**Test 2**

Using the diagram presented in the section called “Diagram 1” on page 2. You are to first create the 4 directional linked list to appear as illustrated in the diagram. Use the same names for each node as shown in the diagram and the directions between the nodes are two way. You can directly input these nodes and their directions in the main.

Your program should then ask the user for a starting node and an ending node. They are able to choose any starting and ending locations and your program should be able to begin from that starting node (by placing a pointer to that node) and going through each of the 4 directions until it finds the ending node. At this point, it should output the path it took. An example of the ending result is shown below.

This test will be covering linked lists as well as recursion. Because of this, you are to solve the problem using one recursive function. This function’s base case is if the current node it is located in is equal to the node the user typed in as the ending node. If it is, you are to output the name of that node and terminate the current node (this base case gives us the path in reverse, which is what we want).

**Assuming the testing goes up, left, down, right**

**OUTPUT:**

**Please enter the starting node: B**

**Please enter the destination node: I**

**Going left to A**

**Going down to D**

**D ends**

**A ends**

**Going down to E**

**Going down to H**

**Going left to G**

**G ends**

**H ends**

**Going right to F**

**Going up to C**

**C ends**

**Going down to I**

**Destination Reached, path was: I F E B**

Be aware that if the testing goes up, right, down, left, it would give a different answer. Your program will allow ANY start node and ANY destination node… even if they type in the same start and destination.

**Diagram 1**