**Dictionary Project**

## CSE 135 : Data Structure Lab

## Team Information

# Team Name: The Core Programmers

# Team Members

### Al-Amin (132-15-2625)

### Istiaq Hasan (132-15-2698)

### Parvez Hssain (132-15-2755)

**Section: B**

**Department: Computer Science and Engineering**

**Date of Submission: 27-April -2014**

**Submitted to:**

## Bushra Hossain

Lecturer,  
Department of CSE

Faculty of Science and information Technology.

**Contents**

Project Description & Input Outputs ………………………………………. 3

Data Structures and their reasons ……………………………………………. 4

Description of Algorithm …………………………………………………………. 5

* Adding word ……………………………………………….. 5
* Searching word …………………………………………….. 6
* Recent search ………………………………………………. 7
* Show recent add …………………………………………… 8
* Show word list ……………………………………………. . 8
* Lexicographic Sort ………………………………………. 9

Faced Problems …………………………………………………………..... 10

* Data Record ……………………………………………….. 10
* Sorting Words ……………………………………………. 10
* File Searching ………………………………………..….. 10
* Design Theorem ………………………………………… 10

Project Summary ……………………………………………………………. 11

Recommendation …………………………………………………………… 11

Input – Output Screen Shot ……………………………………… (14-18)

**Project Description & Input Outputs**

**O**ur Project is to make a digital dictionary using C programming language.

Here admin can add and delete words and meaning of words in English. Admin can manage everything in the program.

User can search words and get the meanings which have been provided by the admin. Also a user can add words and word meanings and they are able to see the recent changes which they have made. And they are able to see total word list.

**Data Structures and their reasons**

**The objectives of the projects are shown below…..**

* Using File function.
* Using function.
* Using pointer and Array.
* Using data structure.
* Using c language.
* Using Node.
* Using Ling list.

At first we have selected a data structure called **NODE**. In this structure we have defined some variables. They are-

* String – In which we can keep max 20 characters.
* Two Dimensional String – In which we can keep 5 characters in 1st dimension and we can keep max 20 characters in 2nd dimension.
* Integer – In which we can keep some integers.
* Link list – In which we can keep dynamic memory locations.
* Array of Pointer - Size 26.

**Description of Algorithm**

* **Adding word**

Open file for write

Intput word

Print in file this word

Called add function (char \* str)

Malloc ( q ) // q is the local variable

Copy (q->data ,str)

q->link = NULL

for(0 to 5) {

input mean[i] // mean is the local variable

copy(q->m[i],mean[i])

if(i!=4)

add more meaning (y/n)

else

you cannot enter more than meaning

}

q-mcount =i

if(dic[j]==NULL or compare( dic [ j ] -> data, str ) > 0 )

{ //dic , j, r, is the local variable

r = dic [ j ]

dic [ j ] = q

q -> link = r

return }

else

{

while ( temp != NULL ) // temp is the local variable

{

if ( ( compare ( temp -> data, str ) < 0 ) and ( ( compare ( temp -> link -> data, str ) > 0 ) or

temp -> link == NULL ) )

{

q -> link = temp -> link

temp -> link = q

return

}

temp = temp -> link

}

}

* **Searching word**

Open file for read

Input word

Called search function (char \* st)

Fr = open file

Input form file in str till end of file // str local variable

If ( search, st ) != NULL )

{

Print ( str )

Plug +1

}

If(plug==0)

Call recent\_search( st ) function

Close file

}

* **Recent search (char \*str)**

N= dic ( toupper ( str [0] ) – 65 )

Copy ( temp2, str )

Convert to upper case using strupr ( temp2 )

While( n != NULL )

Copy ( temp1, n->data )

If ( compare( strupr ( temp1 ), temp2 ) )

{

Print ( n -> data , n->m[0])

For ( 1 to n-> mcount )

Print ( n->m[j] )

Plug +1

}

N=n->link

}

If ( plug==0)

Print ( words does not exists )

}

}

* **Show recent add**

for ( 0 to 25 )

{

n = dic [ i ]

while ( n != NULL )

{

printf (n -> data, n -> m [ 0 ] )

for (1 to n -> mcount)

printf (n -> m [ j ] )

n = n -> link

}

}

* **Show word list**

fr = open file

n=0;

while ( fgets ( str[ n ], 80, fr ) != NULL)

{

print(str[n]);

n=n+1

}

Close file

}

* **Lexicographic Sort**

fr = open file

n=0;

while(fgets(str[n],80, fr) != NULL)

{

N=n+1

}

Close file

for(0 to n-1)

for(i+1 to n){

if(strcmp(str[i],str[j])>0)

{

copy(temp,str[i])

copy(str[i],str[j])

copy(str[j],temp)

}

}

fr = open file for write and right

for(0 to n){

print in file (str[i]

}

Close file

}

**Faced Problems**

**D**uring designing and making this program some problems were come and fortunately we have managed to overcome those. Let’s have a look what happened.

