Data Structures & Algorithms Lab CSL-221

*A Project Report On*

“DIGITAL DICTIONARY (USING BST)”

**Name: Umer Bahadur  
ERNO: 02-134221-007**

*Under The Supervision Of*

**Ms. Lubna Siddiqui**

**(Course Lecturer)**

**Ms. Saba Imtiaz**

**(Lab Instructor)**



Bahria University Karachi Campus

(FALL 2022)

**Table of Contents**

*Pg. No*

**Chapter 1: Introduction**

1.1 Theory 3

1.2 Project Objectives 3-4

**Chapter 2: Design & Working**

2.1 Program Description 5

2.2 Flowchart Diagrams 6-10

2.3 Code 11-17

2.4 Output Screenshots 18-21

**Chapter 3: Conclusion**  22

**CHAPTER 1: INTRODUCTION**

* 1. **Theory:**

A digital dictionary is a computer-controlled and functionally automated linguistic reference device that is specially designed to serve the lexicographic requirements of the target audience. For some people, it may refer to those inbuilt dictionaries that are used in desktop word processing programs. For others it may refer to those Machine-Readable Dictionaries (MRD) that are used with spelling and grammar checking, lemmatization, stemming and lexical database generation activities in language processing scheme. A Digital Dictionary is required in order to facilitate the targeted audience through technology.

* 1. **Project Objectives:**

The present age is the age of computer technology. At this age, the computer plays a crucial role in every sphere of our life and living starting from education, finance, health, medicine, entertainment, communication, and others. Moreover, a major part of our daily activities is carried out with the active assistance of a computer or a device of this kind. The education of students, the teaching of teachers, office activities in private and public sectors, treatment of patients, exchange of data and information from one place to another – for all activities we depend heavily on the computer either to alleviate the load of a task or to enhance speed and quality of our activities. In essence, we use a computer in various works to facilitate our working skills; gather, distribute, and interchange data and information; relieve ourselves from fatigue and boredom; save us from tension and anxiety, as well as to achieve our goals with greater accuracy and success. This is the way the world will go on, and in the future, our life and living will more and more dependent on the computer. Our future education system has no chance to escape from this change. Directly or indirectly, it will have to depend on the computer as online teaching, online assignment, online assessment, and online certification are being introduced as a part of digital education. The future growth of education and academic activities will depend on educational technology, the success of which will heavily depend on generation and availability of several linguistic resources and devices, such as, dictionaries, lexicon, grammars, thesauruses, course books, study materials, reference books, spelling checkers, grammar checkers, encyclopedias, etc. in digital form.

In the present context of the expansion of higher education, mass literacy, adult literacy, and inter-community information exchange the utility of a digital dictionary is beyond question. The functional relevance of a digital dictionary can be further visualized in general and specialized education, up-gradation of linguistic knowledge, standardization of language.

Specifically, objectives of this project will consist of:

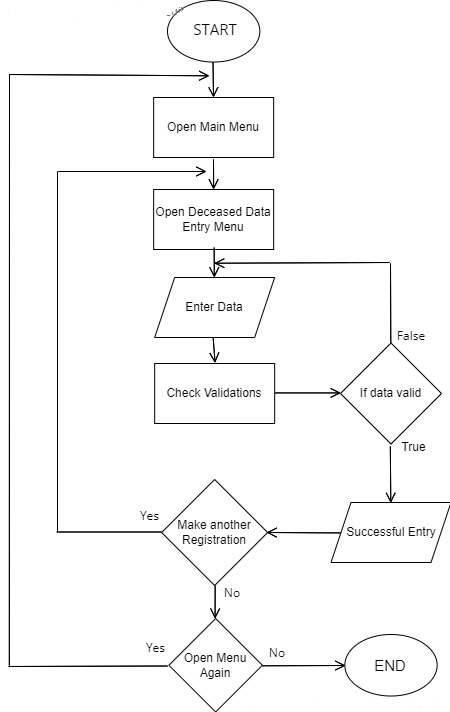
1. To design a digital dictionary that can be used to store words and meanings.
2. To design an application that can update meanings of the words in dictionary.
3. To design an application that can quickly fetch the meaning of the words.
4. To design an application to view dozens of words with their meanings at once.

