

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

⊗⊗⊗ not monospaced font
#ifndef LLCP_INT_H Shackelford_Jade
#define LLCP_INT_H

15
50

(partial credit)

```
#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& minVal, int& maxVal);  
double FindAverage(Node* headPtr);  
void ListClear(Node*& headPtr, int noMsg = 0);  
void FormUniquePairs(Node* headPtr);
```

```
#endif
```

```
#include <iostream>
```

```
#include <cstdlib>
```

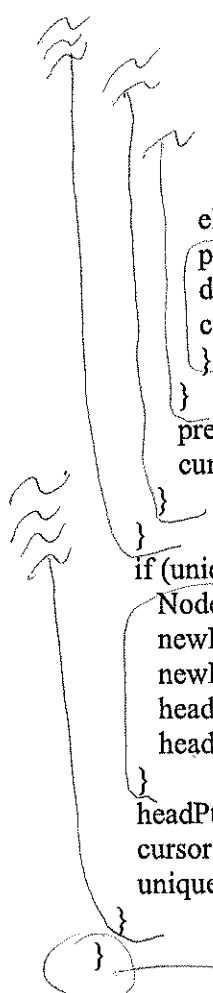
```
#include "llcpInt.h"
```

```
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;  
                }  
            }  
            cursor = cursor->link;  
        }  
        headPtr = headPtr->link;  
        precursor = headPtr;  
    }  
}
```



The diagram shows a vertical linked list structure. At the top, there are two wavy lines representing the start of the list. A vertical line descends from the first wavy line, with a horizontal branch pointing to the left towards the 'else' block. Another vertical line descends from the second wavy line, with a horizontal branch pointing to the left towards the 'if (unique)' block. The main vertical line continues down to a circle at the bottom, which contains a closing curly brace '}'. A horizontal line extends from this circle to the right, pointing towards the handwritten text 'compilation error'.

```
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