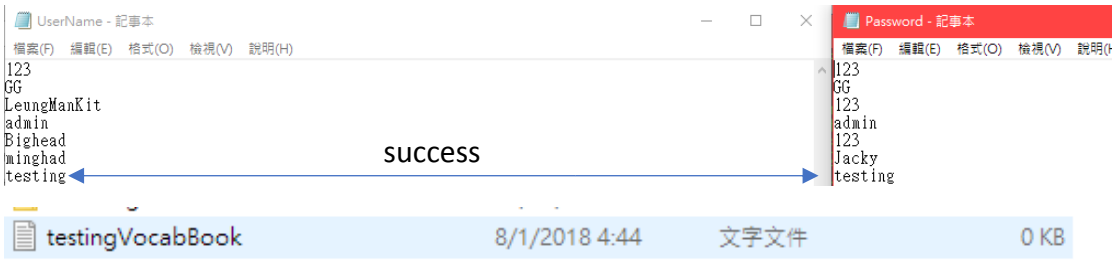
Testing:Log-in system

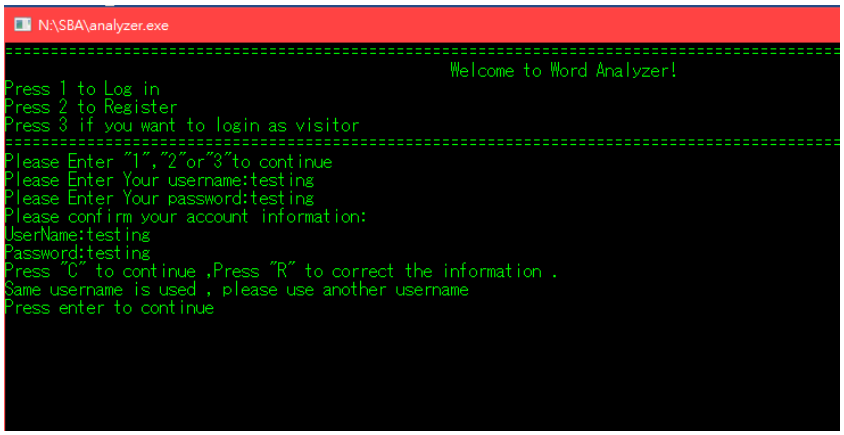


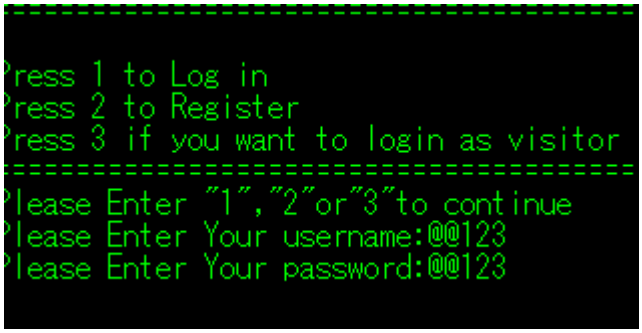
This is the starting UI



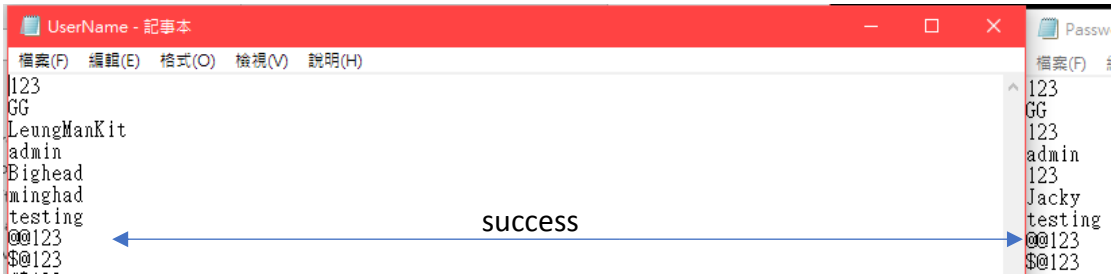
First try to register





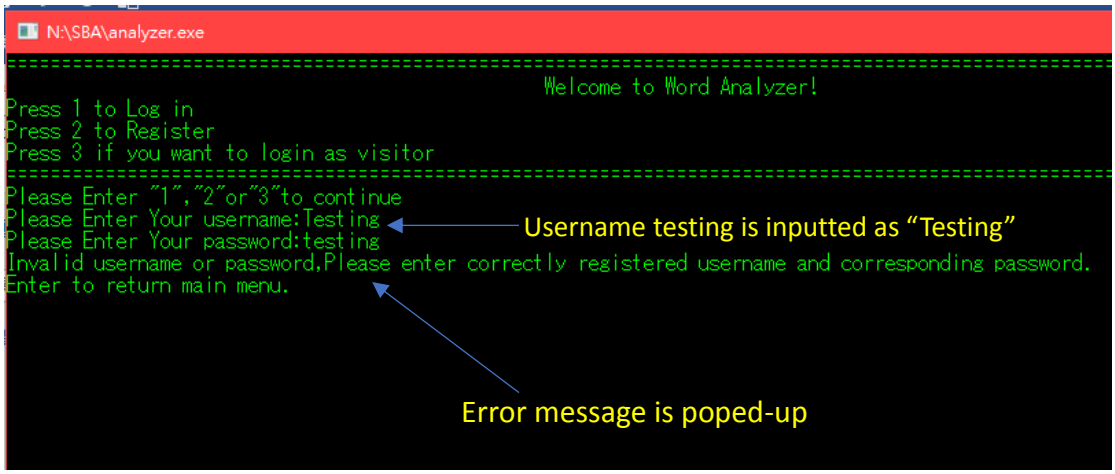


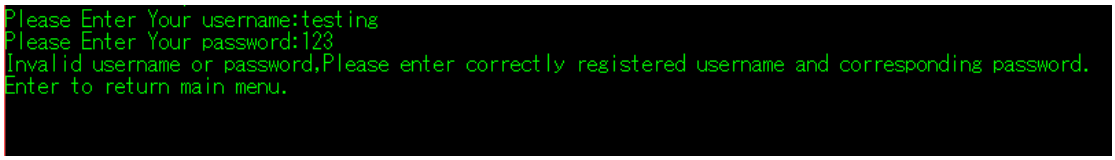
Input	Type of input	Expected output	Actual Output	Test result
testing	Valid input	Register success	Register success	Pass
@@123	boundary input	Register success	Register success	Pass