**Data Record**: Very beginning of the program we have faced a big problem about recording data. We searched in the internet and at last we managed it keeping a notepad file so that we can save there and fetch information if needed.

**Sorting Words:** Sorting words was one of the great problems we have ever faced. Then we made it notified to our class teacher and she helped us soulfully.

**File Searching:** As we used notepad for primary database it was hard to find the algorithm of searching data from files. It took about a week solving this.

**Design Theorem:** We wanted to make our program nicer to look at. In C programming language it wasn’t easy to solve this.

**Project Summary**

**T**his project was very useful for us as newbies. It cleared our terms of thinking and making algorithms. Finally we have made this. We have managed us to create everything from our best.

This project is one of the best examples of making a C program using data structure. As someone sees s/he will find it out so much friendly and really we made it.

We thank our class teacher from our heart. She is a great inspiration for us. She helps us in every inch. Really we are fortunate.

**Recommendation**

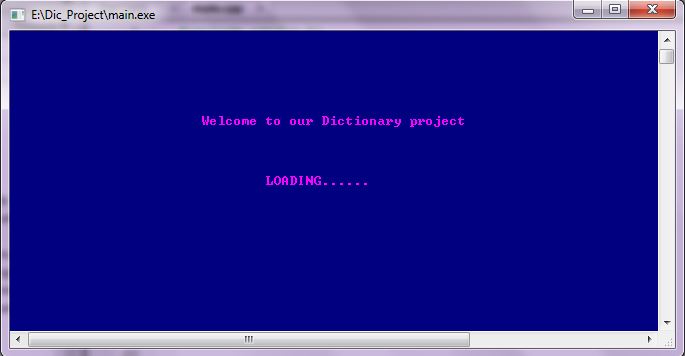
**Bushra Hossain**

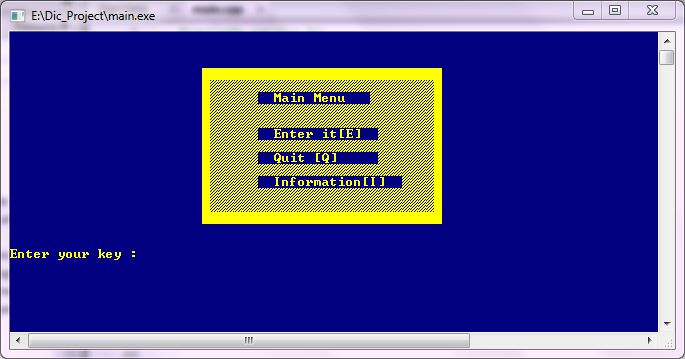
Lecturer, Dept of Computer Science and Technology

Daffodil International University

**Input – Output Screen Shot**

**Starting……**



**Home Page ….**

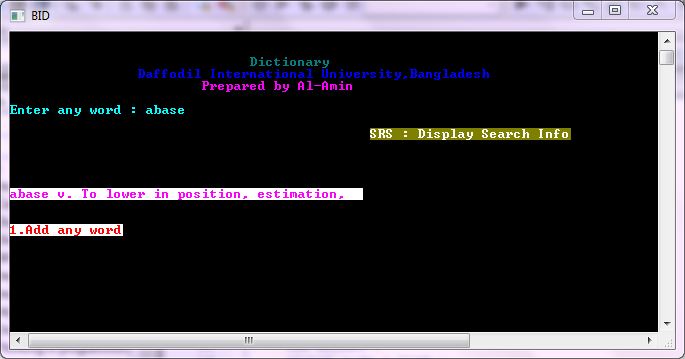
Home Page of the Projects are….

* Enter the main
* Quit the Program
* Our team Information

**Main Menu Page….**



* Add word and saving this.
* Search required word.
* Show Dictionary.
* Return Home page.

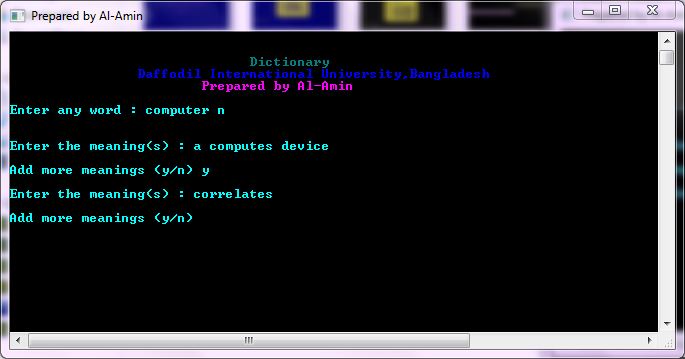
**Add word**

Enter any word

If this already exits, it shows this.