**CHAPTER 2: DESIGN & WORKING**

* 1. **Program Description:**

The proposed project is a Digital Dictionary. Digital Dictionary is a machine-readable version of a standard dictionary; organized alphabetically.  Digital Dictionaries bring convenience to the user as they can easily be accessed to find the words and their meanings without having need to carry hard-cover heavy dictionaries. This also makes the learning process for students easy. The main process in this digital dictionary is the search process where user inputs the word and gets the meaning of it. User can also add a new word with its meaning if it doesn’t already exist in the dictionary. Furthermore, there is an option of deleting a word and meaning from the dictionary. Another functionality this project provides is that user can update the meaning of a word if required. The display all operation will allow the user to see all the words and the meanings altogether at once. The operations this project offers are:

1. Insert Word/Meaning: The operation will allow to enter word with its meaning.
2. Search Word/Meaning: This function will let the user search meaning of a specific word.
3. Update Meaning: The user can update the meaning of any word through this functionality.
4. Delete Word/Meaning: The feature enables the user to remove the word and a meaning from the dictionary.
5. Display All Words/Meanings: It will show the complete dictionary having words with their meanings.
6. Filing: Concept of filing is used in order to store all the words and meanings in a .txt file.
   1. **Flowchart Diagrams:**
7. Insert Word/Meaning.

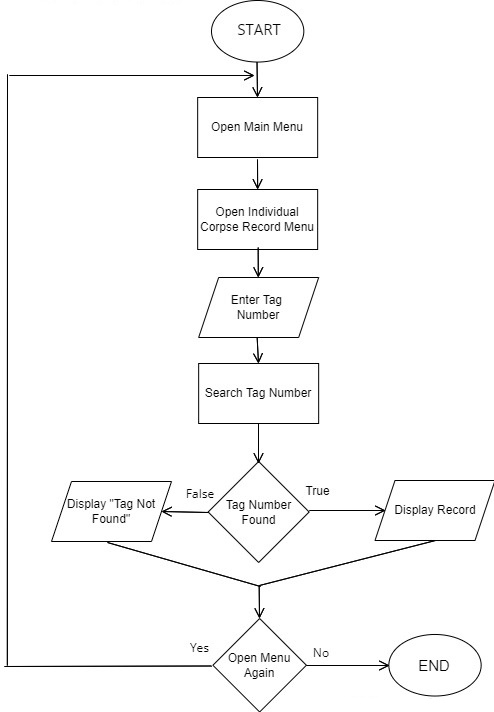


Open Insert Word Menu

Enter Word/ Meaning

Insert another word

1. Search Meaning.



Open Search Meaning Menu

Enter Word

Search Word

Display Meaning

Word Found

Display “Word not found”

1. Update Meaning.



Update Meaning

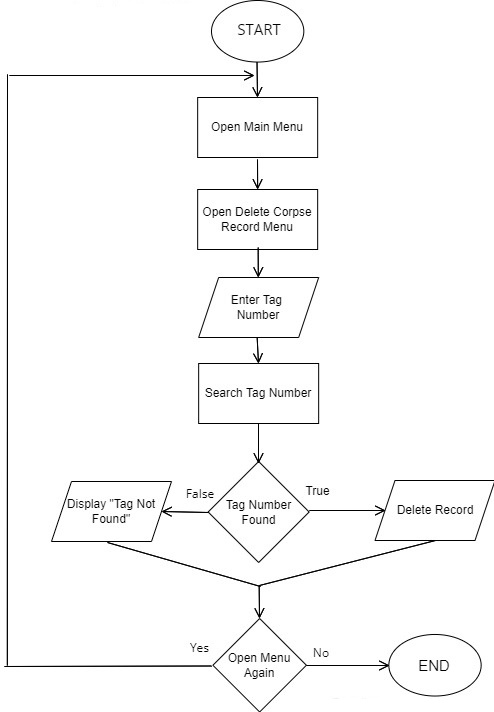
Display “Word not found”