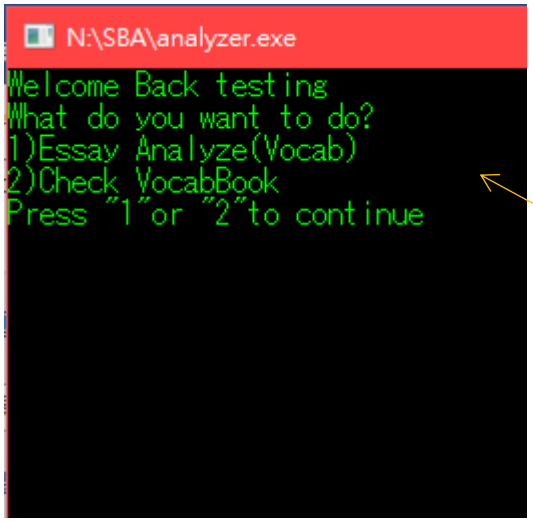


Test Log in

Input	Type of input	Expected output	Actual Output	Test result
Ac:testing Pw:testing	Valid input	Log in	Log in	Pass
Ac:Testing Pw:testing	Invalid input	Error message (Invalid username/pw)	Error message (Invalid username/pw)	Pass



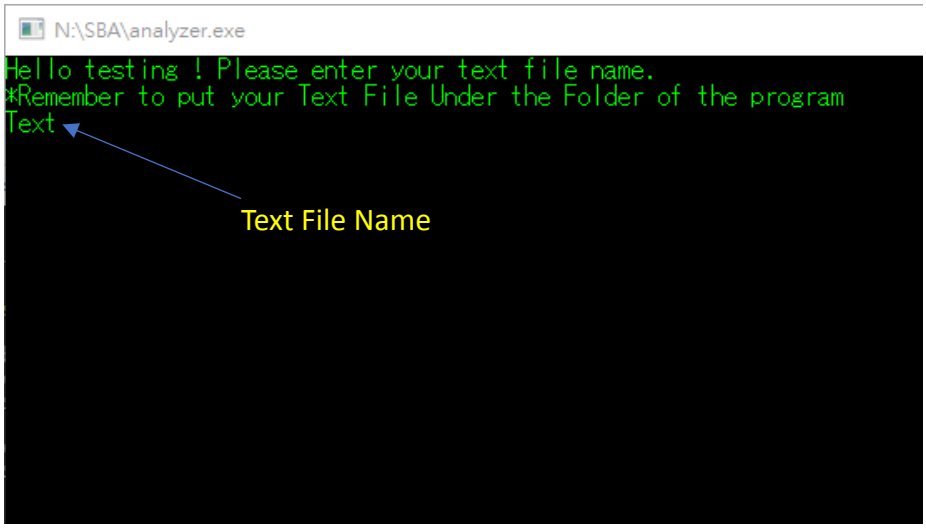


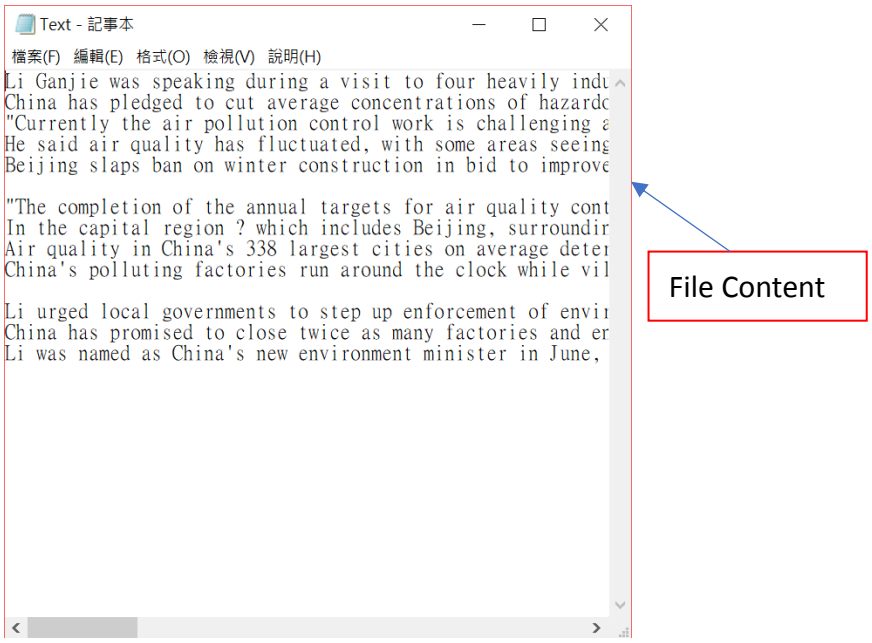


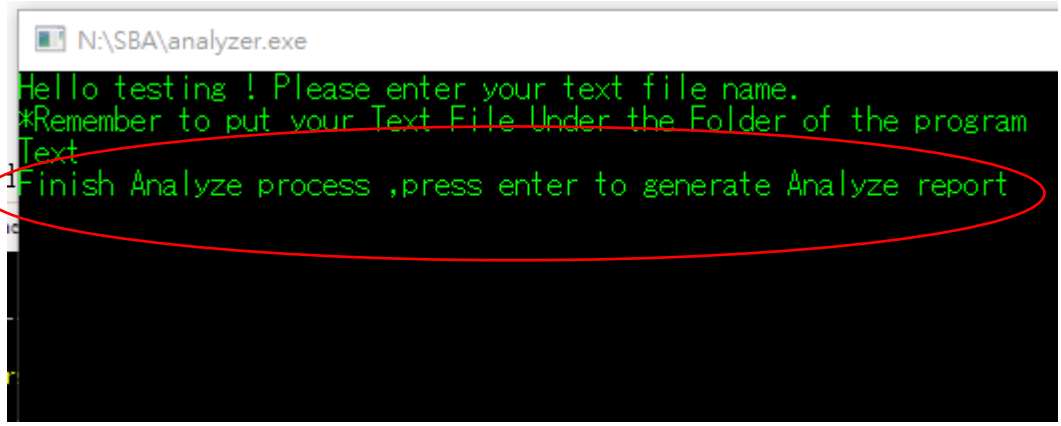
Login success interface

Input	Type of input	Expected Output	Actual Output	Test result
'1'	Valid input	Go into Essay Analyze	Essay Analyze Is ran	Pass
'2'	Valid input	Go into Check VocabBook	Check VocabBook is ran	Pass
Button except 1 ,2	Invalid	Display message	"Press '1' or '2 'to continue" shown	Pass

