```cpp
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
// linked_list_game.cpp
// A game using my linked list implementation in C++.
// Author: Lauren E. Scott
// June 27, 2014
//
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * */


#include <iostream>
#include <stdbool.h>
#include <set>
#include "linkedlistcpp.h"
using namespace std;

/* * * * * * * * * * * * * * * * * * * * * * * * * */
// Function: initialize_riddles
// A convenient place to create all riddles used in each of the 10 levels.
/* * * * * * * * * * * * * * * * * * * * * * * * * */

void initialize_riddles(string riddle[10]) {
    riddle[0] = "I have wings, feathers, and I can fly. What am I?\n";
    riddle[1] = "I am something that you sit on every day, and I have a pret
    riddle[2] = "I am on trees, bushes, flowers, and I help them soak up the
    riddle[3] = "There is nothing more heroic than carrying me and a shield
    riddle[4] = "I cover lots of animals on this earth, and I am soft to the
    riddle[5] = "You don't want to be me, but I often make you cough and sne
    riddle[6] = "I give life to most beings on Earth, and I can be both free
    riddle[7] = "I am what makes up the game that you play right now.";
    riddle[8] = "Some say I am the tastiest food on the planet. What am I?";
    riddle[9] = "You have reachd me. What am I?";
}

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
// Function: initialize_answers
// A convenient place to create all answers used in each of the 10 levels.
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * */

void initialize_answers(string answer[10]) {
    answer[0] = "BIRD";
    answer[1] = "TOILET";
    answer[2] = "LEAF";
    answer[3] = "SWORD";
    answer[4] = "FUR";
    answer[5] = "SICK";
```

```
46      answer[6] = "WATER";
47      answer[7] = "CODE";
48      answer[8] = "BACON";
49      answer[9] = "END";
50  }
51
52  /* * * * * * * * * * * * * * * * * * * * * * */
53  // Function: initialize_chars
54  // A convenient place to define the sets of characters used in each of the 1
55  /* * * * * * * * * * * * * * * * * * * * * * */
56
57  void initialize_chars(string allowable_chars[10]) {
58      allowable_chars[0] = "XOINBRSAD";
59      allowable_chars[1] = "MWROIDTALE";
60      allowable_chars[2] = "LTWDRAOIEGF";
61      allowable_chars[3] = "XWTSQAIRDOL";
62      allowable_chars[4] = "ERAFLU";
63      allowable_chars[5] = "MSRICWKA";
64      allowable_chars[6] = "DKRGWPATE";
65      allowable_chars[7] = "GUOADEC";
66      allowable_chars[8] = "AKRGDBWECNO";
67      allowable_chars[9] = "IRGCDNRE";
68  }
69
70  /* * * * * * * * * * * * * * * * * * * * * * */
71  // Function: win_check
72  // Checks to see if the list contents match the answer (which is in string f
73  /* * * * * * * * * * * * * * * * * * * * * * */
74
75  bool win_check(LList<char> l, string answer) {
76      Node<char>* checker = l.getHead();          // We will use a node pointe
77      int i = 0;
78
79      while(checker != 0) {                       // Loop checks list against
80          if(checker->getData() != answer[i])
81              return false;
82  //      cout << checker->getData() << " = " << answer[i] << " ? " << endl;
83          i++;
84          checker = checker->getNext();
85      }
86      if(i == answer.length())
87          return true;
88      return false;
89  }
90
```

```
/* * * * * * * * * * * * * * * * * * * * * * */
// Function: serve_level
// The heavy lifting for the game. Takes in the player's input with regard t
// how they want to manipulate the list, and updates the list accordingly. A
// checks for a win every time the list is manipulated.
/* * * * * * * * * * * * * * * * * * * * * * */

int serve_level(string riddle, string answer, string allowable_chars) {
    bool correct_answer = false;
    int player_input_1;
    char player_input_2;
    LList<char> l;

    cout << riddle << endl;
    while(!correct_answer) {
        player_input_1 = 0;
        cout << "Available letters: " << allowable_chars << endl;
        cout << "Choose something to do to the list: " << endl;
        cout << "[1] to Append a letter to the back of the list. " << endl;
        cout << "[2] to Push a letter to the front of the list. " << endl;
        cout << "[3] to Delete a letter from the list. " << endl;
        cout << "The list will be printed out each time you modify it. " <<e
        cin >> player_input_1;
        if(player_input_1 == 1) {
            cout << "Select a letter to append. " << endl;
            cin >> player_input_2;
            if(allowable_chars.find(player_input_2) != string::npos) {
                l.append(player_input_2);
                cout << player_input_2 << " appended to the list! " << endl;
                l.print();
            } else {
                cout << "Please use a valid letter! ";
            }
        } else if(player_input_1 == 2) {
            cout << "Select a letter to push to the front of the list. " <<
            cin >> player_input_2;
            if(allowable_chars.find(player_input_2) != string::npos) {
                l.push_front(player_input_2);
                cout << player_input_2 << " pushed to the front of list! " <
                l.print();
            } else {
                cout << "Please use a valid letter! ";
            }
        } else if(player_input_1 == 3) {
            cout << "Select a letter to delete. " << endl;
```

```
136             cin >> player_input_2;
137             if(allowable_chars.find(player_input_2) != string::npos) {
138                 l.del(player_input_2);
139                 cout << player_input_2 << " deleted from the list! " << endl
140                 l.print();
141             } else {
142                 cout << "Please use a valid letter! ";
143             }
144         } else {
145             cout << "Please select a valid option (1, 2, or 3). " << endl;
146         }
147         if(win_check(l, answer) == true) {
148             cout << "Nice job! The answer is " << answer << " ! " << endl;
149             correct_answer = true;
150         }
151         cin.clear();
152         cin.ignore(10000, '\n');
153     }
154     return 0;
155
156 }
157
158 /* * * * * * * * * * * * * * * * * * * * * * * */
159 // Function: main
160 // Starting point of the game, main method.
161 /* * * * * * * * * * * * * * * * * * * * * * * */
162
163 int main() {
164     string riddle[10], answer[10];
165     string allowable_chars[10];
166     initialize_riddles(riddle);
167     initialize_answers(answer);
168     initialize_chars(allowable_chars);
169     int curr_level = 0;
170
171     cout << "------------ Welcome to the Linked List Game! ----------------
172     cout << " Your goal is to answer riddles by using the properties of link
173     cout << " Press Ctrl+C at any time to quit. " << endl;
174     cout << " Press 1 to append to the list. " << endl;
175     cout << " Press 2 to push to the front list. " << endl;
176     cout << " Press 3 to delete from the list. " << endl;
177     cout << " Here is your first riddle:" << endl;
178     for ( curr_level = 0; curr_level <= 10; curr_level++) {
179         cout << "------- Level " << curr_level+1 << " ------- " << endl;
180         serve_level(riddle[curr_level], answer[curr_level], allowable_chars[
```

```
181     }
182     cout << "Nice job! You've beaten the Linked List Game! :) ";
183
184 }
185
186
```