Word Found

Search Word

Enter Word

Open Update Meaning Menu

1. Delete Word/Meaning.

Display “Word not found”

Delete Word

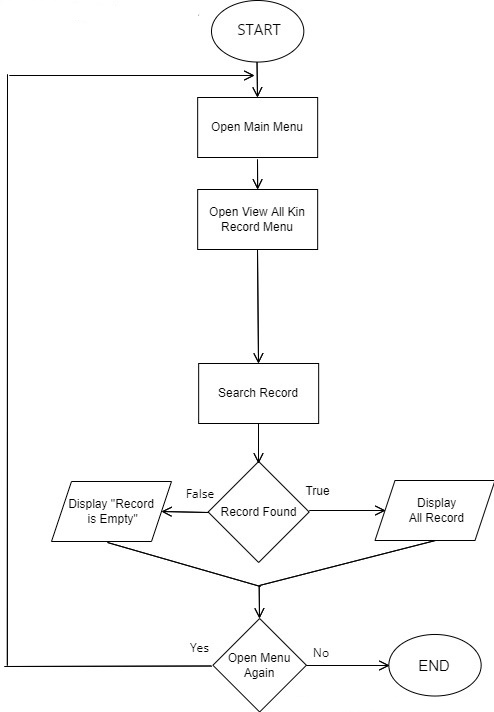
Word Found

Search Word

Enter Word

Open Delete Word Menu

1. Display Complete Dictionary.



Display All Words

Open Display All Words Menu

* 1. **Code:**

#include<iostream>

#include<stdio.h>

#include<stdlib.h>

#include<cstdio>

#include<cstring>

#include<windows.h>

#include<string.h>

#include<fstream>

using namespace std;

struct node

{

char word[20];

char meaning[300];

node\* left;

node\* right;

};

node\* root = NULL;

void writefile\_dictionary(char word[20], char meaning[300]);

node\* Create(char word[20], char meaning[300]);

node\* Insert(node\* root, char word[20], char meaning[300]);

void Display\_Dictionary(node\* root);

void search(node\* root, char word[20]);

void update(node\*& root, char word[20]);

node\* min\_val(node\* root);

node\* Delete(node\* root, char word[20]);

void menu\_header();

node\* Create(char word[20], char meaning[300])

{

node\* temp = new node();

strcpy\_s(temp->word, word);

strcpy\_s(temp->meaning, meaning);

temp->left = NULL;

temp->right = NULL;

return temp;

}

node\* Insert(node\* root, char word[20], char meaning[300])

{

if (root == NULL)

{

node\* root = Create(word, meaning);

return root;

}

if (strcmp(word, root->word) < 0)

{

root->left = Insert(root->left, word, meaning);

return root;

}

else if (strcmp(word, root->word) > 0)

{

root->right = Insert(root->right, word, meaning);

return root;

}

else if (strcmp(word, root->word) == 0)

{

cout << "The word is already in dictionary!" << endl;

return root;

}

return root;

}

void Display\_Dictionary(node\* root)

{

if (root != NULL)

{

Display\_Dictionary(root->left);

printf("Word: %s \n", root->word);

printf("Meaning: %s \n\n", root->meaning);

Display\_Dictionary(root->right);

}

}

void search(node\* root, char word[20])

{

if (root == NULL)

{

cout << endl;

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ " << endl;

cout << "| |" << endl;

cout << "| The word is not in Dictionary! |" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl << endl;

}

else if (strcmp(word, root->word) == 0)

{

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl << endl;

printf("Word: %s \n", root->word);

printf("Meaning: %s \n", root->meaning);

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

}

else if (strcmp(word, root->word) < 0)

{

return search(root->left, word);

}

else if (strcmp(word, root->word) > 0)