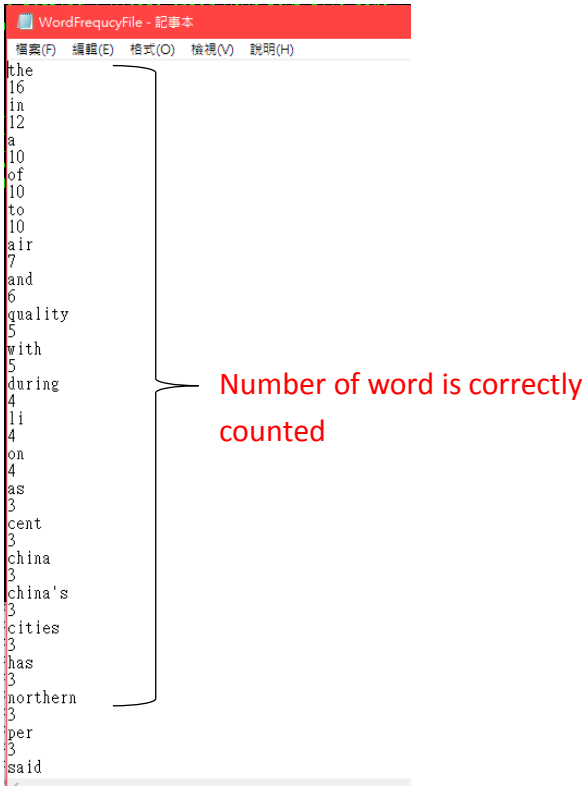
Testing:SortWordForUser

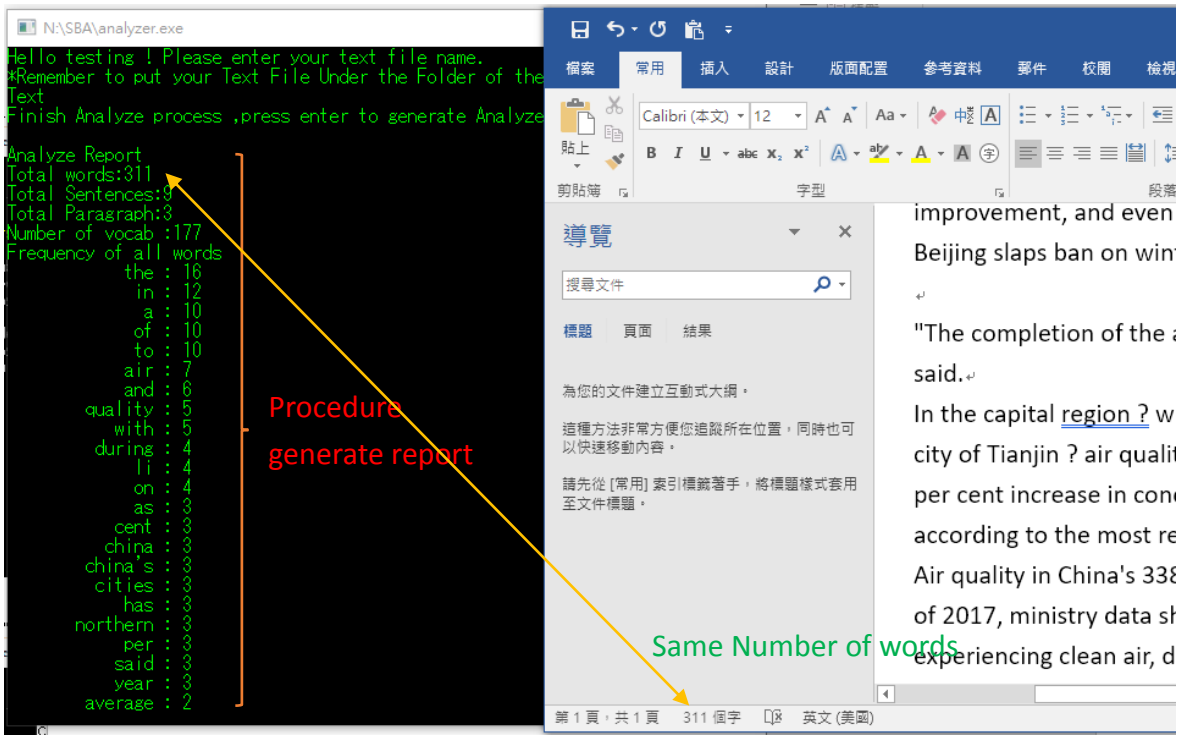







Vocabulary - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
a
aaa
accorair
according
acute
air
airborne
all
and
annual
are
area's
areas
around
as
at
august
average
avoid
ban
battle
beijing
bid
bigger
by
capital
cent
challenging
china
china's
choking
cities
city
clean
clock
close
coming
compared
completion
complex
concentrations
Words are sorted


WordFrequcyFile - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
the
16
in
12
a
10
of
10
to
10
air
7
and
6
quality
5
with
5
during
4
li
4
on
4
as
3
cent
3
china
3
china's
3
cities
3
has
3
northern
3
per
3
said
Number of word is correctly counted


N:\SBA\analyzer.exe
Hello testing ! Please enter your text file name.
#Remember to put your Text File Under the Folder of the
Text
Finish Analyze process ,press enter to generate Analyze
Analyze Report
Total words:311
Total Sentences:9
Total Paragraph:3
Number of vocab :177
Frequency of all words
the : 16
in : 12
a : 10
of : 10
to : 10
air : 7
and : 6
quality : 5
with : 5
during : 4
li : 4
on : 4
as : 3
cent : 3
china : 3
china's : 3
cities : 3
has : 3
northern : 3
per : 3
said : 3
year : 3
average : 2
Procedure generate report
Same Number of words
導覽
搜尋文件
標題 頁面 結果
第 1 頁，共 1 頁 311 個字 英文 (美國)