Now add another word

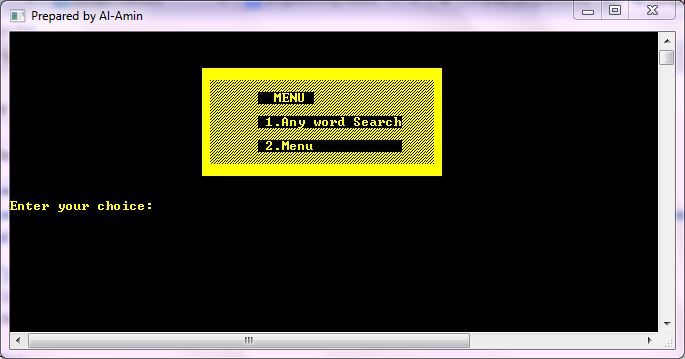
**Add Another word…..**

****

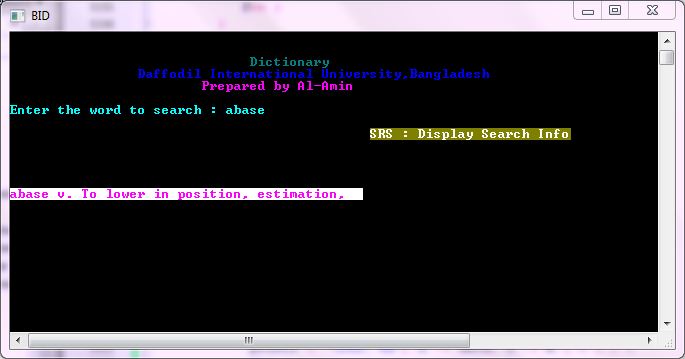
* Enter any word
* Enter the meaning
* Add more meanings
* Enter the meaning

And Program does the rest.

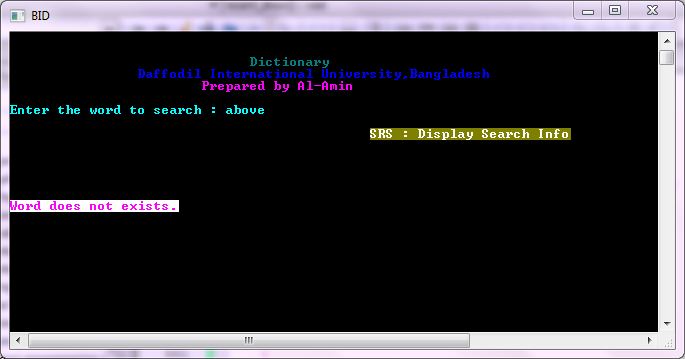
**Page for Searching word….**



* Enter word for searching
* Return Main Menu.

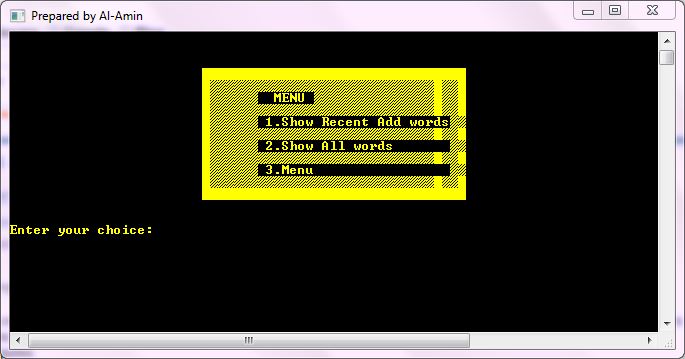


If get you’re searching word program will show this page

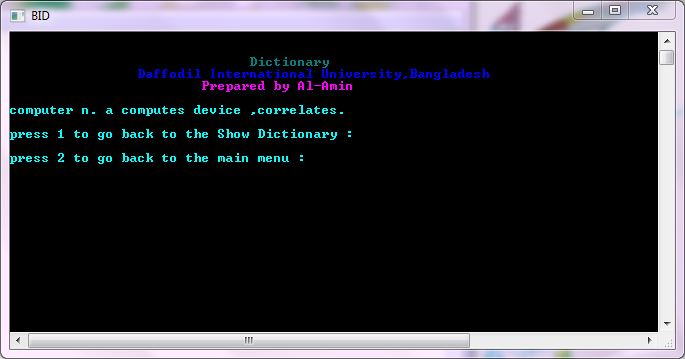


If your searching does not remain it shows this page.

