#include<stdlib.h>

#include<fstream>

#include <ios>

#include<iostream>

#include <sstream>

#include <windows.h>

#include <iomanip> // SetConsoleCursorPosition()

#include <string>

using namespace std;

#define YELLOW 14

void gotoXY(int x, int y) {

COORD coord = { x, y };

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coord);

}

struct node

{

string info;

string meaning;

string synonym;

string antonym;

string findword;

node \*left\_child, \*right\_child;

};

class Tree

{

private:

public:

node \*Root;

string num;

string n,s,a;

Tree()

{

Root = NULL;

}

void insert(node \*&temp, string value, string mean,string s,string A) {

if (!temp) {

temp = new node;

temp->info = value;

//cout << "its meaning";

temp->meaning = mean;

temp->synonym = s;

temp->antonym = A;

temp->left\_child = temp->right\_child = NULL;

}

else if (value < temp->info)

insert(temp->left\_child, value, mean,s,A);

else if (value > temp->info)

insert(temp->right\_child, value, mean, s, A);

else if (value == temp->info)

{

cout << "Number is already present\n";

return;

}

}

bool search(node \*temp, string num)

{

//char b;

if (!temp)

{

cout << "not Found in Dictionary" << endl;

return false;

}

else if (temp->info == num)

{

cout << "\n\n Meaning is " << "\n" << temp->meaning << "\nSynonym is \n\n" << temp->synonym << "\nAntonym is\n \n" << temp->antonym <<endl;

//cout << temp->meaning<<endl;

return true;

}

else if (temp->info>num)

search(temp->left\_child, num);

else if (temp->info<num)

search(temp->right\_child, num);

}

};

int main(int)

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 11); //replace the 0 with a number for the color you want

gotoXY(15, 5);

//cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

char arri[10][10] = {

{ 176, 176, 176, 176, 176, 176, 176, 176, 176, 176 },

{ 176, 176, 176, 176, 176, 176, 176, 176, 176, 176 },

{ 68, 73, 67, 84, 73,79,78,65,82,89 },

{ 176, 176, 176, 176, 176, 176, 176, 176, 176, 176 },

{ 176, 176, 176, 176, 176, 176, 176, 176, 176, 176 }

};

for (int i = 0; i < 10; i++)

{

for (int j = 0; j < 10; j++)

cout <<arri[i][j];

}

///////////////////////

// system("color ");

//string word;

Tree obj;

string arr[300];

//////////////////////1///////////////////////////////////

ifstream File;

ofstream file1;

File.open("myword.txt");

for (int i = 0; i < 100; i++)

{

File >> arr[i];

}

for (int i = 0; i < 100; i += 4)

{

//cout << arr[i] << endl;

obj.num = arr[i];

obj.n = arr[i + 1];

obj.s = arr[i +2];

obj.a = arr[i + 3];

obj.insert(obj.Root, obj.num, obj.n,obj.s,obj.a);

}

//cout << endl;

string findword;

char a;

//cout << "Enter the word to find its meaning " << endl;

do{

cout << "Enter the word to find its meaning " << endl;

cin >> findword;

obj.search(obj.Root, findword);

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 14);

cout << "\n\ndo you want to find another meaning y/n " << endl;

cin >> a;

}

while (a == 'Y' || a == 'y');

char y;

string word1;

string word2,word3,word4;

int n = 101;

int p,q;

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 13);

cout << "\n\ndo you want to enter the new word (y/n) " << endl;

cin >> y;

if (y == 'Y' || y == 'y')

{

cout << "how many words you want to enter " << endl;

cin >> p;

q = p;

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

{

cout << "Enter word " << endl;

cin >> word1;

cout << " Enter Meaning" << endl;

cin >> word2;

cout << "Enter Synonym" << endl;

cin >> word3;

cout << "Enter Antonym" << endl;

cin >> word4;

arr[n] = word1;

arr[n + 1] = word2;

arr[n + 2] = word3;

arr[n + 3] = word4;

///////////////////////////////////

file1.open("myword.txt", std::ios\_base::app);

file1 << "\n";

file1 << arr[n];

file1 << "\n";

file1 << arr[n + 1];

file1 << "\n";

file1 << arr[n + 2];

file1 << "\n";

file1 << arr[n + 3];

file1.close();

obj.num = arr[n];

obj.n = arr[n + 1];

obj.s = arr[n + 2];

obj.a = arr[n + 3];

obj.insert(obj.Root, obj.num, obj.n, obj.s, obj.a);

if (q == p)

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 8); //replace the 0 with a number for the color you want

cout << "Thankyou for adding " << endl;

}

}

char k;

cout << "\n\n Do you want to search your added word(y/n) " << endl;

cin >> k;

if (k == 'Y' || k == 'y')

{

cout << "Enter your word\n " << endl;

cin >> findword;

obj.search(obj.Root, findword);

}

///////////////////////////////////////////////////////

// cout << line;

}

else

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 8); //replace the 0 with a number for the color you want

cout << "\*\*\*\*Thankyou for visiting the dictionary " << endl;

}

//system("cls");

//obj.print(obj.Root);

//cout << "Enter number to delete\n";

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

//{

// cin >> obj.num;

// obj.remove(obj.Root, obj.num);

// system("cls");

// obj.print(obj.Root);

// }

system("pause");

}