Testing:CheckWrongWord

```
Please Find your vocabulary "Vocabulary.txt" at SBA/
Need to CheckWrongWord?(Y/N)
*Suggested to Check if it is your own work*
```

WrongWord - 記事本

檔案(F) 編輯(E) 格式(O)

```
aaa
accorair
according
area's
areas
beijing
bigger
china's
choking
cities
concentrations
country' s
cuts
days
deteriorated
difficulties
earlier
emissionsli
enveloped
experiencing
faces
factories
fluctuated
ganjie
governments
hebei
henan
includes
industrialised
largest
laws
levels
li
ministry's
months
named
near-record
particles
pledged
pm2
points
```

CorrectWord - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V)

```
a
acute
air
airborne
all
and
annual
are
around
as
at
august
average
avoid
ban
battle
bid
by
capital
cent
challenging
china
city
clean
clock
close
coming
compared
completion
complex
construction
control
currently
cut
data
down
during
emission
end
enforce
```

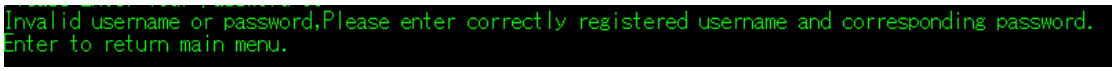
Correct Word

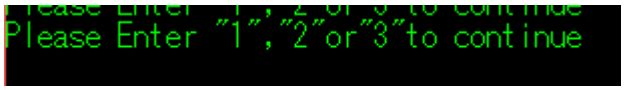
Wrong Word

Problem discover and handling

Manual input error is unavoidable , hence the program should be repeat receive data until user enter a valid input , therefore ,repeat until is widely used in my program

Error and Help Messages





Phase 5: Evaluation

After testing , I found that there are two places could be improved .

First, in the procedure CheckWrongWord ,all plural form or +s/es/ed/ies word are all wrongly identify as wrong word , in fact ,they are all correct but just changed its spelling. As program could find a completely match word in 'Dictionary', so it identify the words as wrong .

To improve the program , I think that the program could do few more operation to rise its accuracy. For example , when word 'packages' is defined as a wrong word, system could try to remove the last character of the word and check again , if it cannot found , continue the operate until third time (for at most +ies) . If it still incorrect then it is a wrong word

For some words that changed its whole word , we could only add them into dictionary through admin account.

Second, during software development , it is also found the some easy(common) words exists so frequently .Therefore , to improve the user experience , I think that common word such as pronoun and preposition (I ,you , we ,they ,is ,am are, etc). Furthermore , vocab in the user vocab book should be delete in CorrectWord.txt .Therefore , time and effort could be saved by user by deleting useless word on the list so user can edit the list of word need to add into their vocab list .

Third, File that not exist and file contain improper language (other than English)may cause crash of the program .This problem need to be solve in future development

Appendix

Acknowledgment:

New Senior Secondary Information and Communication Technology
(Software Development Elective D2)

