//Queue <Vertex<String>> tempQueue = new ListQueue<Vertex<String>>();  
360 Stack <Vertex<String>> tempStack = new ListStack<Vertex<String>>();  
361 int j = 0;  
362   
363 System.out.printf("\n\nThe route with the fewest number of connections from %s to %s is: \n", myCity1, myCity2);  
364   
365 while(containsFalse() == true)  
366 {  
367 ArrayList<Vertex<String>> neighbors = ( ArrayList<Vertex<String>> )  
368 graph.getNeighbors( temp );  
369   
370 stack.push(temp);   
371   
372 //this is never executing... dont know why  
373 //String v = temp.toString();  
374 if(temp.equals(myCity2))  
375 {  
376 System.out.println("This should execute second");  
377 System.out.print(temp);  
378 stack.push(myCity2);  
379 if(stack.size() < tempStack.size() || tempStack.isEmpty())  
380 {  
381 while(stack.isEmpty() == false)  
382 {  
383 tempStack.push(stack.pop());  
384 }  
385 int connections = tempStack.size()-1;  
386 }  
387 }  
388 //now this is being skipped... why?  
389 else if(visited[cityIndex(temp)] == false)  
390 {  
391 visited[cityIndex(temp)] = true;  
392 temp = neighbors.get(j);  
393 System.out.print(temp);  
394 System.out.println("This should execute first");  
395 /\*  
396 if(j < neighbors.size())  
397 {  
398 temp = neighbors.get(j);  
399 j++;  
400 System.out.println("This should execute second");  
401 }  
402 else  
403 {  
404 for(int i = 0; i < visited.length; i++)  
405 {  
406 visited[i] = true;  
407 }  
408 System.out.println("This shouldn't execute");  
409 }  
410 \*/  
411 }//end if statement  
412 else  
413 {  
414 j++;  
415 System.out.print(temp);  
416 if(j == neighbors.size())  
417 {  
418 j = 0;  
419 stack.pop();  
420 }  
421 System.out.println("is this executing?");  
422 }  
423 }//end while statement  
424 resetVisited();  
425 int connections = tempStack.size()-1;  
426   
427 while(stack.size() > 0)  
428 {  
429 stack.pop();  
430 }  
431 while(tempStack.size() > 0)  
432 {  
433 temp = tempStack.pop();  
434 System.out.println(temp);  
435 System.out.printf("Connections: %d", connections);  
436 }