Nick McComb
CS 163
2/12/2013
Karla Fant

Assignment 2

1)

While this method may not be the most efficient, it would certainly work

This function is passed the head of the two linear linked lists to compare, and the head of a third list with a different datatype.

1 . Check if the lists have the same number of nodes using a recursive function

a. If they don't, they don't have the same contents

b. If they do, then navigate to the end of the first LLL recursively, adding the contents to the third list while doing it, and also a valid boolean

c. After the first list is copied,

i. Traverse to the end of the second list, and with every recursive traversal, recursively traverse through the third list and compare the current value in the second list to valid entries in the second one.

1. If a valid entry is found, then set it to be invalid, and continue checking the next node in the 2nd list recursively

2. If a valid entry is not found, then the lists are not identical, return 0 and exit the function

i. If the whole second list is traversed and all comparisons have found valid opposites, then the lists are identical. Return an 'identical' notification

Loop Invariant
Something that is true for every execution of a loop.

Example:

```
int j = 9;
for (int i = 0; i < 10; i++)
  j--
```

For every execution of the above loop, j + i == 9.

Unit Testing
This is the testing of individual units of a program separately. This can ensure a much better

combination, because if the individual aspects are ensured to work, then it is much easier to implement them together.

Default Arguments
Arguments that are passed to a function by default if no other arguments are passed

```
int number(int first = 5);

int number(int first)
{
  return first;
}

cout << number() << " " << number(10) << endl;  //This would output "5 10"
```

Function Overloading
This is when a function has multiple definitions, with different arguments. The program then decides which version of the function to use based on what type of data is passed to it when it is called

Example:

```
int number(int first)
{
  return first;
}

int number(int first, int second)
{
  return second + first;
}
```

Pass by Pointer vs. Pass by Reference
When you pass by pointer, you are passing an address to a piece of data, whereas when you are passing an item of data by reference, you pass an alias to the variable and it can still be accessed as a normal variable, whereas something passed by pointer cannot.

Example
You have to dereference the variable y before you can access it, while x can be accessed and modified just like a local variable.

```
int add_x_to_y(int & x, int * y)
{
  x = x + *y
}
```

Pointer Arithmetic
This is when pointers' values are added or subtracted the size of the data they are pointing to.

Example:

```
char array[6];

//Read in information...

char * pointer = array;
++pointer;
cout << pointer << endl;  //This would output the second character that is stored in the array
```

Data Abstraction
Data Abstraction is when certain aspects of implementation are hidden under layers of 'abstraction' that can further help understanding and debugging of a program of alrogithm.

Example:

instead of individually executing the following actions

```
int i = 5;
int j = 10;
int p = 40;
int x = ((i+4) / j) + 34 * (p/i) + 30;
```

You can hide the math under another layer of abstraction which promotes readability

```
int preformOperation(int i, int j, int p)
{
  return ((i+4) / j) + 34 * (p/i) + 30;
}

int i = 5;
int j = 10;
int p = 40;
int x = performOperation(i,j,p);
```

3)

As the developer of an Abstract Data Type you have to be considerate of the end user’s experience. IT has to be kept in mind that the user will not have any experience with the inner workings of the ADT, so they would not know what to do with most error messages that would come out of the inner workings of an ADT. This would be ethically wrong, because the user would not have any knowledge of the errors that were produced. The job of interpreting those errors is that of the author of the client program. It is this programmer that will know what to do with these errors, and therefore is ethically right for this person to handle and display relevant errors to the user.