{

return search(root->right, word);

}

}

void update(node\*& root, char word[20])

{

if (root == NULL)

{

cout << endl;

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ " << endl;

cout << "| |" << endl;

cout << "| The word is not in Dictionary! |" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl << endl;

}

else if (strcmp(word, root->word) == 0)

{

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl << endl;

printf("Existing Meaning: %s \n", root->meaning);

cout << "Enter New Meaning: ";

scanf\_s("\n");

scanf\_s("%[^\n]%\*c", root->meaning, 300);

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl << endl;

}

else if (strcmp(word, root->word) < 0)

{

return update(root->left, word);

}

else if (strcmp(word, root->word) > 0)

{

return update(root->right, word);

}

}

node\* min\_val(node\* root)

{

node\* curr = root;

while (curr->left != NULL)

{

curr = curr->left;

}

return curr;

}

node\* Delete(node\* root, char word[20])

{

if (root == NULL)

{

return root;

}

else if (strcmp(word, root->word) < 0)

{

root->left = Delete(root->left, word);

}

else if (strcmp(word, root->word) > 0)

{

root->right = Delete(root->right, word);

}

else

{

if (root->left == NULL && root->right == NULL)

{

free(root);

return NULL;

}

else if (root->left == NULL) {

node\* temp = root->right;

free(root);

return temp;

}

else if (root->right == NULL) {

node\* temp = root->left;

free(root);

return temp;

}

else

{

node\* temp = min\_val(root->right);

strcpy\_s(root->word, temp->word);

strcpy\_s(root->meaning, temp->meaning);

root->right = Delete(root->right, temp->word);

}

}

return root;

}

void menu\_header()

{

cout << endl << endl << endl;

cout << "\t\_\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_ \_ \_\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_ \_ \_\_\_ \_\_\_\_\_\_\_\_ \_\_" << endl;

cout << "\t| \_ \\\_ \_| \_\_ \\\_ \_|\_ \_/ \_ \\ | | | \_ \\\_ \_/ \_\_ \\\_ \_|\_ \_| \_ | \\ | | / \_ \\ | \_\_\_ \\ \\ / /" << endl;

cout << "\t| | | | | | | | \\/ | | | |/ /\_\\ \\| | | | | | | | | / \\/ | | | | | | | | \\| |/ /\_\\ \\| |\_/ /\\ V / " << endl;

cout << "\t| | | | | | | | \_\_ | | | || \_ || | | | | | | | | | | | | | | | | | . ` || \_ || / \\ / " << endl;

cout << "\t| |/ / \_| |\_| |\_\\ \\\_| |\_ | || | | || |\_\_\_\_ | |/ / \_| |\_| \\\_\_/\\ | | \_| |\_\\ \\\_/ / |\\ || | | || |\\ \\ | | " << endl;

cout << "\t|\_\_\_/ \\\_\_\_/ \\\_\_\_\_/\\\_\_\_/ \\\_/\\\_| |\_/\\\_\_\_\_\_/ |\_\_\_/ \\\_\_\_/ \\\_\_\_\_/ \\\_/ \\\_\_\_/ \\\_\_\_/\\\_| \\\_/\\\_| |\_/\\\_| \\\_| \\\_/ " << endl << endl;

cout << endl << endl;

cout << "\t\t ----------------------------MAIN MENU--------------------------" << endl;

cout << "\t\t \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "\t\t| | |" << endl;

cout << "\t\t| (1) Insert Word/Meaning | (2) Search Word/Meaning |" << endl;

cout << "\t\t| (3) Update Meaning | (4) Delete Word/Meaning |" << endl;

cout << "\t\t| (5) Display All Words/Meaning | (6) Exit |" << endl;

cout << "\t\t|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cout << endl << endl;

}

void writefile\_dictionary(char word[20], char meaning[300])

{

fstream myfile;

myfile.open("Dictionary.txt", ios::app);

if (!myfile)

{

cout << "No file created";