**Show Dictionary page…**..



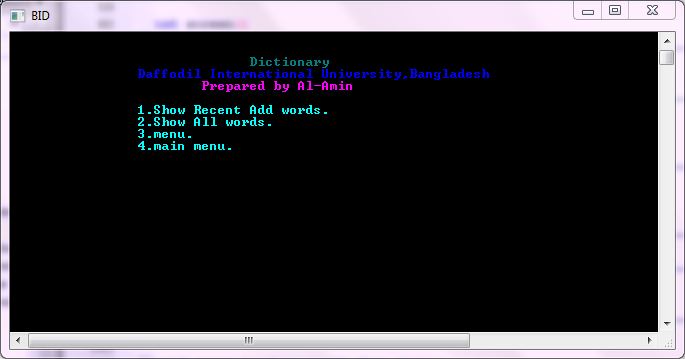
* Shows Recent add words
* Shows all words in dictionary
* Returns Main Menu



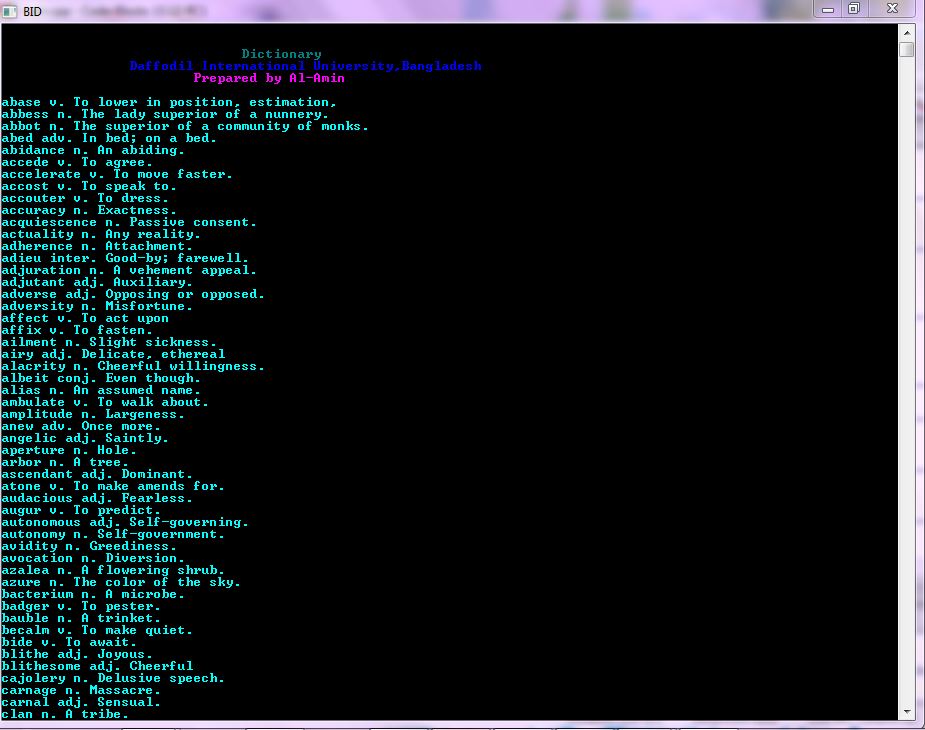
This is the interface of recently added word showing.

Returns Show Dictionary page

Returns Main Menu.

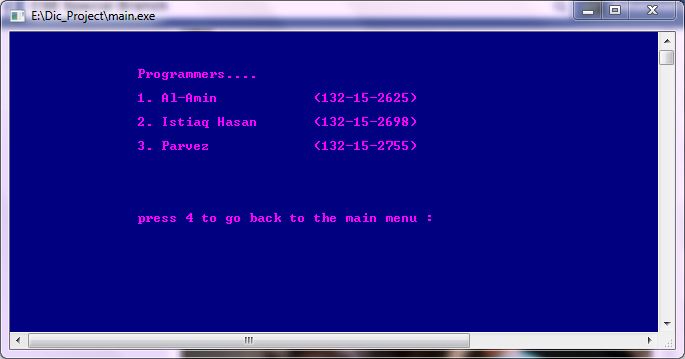


* Shows Recently added words
* Shows All word
* Returns Main menu
* Returns Home page



This page is show all word of the dictionary.

**Team information interface.**



**Exit page…..**