https://www.freepascal.org/

source code

```
program analyzer;
uses Crt;
const
ConsoleHeight=30;
ConsoleWidth=120;
var
TotalNumberOfWords,TotalNumberOfParagraphs,TotalSentences,NumberOfVocab:int
eger;
FrequencyOfGivenWord:array[1..1000]of integer;
TempUserNameList:array [1..100] of string;
TempPasswordList:array [1..100] of string;
VocabBook:array[1..100]of text;
t:integer;
LoginedUser:string;
UserNameList,PasswordList:Text;
////////////////////////////////////////////////////////////////////////////
//store the words of txt file into Vocabulary.txt



Procedure ReadFile ;
var
FileName:string;
Word:array[1..1000]of string;
Sentence:array[1..30]of string;
CheckWord,NumberOfLine:integer;
```

```
wordfile:text;
GeneratedVocabList:text;

{integer for Sentence Array}
CountNoOfLn,LinePosition,NumberOfWord,SentenceLength:integer;
{integer for Sentence Array}

WordInitiallize:integer;
temp:string;
Skip:boolean;


begin
        {initialize}
        Skip:=false;
        TotalSentences:=0;
        TotalNumberOfParagraphs:=1;
        {initialize}
        clrscr;
        {Load Text into Variables}
        for WordInitiallize:=1 to 1000 do
                Word[WordInitiallize]:='';
        {Load Text into Variables}

        {Get the file}

        writeln('Hello ',LoginedUser,' ! Please enter your text file name.');
        writeln('*Remember to put your Text File Under the Folder of the program');
        readln(FileName);
        assign(wordfile,FileName+'.txt');
        reset(wordfile);
        {Get the file}

        {Load Text into Array_Sentence}
        NumberOfLine:=0;
        while not EOF(wordfile)do
```

```
        begin
                NumberOfLine:=NumberOfLine+1 ;
                readln(wordfile,Sentence[NumberOfLine]);
                if Sentence[NumberOfLine]='' then
                    TotalNumberOfParagraphs:=TotalNumberOfParagraphs+1;
        end;
        close(wordfile);
        {Load Text into Array_Sentence}

        {Chop Sentence into word}
        NumberOfWord:=1;
        for CountNoOfLn:=1 to NumberOfLine do
        begin
                SentenceLength:=length(Sentence[CountNoOfLn]);
                for LinePosition:=1 to SentenceLength do
                begin
                        temp:=copy(Sentence[CountNoOfLn],LinePosition,1);
                        if (temp <> ' ') and (temp <> '.') and (temp <> ',') and (temp <> '"')
then    {if the word dosent end}
                        begin
                                Skip:=false;
                                if Word[NumberOfWord]=' ' then
                                Word[NumberOfWord]:=temp
                                else
                                      Word[NumberOfWord]:=
concat(Word[NumberOfWord],temp)
                        end
                        else
                        begin
                                if temp='.' then
                                    if copy(Sentence[CountNoOfLn],LinePosition+1,1)=''
then
                                        TotalSentences:=TotalSentences+1;
                                    if Skip=false then
                                    begin
                                            Skip:=true;
                                            NumberOfWord:=NumberOfWord+1;
                                            TotalNumberOfWords:=NumberOfWord-1;
```

```
                                end;
                        end;
                end;
        end;

        {Chop Sentence into word}


        {Generate Vocab List}
        Assign(GeneratedVocabList,'Vocabulary.txt');
        rewrite(GeneratedVocabList);
        CheckWord:=1;
        while Word[CheckWord]<>'' do
        begin
                writeln(GeneratedVocabList,Word[CheckWord]);
                CheckWord:=CheckWord+1;
        end;
        //writeln('Please Check Your Vocab in E:\Program\SBA\Vocabulary.txt');
        CloseFile(GeneratedVocabList);
        {Generate Vocab List}
end;
////////////////////////////////////////////////////////////////////////////
{
Procedure EditDictionary
var
x:string;
Word,DictionaryWord:array [1..200000] of string;
Dictionary,Inputfile:text;
assign(Dictionary,'Dictionary.txt');
writeln('Enter the file you want to')
assign(Inputfile,x+'.txt');

}

Procedure GenerateReport;
var
x,y:integer;
WordFrequcyFile:text;
```

```
WordFrequency:array[1..1000]of integer;
Word:array[1..1000]of string;

begin
	x:=0;
	y:=0;

	assign(WordFrequcyFile,'WordFrequcyFile.txt');

	reset(WordFrequcyFile);
	while not eof(WordFrequcyFile) do
	begin
		x:=x+1;
		readln(WordFrequcyFile,Word[x]);
		readln(WordFrequcyFile,WordFrequency[x]);
	end;

	close(WordFrequcyFile);
	writeln('Finish Analyze process ,press enter to generate Analyze report');
	readln();
	writeln('Analyze Report');
	writeln('Total words:',TotalNumberOfWords);
	writeln('Total Sentences:',TotalSentences);
	writeln('Total Paragraph:',TotalNumberOfParagraphs);
	writeln('Number of vocab :',NumberOfVocab);
	writeln('Frequency of all words');
	for y:= 1 to x do
	writeln(Word[y]:15,' : ',WordFrequency[y]);

	writeln('Please Find your vocabulary "Vocabulary.txt" at SBA/');
end;


Procedure CheckWrongWord(TxtName:String);
var
DictionaryWord:array[1..100000]of string;
UncheckedWord:array[1..1000]of string;
WrongWord:array[1..1000]of string;
```

```
CorrectWord:array[1..1000]of string;
x,y,PointerW,PointerC,top,mid,bottom,position,count,q,NoOfWord:integer;
temp:string;
Found:boolean;
Dictionary,TxtFile,WrongWordFile,CorrectWordFile:text;
begin
        x:=0;
        y:=0;
        PointerW:=0;
        PointerC:=0;
        for x:=1 to 100000 do
                DictionaryWord[x]:='';
        for x:=1 to 1000 do
        begin
                UncheckedWord[x]:='';
                WrongWord[x]:='';
                CorrectWord[x]:='' ;
        end;
        assign(TxtFile,TxtName+'.txt');
        assign(Dictionary,'Dictionary.txt');
        assign(WrongWordFile,'WrongWord.txt');
        assign(CorrectWordFile,'CorrectWord.txt');
        reset(TxtFile);
        x:=0;
        while not eof(TxtFile) do
        begin
                x:=x+1;
                readln(TxtFile,UncheckedWord[x])
        end;
        close(TxtFile);

        NoOfWord:=x;
        x:=0;
        for x:=0 to NoOfWord-1 do
        begin
                temp:=lowercase(UncheckedWord[x+1]);
                position:=1;
                for y:=1 to x do
```

```
        if temp > UncheckedWord[y] then
            position:=position+1;
        if UncheckedWord[position-1]<>temp then
        begin
                for count:=x downto position do
                UncheckedWord[count+1]:=UncheckedWord[count];
                UncheckedWord[position]:=temp;
        end
end;

y:=0;
reset(Dictionary);
while not eof(Dictionary) do
begin
        y:=y+1;
        readln(Dictionary,DictionaryWord[y]);

end;
close(Dictionary);



PointerW:=0;
for x:=1 to NoOfWord do
begin

        top:=1;
        bottom:=y;
        mid:=(top+bottom)div 2;
        found:=false;
        temp:=UncheckedWord[x];
        {
        for q:=1 to 200 do
        begin
                writeln(DictionaryWord[q]) ;
                if y mod 100 =0 then
                    readln;
        end;
```

```
        }
        repeat
                begin
                mid:=(top+bottom)div 2;
                if temp<DictionaryWord[mid] then
                begin
                    bottom:=mid-1;
                    //writeln(temp,'<',DictionaryWord[mid] ,' ',1);
                end
                else if temp>DictionaryWord[mid] then
                begin
                    top:=mid+1;
                    //writeln(temp,'>',DictionaryWord[mid] ,' ',2);
