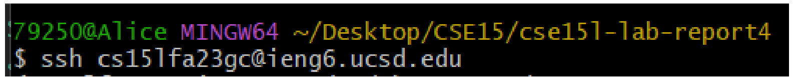
cse-15l-lab-report4

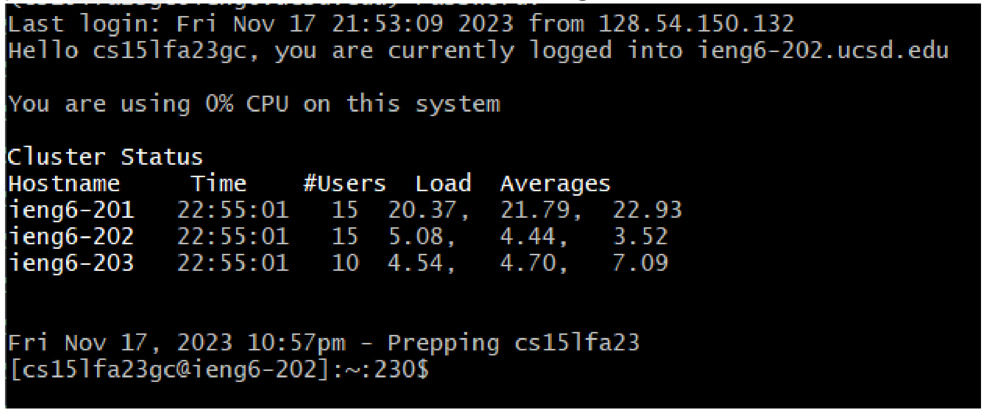
CSE 15L lab report 4

Log into ieng6

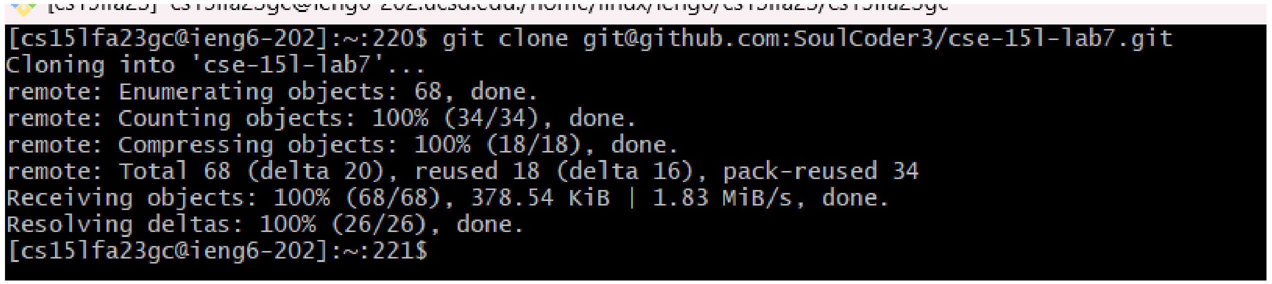

```
79250@Alice MINGW64 ~/Desktop/CSE15/cse15l-lab-report4
$ ssh cs15lfa23gc@ieng6.ucsd.edu
```

```
// Key Pressed:
ssh<space>cs15lfa23gc@ieng6.ucsd.edu<enter>
```

- `ssh` : provides a secure encrypted connection between two hosts over an insecure network.
- `<space>` : Enter a space.
- `cs15lfa23gc@ieng6.ucsd.edu` : my ieng6 server address.
- press `<enter>` to run the command and connect ieng6 server.


```
Last login: Fri Nov 17 21:53:09 2023 from 128.54.150.132
Hello cs15lfa23gc, you are currently logged into ieng6-202.ucsd.edu

You are using 0% CPU on this system

Cluster Status
Hostname     Time    #Users  Load  Averages
ieng6-201   22:55:01   15   20.37,  21.79,  22.93
ieng6-202   22:55:01   15   5.08,   4.44,   3.52
ieng6-203   22:55:01   10   4.54,   4.70,   7.09


Fri Nov 17, 2023 10:57pm - Prepping cs15lfa23
[cs15lfa23gc@ieng6-202]:~:230$
```

