```
Script started on 2023-11-10 21:03:28+00:00 [TERM="xterm-256color" TTY="/dev/pts/0" COLUM
NS="69" LINES="100"]
\[\033[01;34m\]\w\[\033[00m\]$ pwd
/home/runner/Lab-17-Linked-List-Find-and-Remove-kcp3s
\[\033[01;34m\]\w\[\033[00m\]$ ls -la
total 2512
drwxr-xr-x 1 runner runner     284 Nov 10 21:03 .
drwxrwxrwx 1 runner runner     132 Nov 10 19:31 ..
-rwxr-xr-x 1 runner runner   17904 Nov 10 21:02 a.out
-rw-r--r-- 1 runner runner      17 Oct 27 20:51 .breakpoints
drwxr-xr-x 1 runner runner      12 Jan 24  2022 .cache
drwxr-x--- 1 runner runner     494 Nov  7 18:53 .ccls-cache
drwxr-xr-x 1 runner runner      68 Nov 10 18:43 .lesson
-rw-r--r-- 1 runner runner    1845 Nov 10 21:01 LinkedList.cpp
-rw-r--r-- 1 runner runner     952 Nov  7 14:01 LinkedList.h
-rwxr-xr-x 1 runner runner 1254392 Oct 27 20:53 main
-rw-r--r-- 1 runner runner    1011 Nov 10 18:43 main.cpp
-rwxr-xr-x 1 runner runner 1255712 Oct 27 20:53 main-debug
-rw-r--r-- 1 runner runner     449 Oct 27 20:53 Makefile
-rw-r--r-- 1 runner runner       0 Nov 10 21:03 Patel_Lab_17.log
-rw-r--r-- 1 runner runner    1426 Dec 21  2022 .replit
-rw-r--r-- 1 runner runner     141 Nov  7 14:25 replit.nix
-rw-r--r-- 1 runner runner      38 Nov  7 14:27 somedata.txt
\[\033[01;34m\]\w\[\033[00m\]$ cat -n main.cpp
     1  #include "LinkedList.h"
     2  #include <fstream>
     3  #include <iostream>
     4
     5  int main() {
     6    std::string filename;
     7    std::cout << "Enter a data file: ";
     8    std::cin >> filename;
     9    std::ifstream datafile;
    10    datafile.open(filename);
    11    if (!datafile) {
    12      std::cout << "ERROR: " << filename << " could not open...\n";
    13      return 0;
    14    }
    15    std::cout << '\n';
    16    LinkedList values;
    17    int value;
    18    int count{0};
    19    values.print();
    20    datafile >> value;
    21    while (datafile) {
    22      values.push_back(value);
    23      count++;
    24      datafile >> value;
    25    }
    26    datafile.close();
    27    values.print();
    28    values.remove_value(504); // remove the last item
    29    values.print();
    30    values.remove_value(667); // remove the first item
    31    values.print();
    32    values.remove_value(68); // remove a middle item
    33    values.print();
    34    values.remove_value(3);   // remove a non-existant item
    35    values.remove_value(441); // remove the last item
    36    values.remove_value(891); // remove another middle item
    37    values.remove_value(67);  // remove another front item
    38    values.print();
    39  }\[\033[01;34m\]\w\[\033[00m\]$ cat -n LīnkedLīst.h
     1  #ifndef _LINKEDLIST_H
     2  #define _LINKEDLIST_H
     3
     4  struct Node {
     5    // some data variables and such
     6    int data;
     7    // Pointer to another Node
     8    Node *next;
     9  };
    10
    11  class LinkedList {
    12  private:
    13    Node *head; // pointer to first item in list
    14    Node *tail; // pointer to last item in list
    15    // TODO:
    16    // find’s a value in a list
    17    // returns a nullptr if it wasn’t found
    18    // returns a pointer to the item if it was found
    19    Node *find(int value);
    20
    21  public:
    22    LinkedList();               // How do I initialize my object?
    23    void push_back(int value);  // add elements to the end of our LL
    24    void print() const;         // print all elements
    25    bool empty() const;         // am i empty
    26    void push_front(int value); // push to front
    27    // TODO
    28    // given an integer value, this function should
    29    // remove the first instance of the value it finds
    30    // in the list, or print an error message if
    31    // it could not remove the value
    32    void remove_value(int value);
    33  };
    34
    35  #endif\[\033[01;34m\]\w\[\033[00m\]$ cat -n LinkedList.cpp
     1  #include "LinkedList.h"
     2  #include <iostream>
     3
     4  LinkedList::LinkedList() {
     5    head = nullptr;
     6    tail = nullptr;
     7  }
     8
     9  bool LinkedList::empty() const {
    10    if (head == nullptr)
    11      return true;
    12    return false;
    13  }
    14
    15  void LinkedList::print() const {
    16    if (empty()) {
    17      std::cout << "ERROR: empty list...\n";
    18      return;
    19    }
    20
    21    std::cout << "Data:\n";
    22    Node *curr = head;
    23    while (curr != nullptr) {
    24      std::cout << curr->data << ’\n’;
    25      curr = curr->next;
    26    }
    27    std::cout << ’\n’;
    28  }
    29
    30  void LinkedList::push_front(int value) {
    31    Node *newNode = new Node;
    32    newNode->data = value;
    33    newNode->next = head;
    34
    35    if (empty()) {
    36      head = newNode;
    37      tail = newNode;
    38      return;
    39    }
    40
    41    head = newNode;
    42  }
```

```

void LinkedList::push_back(int value) {
  Node *newNode = new Node;
  newNode->data = value;
  newNode->next = nullptr;

  if (empty()) {
    head = newNode;
    tail = newNode;
    return;
  }
  tail->next = newNode;
  tail = newNode;
}

void LinkedList::remove_value(int value) {
  // TODO: IMPLEMENT THIS FUNCTION
  Node *prev = find(value);
  if (prev == nullptr) {
    std::cout << "ERROR: " << value << " not found, could not remove...\n";
    return;
  }
  Node *curr = prev->next;


    if(prev == head){
         head = curr->next;
         delete curr;
         return;
     }

      prev->next = curr->next;

  if (curr == tail) {
    tail = prev;
  }
  delete curr;
}

Node *LinkedList::find(int value) {
  // TODO: find the value in the list,
  // return nullptr if it isn't found
  // otherwise return a pointer to the
  // value in the list
  Node *curr = head;
  Node *prev = nullptr;
  while (curr != nullptr) {
    if (curr->data == value) {
        //std::cout << curr->data;
        if(prev == nullptr){
            return nullptr;}

      return prev;
    }
    prev = curr;
    curr = curr->next;
  }
  return nullptr;
}
\[\033[01;34m\]\w\[\033[00m\]$ g++ main.cpp LinkedList.cpp -o lltest
\[\033[01;34m\]\w\[\033[00m\]$ ./lltest
Enter a data file: somedata.txt

ERROR: empty list...
Data:
667
67
248
68
891
```

```
778
228
162
441
504

Data:
667
67
248
68
891
778
228
162
441

ERROR: 667 not found, could not remove...
Data:
667
67
248
68
891
778
228
162
441

Data:
667
67
248
891
778
228
162
441

ERROR: 3 not found, could not remove...
Data:
248
778
228
162

\[\033[01;34m\]\w\[\033[00m\]$ .//test
sh: 8: .//test: not found
\[\033[01;34m\]\w\[\033[00m\]$ ./l̸ltest
Enter a data file: somedata.txt

ERROR: empty list...
Data:
667
67
248
68
891
778
228
162
441
504

Data:
667
67
248
68
891
```

```
778
228
162
441

ERROR: 667 not found, could not remove...
Data:
667
67
248
68
891
778
228
162
441

Data:
667
67
248
891
778
228
162
441

ERROR: 3 not found, could not remove...
Data:
248
778
228
162

\[\033[01;34m\]\w\[\033[00m\]$ ./lltest
Enter a data file: somedata.txt

ERROR: empty list...
Data:
667
67
248
68
891
778
228
162
441
504

Data:
667
67
248
68
891
778
228
162
441

ERROR: 667 not found, could not remove...
Data:
667
67
248
68
891
778
228
```

```
162
441

Data:
667
67
248
891
778
228
162
441

ERROR: 3 not found, could not remove...
Data:
248
778
228
162

\[\033[01;34m\]\w\[\033[00m\]$ ./lltest
Enter a data file: somedata.txt

ERROR: empty list...
Data:
667
67
248
68
891
778
228
162
441
504

Data:
667
67
248
68
891
778
228
162
441

ERROR: 667 not found, could not remove...
Data:
667
67
248
68
891
778
228
162
441

Data:
667
67
248
891
778
228
162
441

ERROR: 3 not found, could not remove...
```

```
Data:
248
778
228
162

\[\033[01;34m\]\w\[\033[00m\]$ exit

Script done on 2023-11-10 21:05:35+00:00 [COMMAND_EXIT_CODE="0"]
```