                end
                else
                     begin
                     found:=true;
                     //writeln('found');
                     end;
                {
                writeln(DictionaryWord[mid]);
                writeln(top,' ',mid,' ',bottom,' ',temp,found);
                readln();
                }
                end;
        until found or (top>bottom);

         if found then
         begin
                PointerC:=PointerC+1;
                CorrectWord[PointerC]:=temp;
         end
         else
         begin
                PointerW:=PointerW+1;
                WrongWord[PointerW]:=temp;
         end;
end;
```

```
        rewrite(WrongWordFile);
        for x:=1 to PointerW do
            writeln(WrongWordFile,WrongWord[x]);
        writeln('Please Check the wrong word in "WrongWord.txt"');
        close(WrongWordFile);

        rewrite(CorrectWordFile);
        for x:=1 to PointerC do
            writeln(CorrectWordFile,CorrectWord[x]);

        close(CorrectWordFile);

end;

Procedure DeleteFromCorrectWord    ;
var
TextFile:text;
PronounFile:text;
PrepositionFile:text;
UserVocabFile:text;

DeleteWord:array [1..200] of string;
WordList:array [1..500] of string;
x,y,z,r,p:integer;
temp:string;
begin
    assign(TextFile,'Correctword.txt');
    assign(PronounFile,'Pronoun.txt');
    assign(PrepositionFile,'Preposition.txt');
    assign(UserVocabFile,LoginedUser+'VocabBook.txt');
    x:=0;
    reset(TextFile);
    while not eof(TextFile) do
    begin
        x:=x+1;
        readln(TextFile,WordList[x]);
    end;
    close(TextFile);
```

```
y:=0;
reset(PronounFile);
reset(PrepositionFile);
reset(UserVocabFile);
while not eof(PronounFile) do
begin
        y:=y+1;
        readln(PronounFile,DeleteWord[y]);
end;
close(PronounFile);
while not eof(PrepositionFile) do
begin
        y:=y+1;
        readln(PrepositionFile,DeleteWord[y]);
end;
close(PrepositionFile);
while not eof(UserVocabFile) do
begin
        y:=y+1;
        readln(UserVocabFile,DeleteWord[y]);
end;
close(UserVocabFile);
readln;
for z:=1 to x do
begin
      temp:=WordList[z];
      for r:= 1 to y do
      begin
            if temp = DeleteWord[r] then
            begin
                for p:= z+1 to x do
                  WordList[p-1]:=WordList[p];
                  x:=x-1;
            end;
      end;
end;
z:=0;
```

```
        rewrite(TextFile);
        for z:=1 to x do
                writeln(TextFile,WordList[z]);
        close(TextFile);
end;
Procedure SortWord(TxtName:string);
var
Word,TempWordList:array[1..1000]of string;
WordFrequcyFile:text;
x,y,NoOfWord,position,pointer:integer;
count,MaxFrequencyOrder,MaxFrequency:integer;
TxtFile,Report,VocabularyBook:text;
Fchar,temp,LastSorted:string;
stop:boolean;
WordFrequencyOrder:integer;
NoRepeatWord:array[1..1000]of string;
TempNoRepeatWord:string;
TempFrequencyOfGivenWord:integer;
CountRepeat:integer;
Button:Char;
p,q,r:integer;

ok:string;

begin
        x:=0;
        y:=0;
        assign(TxtFile,TxtName+'.txt');
        reset(TxtFile);
        while not eof(TxtFile) do
        begin
                x:=x+1;
                readln(TxtFile,Word[x])
        end;
        close(TxtFile);
        NoOfWord:=x;
        //insertionsort
        for x:=0 to NoOfWord-1 do
```

```
        begin
                temp:=lowercase(Word[x+1]);
                position:=1;
                for y:=1 to x do
                if temp > word[y] then
                     position:=position+1;
                if Word[position-1]<>temp then
                begin
                        for count:=x downto position do
                        Word[count+1]:=Word[count];
                        Word[position]:=temp;
                end
        end;
        //insertionsort
        q:=0;
        r:=NoOfWord;

        ////store sorted list in to txtfile
        rewrite(TxtFile);
        for y:=1 to r do
              writeln(TxtFile,NoRepeatWord[y]);
        close(TxtFile);
        ////store sorted list in to txtfile

        //Get Frequency of each word
        for p:=1 to 1000 do
           FrequencyOfGivenWord[p]:=1;
        p:=0;
        for p:=1 to NoOfWord do
        begin
                temp:=Word[p];
                if (temp<>NoRepeatWord[q]) and (temp>'A')and
(temp<'zzzzzzzzzz')then
                begin
                        NoRepeatWord[q+1]:=temp;
                        q:=q+1;
                end
                else
```

```
        begin
                r:=r-1;//words in list-1
                if temp=NoRepeatWord[q] then
                    FrequencyOfGivenWord[q]:=FrequencyOfGivenWord[q]+1;
        end;
end;



 //r is number of word


 MaxFrequency:=0;
 for y:= 1 to r do
 begin
         if FrequencyOfGivenWord[y]>MaxFrequency then
         begin
                 MaxFrequencyOrder:=y;
                 MaxFrequency:=FrequencyOfGivenWord[y];
         end;
 end;
 assign(WordFrequcyFile,'WordFrequcyFile.txt');
 rewrite(WordFrequcyFile);
 for x:=MaxFrequency downto 1 do
 begin
        y:=0;
        for y:=1 to r do
        begin
                if FrequencyOfGivenWord[y]=x then
                begin
                        writeln(WordFrequcyFile,NoRepeatWord[y]);
                        writeln(WordFrequcyFile,FrequencyOfGivenWord[y]);
                end;
        end;
 end;

 close(WordFrequcyFile);
 y:=0;
```

```
        rewrite(TxtFile);
        for y:=1 to r do
              writeln(TxtFile,NoRepeatWord[y]);
        close(TxtFile);
        NumberOfVocab:=r;
end;



////////////////////////////////////////////////////////////////////////
Procedure SortWordForUser(TxtName:string);
var
Word,TempWordList:array[1..1000]of string;

x,y,NoOfWord,position,pointer:integer;
count,tempO,MaxFrequency:integer;
TxtFile,Report,VocabularyBook:text;
Fchar,temp,LastSorted:string;
stop,CheckedWrongWord:boolean;
WordFrequencyOrder:array[1..1000]of integer;
NoRepeatWord:array[1..1000]of string;
CountRepeat:integer;
Button:Char;
p,q,r:integer;

ok:string;

begin
        x:=0;
        y:=0;
        CheckedWrongWord:=false;
        SortWord('Vocabulary');
        GenerateReport;
        repeat
        begin
              writeln('Need to CheckWrongWord?(Y/N)');
              writeln('*Suggested to Check if it is your own work*');
```

```
            Button:=ReadKey;
        end
        until(Button='Y') or (Button='N');
        if Button='Y' then
            begin
                CheckWrongWord('Vocabulary');
                CheckedWrongWord:=true;
            end;
        repeat
        begin
            writeln('Need to add vocabulary into your vocab book?(Y/N)');
            Button:=ReadKey;
        end
        until(Button='Y') or (Button='N');
        if CheckedWrongWord=true then
            assign(TxtFile,'CorrectWord.txt')
        else
            assign(TxtFile,'Vocabulary.txt');
        if Button='Y'then
        begin
            DeleteFromCorrectWord;
            writeln;
            writeln('Please follow steps below:');
            writeln('1.Find "CorrectWord.txt" at SBA/');
            writeln('2.Open the file and delete unwanted words.');
            writeln('3.Save the file and close it.');
            writeln('*Please confirm that there is only one word on each line and no
black lines are added*');
            writeln('After finish above steps ,please press any button to continue.');
            while ok<>'ok' do
            begin
                writeln('Please enter "ok" to continue.');
                readln(ok);
            end;
            for x:=1 to 1000 do
                Word[x]:='';
            reset(TxtFile);