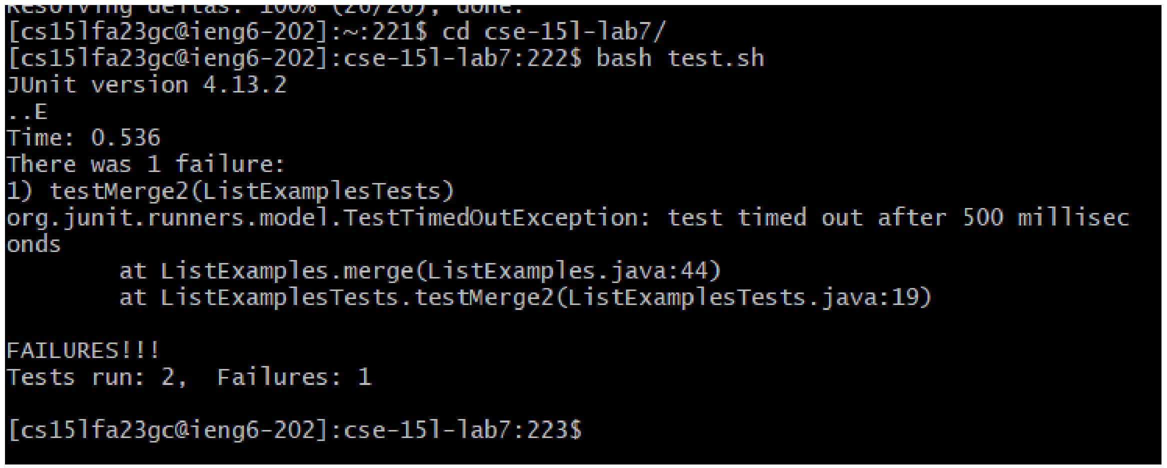
Clone your fork of the repository from your Github account (using the SSH URL)


```
[cs15lfa23gc@ieng6-202]:~:220$ git clone git@github.com:SoulCoder3/cse-15l-lab7.git
Cloning into 'cse-15l-lab7'...
remote: Enumerating objects: 68, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 68 (delta 20), reused 18 (delta 16), pack-reused 34
Receiving objects: 100% (68/68), 378.54 KiB | 1.83 MiB/s, done.
Resolving deltas: 100% (26/26), done.
[cs15lfa23gc@ieng6-202]:~:221$
```

```
// Key Pressed:
git<space>clone<space><cltrl+V><enter>
```

- `git clone` : make a copy in a local directory from remote directory.
- `<cltrl+V>` : paste the SSH key of the clipoard into the current cursor location, where the SSH key already copy from the github page.
- press `<enter>` to run the command and connect ieng6 server.

Run the tests, demonstrating that they fail

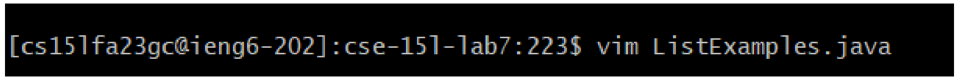

```
[cs15lfa23gc@ieng6-202]:~:221$ cd cse-15l-lab7/
[cs15lfa23gc@ieng6-202]:cse-15l-lab7:222$ bash test.sh
JUnit version 4.13.2
..E
Time: 0.536
There was 1 failure:
1) testMerge2(ListExamplesTests)
org.junit.runners.model.TestTimedOutException: test timed out after 500 millisec
onds
        at ListExamples.merge(ListExamples.java:44)
        at ListExamplesTests.testMerge2(ListExamplesTests.java:19)

FAILURES!!!
Tests run: 2,  Failures: 1

[cs15lfa23gc@ieng6-202]:cse-15l-lab7:223$
```

```
// key pressed:
1. cd<space>cse<tab><enter>
2. bash<space>tes<tab>.sh<enter>
```

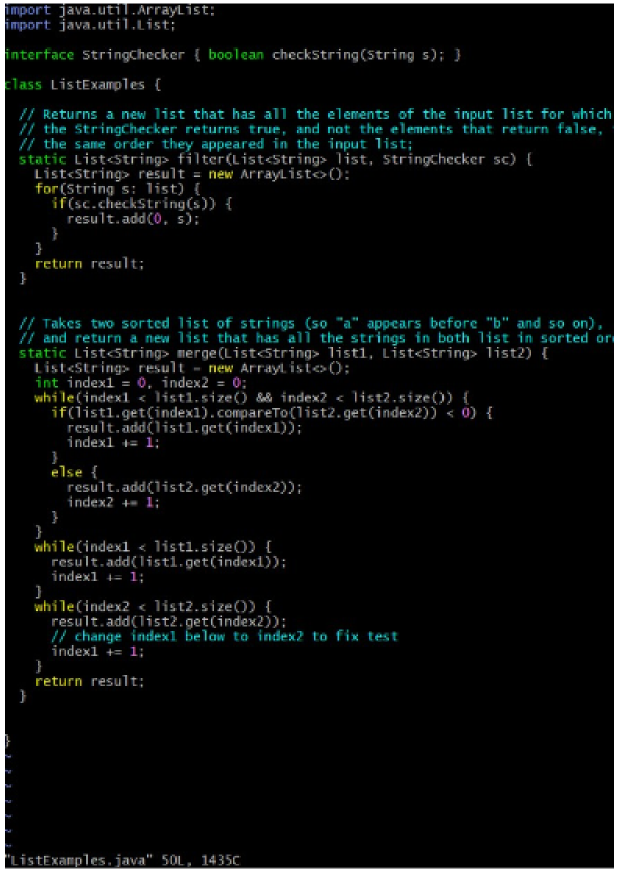
- `cd` : allows user to change the current directory.

