**#include<iostream>**

**#include<string.h>**

**using namespace std;**

**class node {**

**public:**

**char k[20];**

**char m[20];**

**node \*left;**

**node \*right;**

**node(char key[], char meaning[]) {**

**strcpy(k, key);**

**strcpy(m, meaning);**

**left = right = NULL;**

**}**

**};**

**class dict {**

**public:**

**node \*root;**

**void create();**

**void disp(node \*);**

**void insert(node \*&, node \*);**

**int search(node \*, char[]);**

**int update(node \*, char[]);**

**node \*del(node \*, char[]);**

**node \*min(node \*);**

**};**

**void dict::create() {**

**int ch;**

**do {**

**char key[20], meaning[20];**

**cout << "\nEnter Keyword: ";**

**cin >> key;**

**cout << "Enter Meaning: ";**

**cin >> meaning;**

**node \*temp = new node(key, meaning);**

**if (root == NULL) {**

**root = temp;**

**} else {**

**insert(root, temp);**

**}**

**cout << "\nDo you want to add more? (1/0): ";**

**cin >> ch;**

**} while (ch == 1);**

**}**

**void dict::insert(node \*&root, node \*temp) {**

**if (strcmp(temp->k, root->k) < 0) {**

**if (root->left == NULL) root->left = temp;**

**else insert(root->left, temp);**

**} else {**

**if (root->right == NULL) root->right = temp;**

**else insert(root->right, temp);**

**}**

**}**

**void dict::disp(node \*root) {**

**if (root != NULL) {**

**disp(root->left);**

**cout << "\n Key Word: " << root->k << "\t Meaning: " << root->m;**

**disp(root->right);**

**}**

**}**

**int dict::search(node \*root, char k[20]) {**

**int c = 0;**

**while (root != NULL) {**

**c++;**

**if (strcmp(k, root->k) == 0) {**

**cout << "\nNo of Comparisons: " << c;**

**return 1;**

**}**

**if (strcmp(k, root->k) < 0) root = root->left;**

**else root = root->right;**

**}**

**return -1;**

**}**

**int dict::update(node \*root, char k[20]) {**

**while (root != NULL) {**

**if (strcmp(k, root->k) == 0) {**

**cout << "\nEnter New Meaning of Keyword " << root->k << ": ";**

**cin >> root->m;**

**return 1;**

**}**

**if (strcmp(k, root->k) < 0) root = root->left;**

**else root = root->right;**

**}**

**return -1;**

**}**

**node \*dict::del(node \*root, char k[20]) {**

**if (root == NULL) {**

**cout << "\nElement Not Found";**

**return root;**

**}**

**if (strcmp(k, root->k) < 0) {**

**root->left = del(root->left, k);**

**return root;**

**}**

**if (strcmp(k, root->k) > 0) {**

**root->right = del(root->right, k);**

**return root;**

**}**

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

**delete root;**

**return NULL;**

**}**

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

**node \*temp = root;**

**root = root->left;**

**delete temp;**

**return root;**

**}**

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

**node \*temp = root;**

**root = root->right;**

**delete temp;**

**return root;**

**}**

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

**strcpy(root->k, temp->k);**

**root->right = del(root->right, temp->k);**

**return root;**

**}**

**node \*dict::min(node \*q) {**

**while (q->left != NULL) {**

**q = q->left;**

**}**

**return q;**

**}**

**int main() {**

**int ch;**

**dict d;**

**d.root = NULL;**

**do {**

**cout << "\nMenu\n1.Create\n2.Display\n3.Search\n4.Update\n5.Delete\nEnter your choice: ";**

**cin >> ch;**

**switch (ch) {**

**case 1:**

**d.create();**

**break;**

**case 2:**

**if (d.root == NULL) {**

**cout << "\nNo keywords added yet.";**

**} else {**

**d.disp(d.root);**

**}**

**break;**

**case 3: {**

**if (d.root == NULL) {**

**cout << "\nDictionary is Empty. First add keywords then try again.";**

**break;**

**}**

**char k[20];**

**cout << "\nEnter Keyword to search: ";**

**cin >> k;**

**if (d.search(d.root, k) == 1) {**

**cout << "\nKeyword Found.";**

**} else {**

**cout << "\nKeyword Not Found.";**

**}**

**break;**

**}**

**case 4: {**

**if (d.root == NULL) {**

**cout << "\nDictionary is Empty. First add keywords then try again.";**

**break;**

**}**

**char k[20];**

**cout << "\nEnter Keyword to update: ";**

**cin >> k;**

**if (d.update(d.root, k) == 1) {**

**cout << "\nMeaning Updated.";**

**} else {**

**cout << "\nMeaning Not Found.";**

**}**

**break;**

**}**

**case 5: {**

**if (d.root == NULL) {**

**cout << "\nDictionary is Empty. First add keywords then try again.";**

**break;**

**}**

**char k[20];**

**cout << "\nEnter Keyword to delete: ";**

**cin >> k;**

**d.root = d.del(d.root, k);**

**break;**

**}**

**}**

**} while (ch <= 5);**

**return 0;**

**}**