            x:=0;
```

```
while not eof(TxtFile) do
begin
        x:=x+1;
        readln(TxtFile,Word[x])
end;
close(Txtfile);
//Merge vocab
assign(UserNameList,'UserName.txt');
reset(UserNameList);
pointer:=0;
count:=0;
while not eof(UserNameList) do
begin
        pointer:=pointer+1;
        readln(UserNameList,TempUserNameList[pointer]);
end;
close(UserNameList);
assign(VocabBook[pointer],LoginedUser+'VocabBook.txt');
reset(VocabBook[pointer]);
while not eof(VocabBook[pointer]) do
begin
        count:=count+1;
        readln(VocabBook[pointer],TempWordList[count]);
end;
close(VocabBook[pointer]);

            //count= word in original vocab book
for y:=count+1 to count+x do
      TempWordList[y]:=Word[y-count];


//insertionsort
readln();
for x:=0 to y do
begin
        temp:=TempWordList[x+1];
        position:=1;
        for r:=1 to x do
```

```
        if temp > TempWordList[r] then
            position:=position+1;
        if TempWordList[position-1]<>temp then
        begin
                for count:=x downto position do
                TempWordList[count+1]:=TempWordList[count];
                TempWordList[position]:=temp;
        end
end;

for x:=1 to 1000 do
      NoRepeatWord[x]:='';
r:=y;
q:=0;
for x:=1 to y do
begin
        temp:=TempWordList[x];
        if (temp<>NoRepeatWord[q])then
        begin
                NoRepeatWord[q+1]:=temp;
                q:=q+1;
        end
        else
        begin
                r:=r-1;//words in list-1
                if temp=NoRepeatWord[q] then
        end;
end;
rewrite(VocabBook[pointer]);
for x:=1 to r do
      writeln(VocabBook[pointer],NoRepeatWord[x]);
close(VocabBook[pointer]);
repeat
begin
        writeln('Thanks for using our service ,any things else?');
        writeln('1)Analyze next file');
        writeln('2)Display VocabBook');
        writeln('3)Exit');
```

```
                Button:=ReadKey;
            end
            until(Button='1') or (Button='2') or (Button='3');
            if Button='1' then
            begin
                ReadFile;
                SortWordForUser('Vocabulary');
            end
           // else if Button='2' then
        end
        else
        begin
            repeat
            begin
                writeln('Thanks for using our service ,any things else?');
                writeln('1)Analyze next file');
                //writeln('2)Return to main menu');
                writeln('2)Exit');
                Button:=ReadKey;
            end
            until(Button='1') or (Button='2');// or (Button='3');
            if Button='1' then
            begin
                ReadFile;
                SortWordForUser('Vocabulary');
            end
            else if Button='2' then
            {
            begin
                SortWordForUser('Vocabulary');
            end
            else if    Button='3' then
            }
        end;
end;
////////////////////////////////////////////////////////////////////////////
```

```
Procedure EditUserVocabBook    ;
var
Word:array[1..1000]of string;
VocabBook:text;
x,y:integer;
Button:char;
begin
        x:=0;
        assign(VocabBook,LoginedUser+'VocabBook.txt');
        reset(VocabBook);
        while not eof(VocabBook) do
        begin
                 x:=x+1;
                 readln(VocabBook,Word[x]);
        end;
        close(VocabBook);
        writeln('                  Your Vocab List');
        for y:=1 to x do
              writeln(Word[y]);

        writeln('Press any button to continue.');
        readln();
        clrscr;
        repeat
        begin
                writeln('1)Essay Analyze(Vocab)');
                writeln('2)Exit');
                writeln('Press "1"or "2"to continue');
                Button:=ReadKey;
        end
        until(Button='1') or (Button='2');
        if Button='1' then
        begin
                ReadFile;
                SortWordForUser('Vocabulary');
        end
        else
```

```
end;

procedure AdministratingSystem ;
var
UserNameFile,InputFile,Dictionary:text;
x,y,z,r,position,count:integer;
filename,temp:string;
UserName:array[1..100]of string;
DictionaryWord:array[1..50000]of string;
InputWord:array[1..50000]of string;
Button:Char;
begin
        clrscr;
        writeln('Welcome back administrator,what do you want to do?');
        repeat
        begin
                writeln('1)View user list');
                writeln('2)Update Dictionary');
                writeln('Press "1"or "2"to continue');
                Button:=ReadKey;
        end
        until(Button='1') or (Button='2');
        if Button='1' then
        begin
                assign(UserNameFile,'UserName.txt');
                reset(UserNameFile);
                x:=0;
                while not eof (UserNameFile) do
                begin
                        x:=x+1;
                        readln(UserNameFile,UserName[x]);
                end;
                close(UserNameFile);
                y:=0;
                for y:=1 to x do
                writeln(UserName[y]);
                writeln('Anythings else?');
                repeat
```

```
        begin
                writeln('1)Return to main menu');
                writeln('2)Exit');
                writeln('Press "1"or "2"to continue');
                Button:=ReadKey;
        end
        until (Button='1') or (Button='2');
        if Button='1' then
        AdministratingSystem;
end

else if Button='2' then
begin
        writeln('Please enter the name of the word file');
        readln(filename);
        assign(InputFile,filename+'.txt');
        reset(Inputfile);
        readln();
        x:=0;
        while not eof (InputFile) do
        begin
                x:=x+1;
                readln(InputFile,InputWord[x]);
        end;
        close(InputFile);

        y:=x;

        assign(Dictionary,'Dictionary.txt');
        reset(Dictionary);
        x:=0;
        while not eof (Dictionary) do
        begin
                x:=x+1;
                readln(Dictionary,DictionaryWord[x]);
        end;
        close(Dictionary);
```

```
            for z:=x+1 to x+y do
            DictionaryWord[z]:=InputWord[z-x];
            writeln(z);
            readln();
            y:=0;

            for y:=x to z-1 do
            begin
                  writeln(DictionaryWord[1]);
                  writeln(DictionaryWord[2]);
                  writeln(DictionaryWord[3]);
                  temp:=DictionaryWord[x+1];
                  writeln(temp);
                  position:=1;
                  for r:=1 to y do
                  if temp > DictionaryWord[r] then
                  begin
                        writeln(DictionaryWord[r]);
                        position:=position+1;
                  end;
                  writeln(position);
                  readln();
                  if DictionaryWord[position-1]<>temp then
                  begin
                        for count:=z downto position do
                        DictionaryWord[count+1]:=DictionaryWord[count];
                        DictionaryWord[position]:=temp;
                  end;
            end;
            rewrite(Dictionary);
            for y:= 1 to z do
                  writeln(Dictionary,DictionaryWord[y]);
            close(Dictionary);
            writeln('Inserted.');
            readln();
      end;

end;
```

```
Procedure LoginSystem;
var
FoundUserName,FoundPassword,FoundUser,SameUserName:boolean;
RegisterUserName,RegisterPassword:string;
UserName,Password:string;


Pointer,x,y:integer;
Button:char;



begin
        clrscr;
        for x:=1 to 100 do
        begin
                TempUserNameList[x]:='';
