**#include<iostream>**

**#include<string.h>**

**using namespace std;**

**class dict {**

**char word[20], mean[50];**

**dict \*left, \*right;**

**int ht;**

**public:**

**dict\* create(dict \*root);**

**dict\* insert(dict \*root, char word[], char mean[]);**

**void display(dict \*);**

**int height(dict \*);**

**dict\* rotateright(dict \*);**

**dict\* rotateleft(dict \*);**

**int BF(dict \*);**

**dict\* deletion(dict \*, char \*);**

**dict\* RR(dict\*);**

**dict\* LL(dict\*);**

**dict\* RL(dict\*);**

**dict\* LR(dict\*);**

**};**

**dict\* dict::create(dict \*root) {**

**int n;**

**char w[20], m[50];**

**cout << "\nEnter total number of words:";**

**cin >> n;**

**for (int i = 0; i < n; i++) {**

**cout << "\nEnter word " << i + 1 << " : ";**

**cin >> w;**

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

**cin >> m;**

**root = insert(root, w, m);**

**}**

**return root;**

**}**

**dict\* dict::insert(dict \*root, char w[], char m[]) {**

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

**root = new dict;**

**strcpy(root->word, w);**

**strcpy(root->mean, m);**

**root->left = NULL;**

**root->right = NULL;**

**return root;**

**} else {**

**if (strcmp(w, root->word) > 0) {**

**root->right = insert(root->right, w, m);**

**// Handling BF...**

**} else {**

**root->left = insert(root->left, w, m);**

**// Handling BF...**

**}**

**}**

**// Update height and perform rotations...**

**return root;**

**}**

**void dict::display(dict \*root) {**

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

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

**cout << "\nNode is: " << root->word << "-" << root->mean;**

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

**}**

**}**

**int dict::height(dict \*root) {**

**if (root == NULL) return 0;**

**int lh = (root->left == NULL) ? 0 : 1 + root->left->ht;**

**int rh = (root->right == NULL) ? 0 : 1 + root->right->ht;**

**return max(lh, rh);**

**}**

**dict\* dict::deletion(dict \*T, char \*w) {**

**// Deletion logic...**

**return T;**

**}**

**dict\* dict::rotateright(dict \*x) {**

**// Rotation logic...**

**return NULL;**

**}**

**dict\* dict::rotateleft(dict \*x) {**

**// Rotation logic...**

**return NULL;**

**}**

**int dict::BF(dict \*root) {**

**// Balance Factor calculation...**

**return 0;**

**}**

**int main() {**

**int ch;**

**dict d, \*root = NULL;**

**char w[20], m[50];**

**cout << "\n\*\*\* Dictionary : codyapa \*\*\*";**

**do {**

**cout << "\n\nMENU:";**

**cout << "\n1. Create \n2. Insert \n3. Delete\n4. Display \n5. Exit";**

**cout << "\nEnter your choice:";**

**cin >> ch;**

**switch(ch) {**

**case 1: root = d.create(root); break;**

**case 2:**

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

**cin >> w;**

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

**cin >> m;**

**root = d.insert(root, w, m);**

**break;**

**case 3:**

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

**cin >> w;**

**root = d.deletion(root, w);**

**break;**

**case 4: d.display(root); break;**

**case 5: break;**

**default: cout << "\nInvalid choice!";**

**}**

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

**return 0;**

**}**