}

else

{

myfile << "Word: " << word;

myfile << endl;

myfile << "Meaning: " << meaning;

myfile << endl;

myfile << endl;

}

myfile.close();

}

void main()

{

int choice1;

string choice2, choice3;

bool terminate = false;

char word[20], meaning[300];

cout << endl;

system("color 0E");

Beep(3020, 1100);

char l = 219;

cout << endl << endl << endl;

cout << "Loading.";

cout << endl;

for (int i = 0; i < 12; i++)

{

Beep(3000, 500);

cout << l;

}

do

{

system("cls");

menu\_header();

cout << "Enter serial no. from menu to proceed: ";

cin >> choice1;

switch (choice1)

{

case 1:

do

{

printf("\nInput Word:- ");

scanf\_s("%s", word, 20);

printf("Input Meaning:- ");

scanf\_s("\n");

scanf\_s("%[^\n]%\*c", meaning, 300);

root = Insert(root, word, meaning);

writefile\_dictionary(word, meaning);

cout << endl;

cout << "Enter another word? (yes/no): ";

cin >> choice3;

} while (choice3 == "yes" || choice3 == "Yes");

cout << endl;

break;

case 2:

printf("\nInput Word:- ");

scanf\_s("%s", word, 20);

search(root, word);

cout << endl;

break;

case 3:

printf("\nInput Word:- ");

scanf\_s("%s", word, 20);

update(root, word);

cout << endl;

break;

case 4:

printf("\nInput Word:- ");

scanf\_s("%s", word, 20);

Delete(root, word);

cout << "\nWord deleted succesfully!" << endl;

cout << endl;

break;

case 5:

cout << endl << endl << endl;

cout << "Fetching Data.";

cout << endl;

for (int i = 0; i < 16; i++)

{

Beep(2000, 500);

cout << l;

}

system("cls");

menu\_header();

cout << endl;

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

cout << endl << " | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |";

cout << endl << " | | | |";

cout << endl << " | | COMPLETE DICTIONARY | |";

cout << endl << " | |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| |";

cout << endl << " |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl << endl;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl << endl;

Display\_Dictionary(root);

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl << endl;

break;

case 6:

terminate = true;

choice2 = "no";

cout << endl;

break;

default:

cout << endl << "Invalid serial no. entered!" << endl << endl;

}

if (terminate != true)

{

cout << "Would you like to open menu again? (yes/no): ";

cin >> choice2;

cout << endl;

}

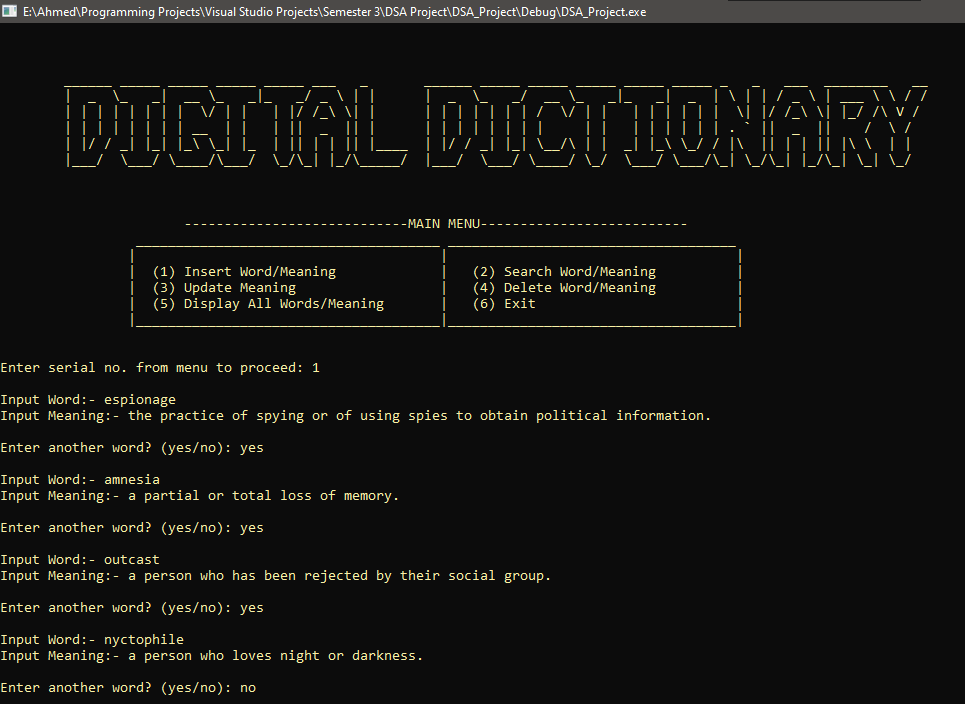
} while (choice2 == "yes" || choice2 == "Yes");

