⊛⊛ wouldn't compile as-is → extra '}'
⇒ compiled ok on fixing →
⇒ gogo test → seg. fault

⊛⊛ not monospaced font

Shackelford_Jade

15/50

(partial credit)

```
#ifndef LLCP_INT_H
#define LLCP_INT_H

#include <iostream>

struct Node
{
  int data;
  Node *link;
};

int   FindListLength(Node* headPtr);
bool  IsSortedUp(Node* headPtr);
void  InsertAsHead(Node*& headPtr, int value);
void  InsertAsTail(Node*& headPtr, int value);
void  InsertSortedUp(Node*& headPtr, int value);
bool  DelFirstTargetNode(Node*& headPtr, int target);
bool  DelNodeBefore1stMatch(Node*& headPtr, int target);
void  ShowAll(std::ostream& outs, Node* headPtr);
void  FindMinMax(Node* headPtr, int& minValue, int& maxValue);
double FindAverage(Node* headPtr);
void  ListClear(Node*& headPtr, int noMsg = 0);
void  FormUniquePairs(Node* headPtr);

#endif

#include <iostream>
#include <cstdlib>
#include "llcpInt.h"
using namespace std;

// definition of FormUniquePairs of Assignment 5 Part 1
// (put at near top to facilitate printing and grading)

void FormUniquePairs(Node* headPtr) {
  Node *precursor = 0, *cursor = headPtr->link;
  bool unique = true;
  while (headPtr != 0) {
    while (cursor != 0) {
      if (headPtr->data == cursor->data) {
        if (unique) {
          precursor->link = cursor->link;
          cursor->link = headPtr->link;
          headPtr->link = cursor;
          headPtr = cursor;
          unique = false;
        }
```

```
          else {
          precursor->link = cursor->link;
          delete cursor;
          cursor = precursor;
          }
        }
        precursor = cursor;
        cursor = cursor->link;
      }
    }
    if (unique) {
      Node * newPtr = new Node;
      newPtr->data = headPtr->data;
      newPtr->link = headPtr->link;
      headPtr->link = newPtr;
      headPtr = headPtr->link;
    }
    headPtr = headPtr->link;
    cursor = headPtr->link;
    unique = true;
  }
}
```

compilation error