                TempPasswordList[x]:='';
        end;
        TextColor(LightGreen);
        for x:=1 to ConsoleWidth do
        write('=');
        GotoXY(50,2);
        writeln('Welcome to Word Analyzer!');
        writeln('Press 1 to Log in');
        writeln('Press 2 to Register');
        writeln('Press 3 if you want to login as visitor');
        for x:=1 to ConsoleWidth do
        write('=');
        repeat
        begin
                writeln('Please Enter "1","2"or"3"to continue');
                Button:=ReadKey
        end;
        until (Button='1') or (Button='2') or(Button='3');
        if Button='3' then//Visitor
             begin
```

```
                ReadFile;
                SortWord('Vocabulary');
                GenerateReport;
                CheckWrongWord('Vocabulary');
                //SortWord('Vocabulary');
                writeln('Any thing else?');
                repeat
                begin
                        writeln('1)Return to main menu');
                        writeln('2)Exit');
                        Button:=ReadKey
                end
                until (Button='1') or (Button='2') ;
                if Button='1' then
                    LoginSystem;
            end
        else if Button='2' then//register
        begin
                SameUserName:=false;
                write('Please Enter Your username:');
                readln(UserName);
                write('Please Enter Your password:');
                readln(Password);
                writeln('Please confirm your account information:');
                writeln('UserName:',UserName);
                writeln('Password:',Password);
                repeat
                begin
                        writeln('Press "C" to continue ,Press "R" to correct the
information .');
                        Button:=ReadKey;
                end
                until(Button='C') or (Button='R');
                if Button='C' then
                begin
                        assign(UserNameList,'UserName.txt');
                        assign(PasswordList,'Password.txt');
                        reset(UserNameList);
```

```
                    pointer:=0;
                    while not eof(UserNameList) do
                    begin
                        pointer:=pointer+1;
                        readln(UserNameList,TempUserNameList[pointer]);
                    end;
                    close(UserNameList);
                    pointer:=0;
                    reset(PasswordList);
                    while not eof(PasswordList) do
                    begin
                        pointer:=pointer+1;
                        readln(PasswordList,TempPasswordList[pointer]);
                    end;
                    close(PasswordList);
                    for x:=1 to pointer do
                    begin
                        if TempUserNameList[x]=UserName then
                            SameUserName:=true;
                    end;
                    if SameUserName=true then
                    begin
                        writeln('Same username is used , please use another
username');
                        writeln('Press enter to continue');
                        readln;
                        clrscr;
                        LoginSystem;
                    end
                    else
                    begin
                        pointer:=pointer+1;
                        TempUserNameList[pointer]:=UserName;
                        TempPasswordList[pointer]:=Password;
                        assign(VocabBook[pointer],UserName+'VocabBook.txt');
                        rewrite(VocabBook[pointer]);
                        close(VocabBook[pointer]);
                        rewrite(UserNameList);
```

```
                    for x:=1 to pointer do
                        writeln(UserNameList,TempUserNameList[x]);
                    close(UserNameList);
                    rewrite(PasswordList);
                    for x:=1 to pointer do
                        writeln(PasswordList,TempPasswordList[x]);
                    close(PasswordList);
                    writeln('Congratulations! You have singed up successfully');
                    writeln('Please enter any button to return main menu');
                    clrscr;
                    LoginSystem;
                end;
        end
        else
        begin
        clrscr;
        LoginSystem;
        end;
end
else if Button='1' then
begin
        FoundUserName:=false;
        FoundPassword:=false;
        write('Please Enter Your username:');
        readln(UserName);
        write('Please Enter Your password:');
        readln(Password);
        assign(UserNameList,'UserName.txt');
        assign(PasswordList,'Password.txt');
        reset(UserNameList);
        pointer:=0;
        while not eof(UserNameList) do
        begin
                pointer:=pointer+1;
                readln(UserNameList,TempUserNameList[pointer]);
        end;
        close(UserNameList);
        for x:=1 to pointer do
```

```
                    if TempUserNameList[x]=UserName then
                    begin
                        FoundUserName:=true;
                        pointer:=x;
                    end;
                if FoundUserName=false then
                begin
                    writeln('Invalid username or password,Please enter correctly
registered username and corresponding password.');
                    writeln('Enter to return main menu.');
                    readln();
                    clrscr;
                    LoginSystem;
                end
                else
                begin
                    reset(PasswordList);
                    for y:=1 to pointer do
                        readln(PasswordList,TempPasswordList[y]);
                    close(PasswordList);
                    if TempPasswordList[pointer]<> Password then
                    begin
                        writeln('Invalid username or password,Please enter
correctly registered username and corresponding password.');
                        writeln('Enter to return main menu.');
                        readln();
                        clrscr;
                        LoginSystem;
                    end

                    else if(UserName='admin') then
                    begin
                        writeln('found admin');
                        readln();
                        AdministratingSystem;
                    end

                    else
```

```
                    begin
                    clrscr;
                    LoginedUser:=UserName;

                    writeln('Welcome Back ',LoginedUser);

                    writeln('What do you want to do?');
                    repeat
                    begin
                        writeln('1)Essay Analyze(Vocab)');
                        writeln('2)Check VocabBook');
                        writeln('Press "1"or "2"to continue');
                        Button:=ReadKey;
                    end
                    until(Button='1') or (Button='2');
                    if Button='1' then
                    begin
                      ReadFile;
                      SortWordForUser('Vocabulary');
                    end
                    else
                      EditUserVocabBook;

                    end;
                end;
            end;
end;

begin
LoginSystem;
writeln('Thanks for using our service , see you next time!');



writeln('------------------------------------------------------------------------------------------------------
---------------');
readln();
```

end.