system("pause");

}

* 1. **Output Screenshots:**

1. Insert Word/Meaning.



Graphical user interface, text, application, email

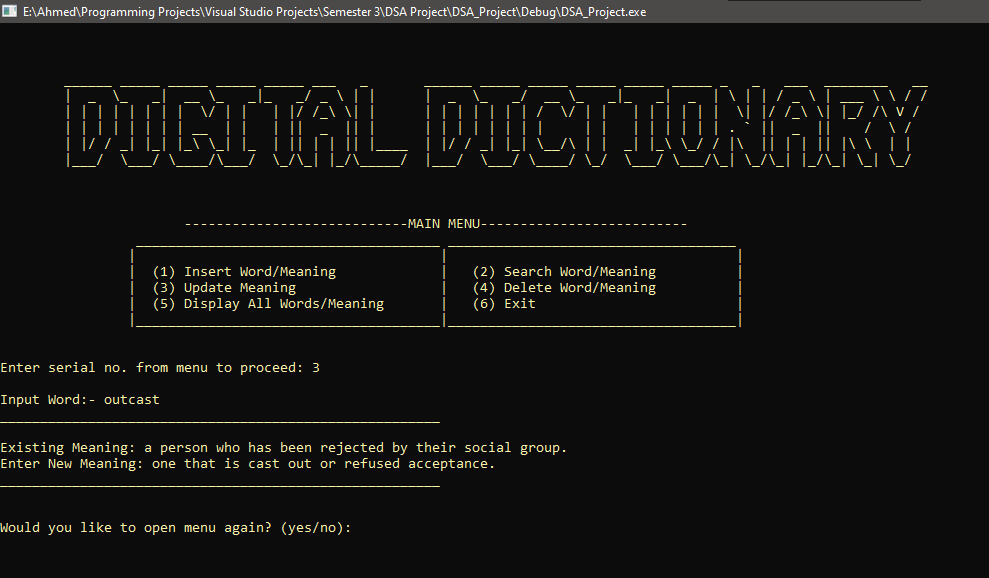
Description automatically generated

1. Search Meaning.

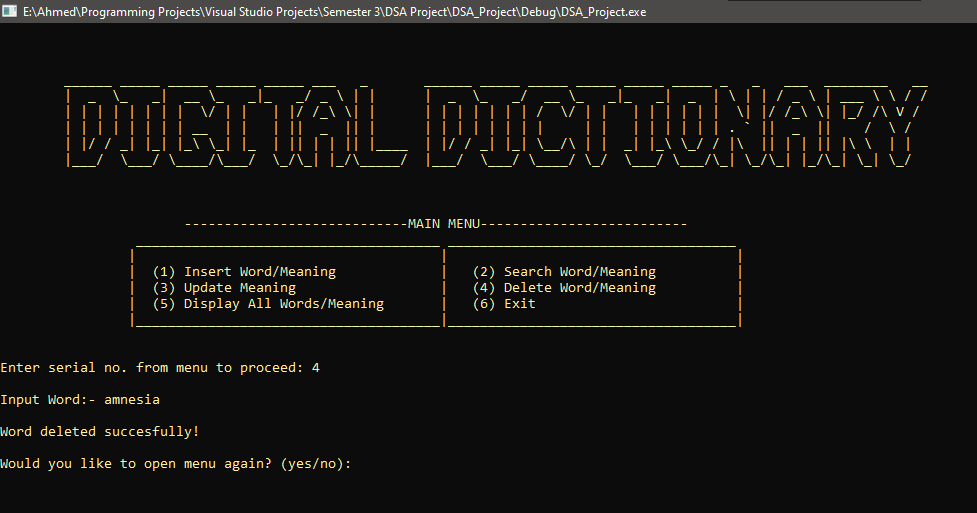
Text

Description automatically generated

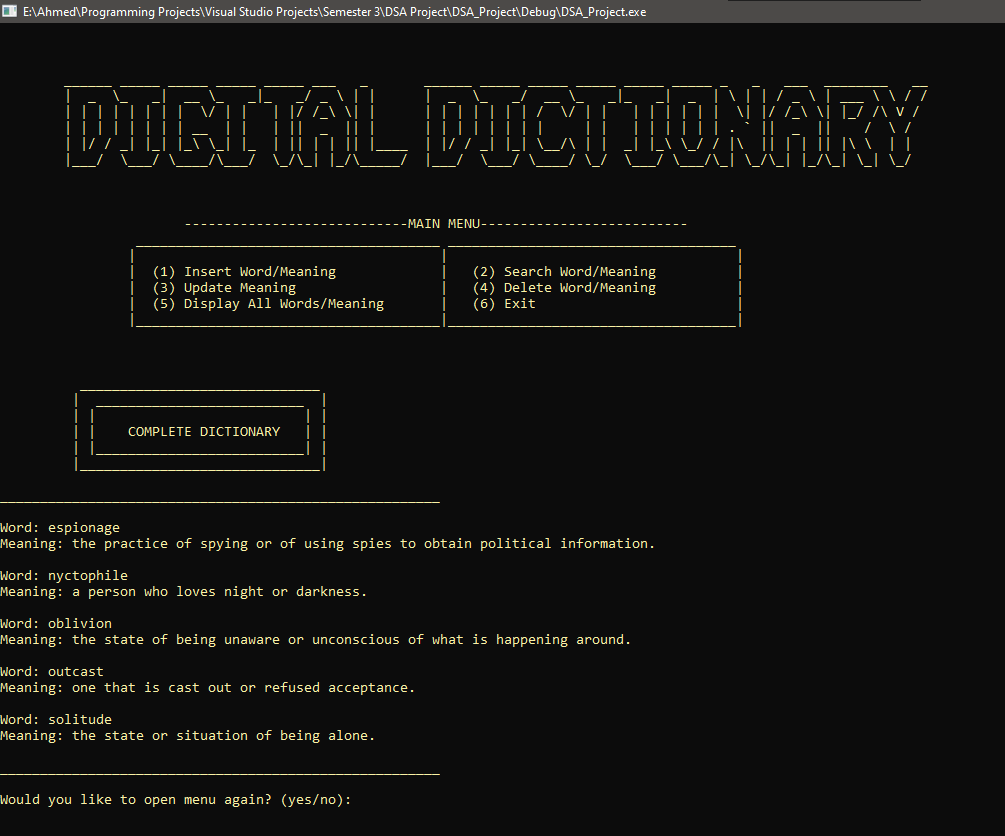
1. Update Meaning.



1. Delete Word/Meaning.



1. Display Complete Dictionary.



**CHAPTER 3: CONCLUSION**

It can be observed that dictionaries are very important in every field of human endeavor. This project provides a computerized version of dictionary which will benefit the targeted audience across many sectors. The digital version of the project lessens the use of paper like in traditional way of keeping the record. The project also has the ability to update the meanings without needing to reprint the whole edition. This development will also reduce the effort as it reduces the time used for searching particular meaning. The use of Binary Search Tree makes the project efficient and fast. All the advantages of BST are encompassed in the project and major operations of BST are applied.