- `<tab>` : command names, parameter names, argument values and file paths can all be autofilled by pressing the Tab key, if the they exist.
- type `cd` and then type `cse<tab>` . `<tab>` autofill the folder name `cse-15l-lab/` .
- press `<enter>` to change the current work directory to `cse-15l-lab/` .
- `<bash>` : run a command processor.
- type `bash` and then type `tes<tab>` . `<tab>` autofill the bash file name `test` .
- press `<enter>` to run the test.

Edit the code file to fix the failing test



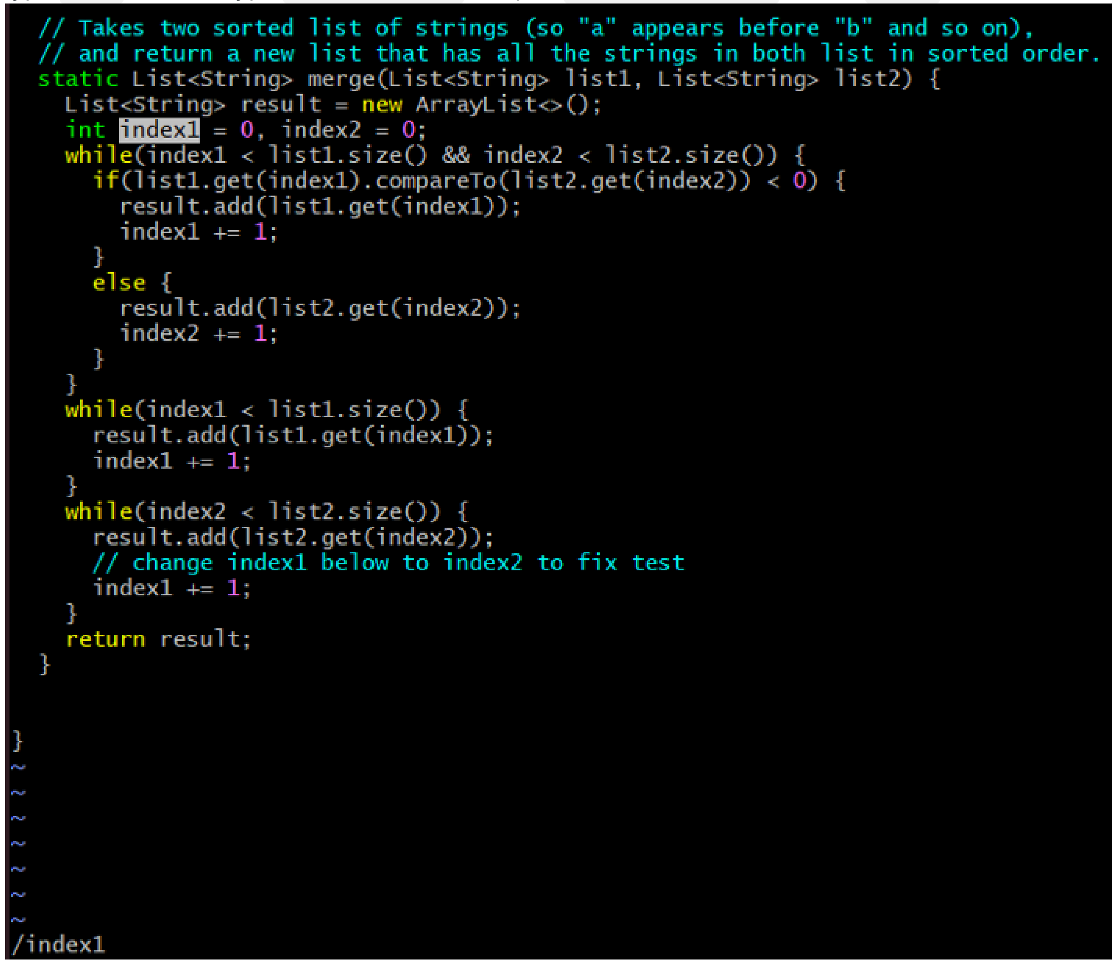
```
[cs15lfa23gc@ieng6-202]:cse-15l-lab7:223$ vim ListExamples.java
```



```
import java.util.ArrayList;
import java.util.List;

interface StringChecker { boolean checkString(String s); }

class ListExamples {

  // Returns a new list that has all the elements of the input list for which
  // the StringChecker returns true, and not the elements that return false,
  // the same order they appeared in the input list;
  static List<String> filter(List<String> list, StringChecker sc) {
    List<String> result = new ArrayList<>();
    for(String s: list) {
      if(sc.checkString(s)) {
        result.add(0, s);
      }
    }
    return result;
  }


  // Takes two sorted list of strings (so "a" appears before "b" and so on),
  // and return a new list that has all the strings in both list in sorted or
  static List<String> merge(List<String> list1, List<String> list2) {
    List<String> result = new ArrayList<>();
    int index1 = 0, index2 = 0;
    while(index1 < list1.size() && index2 < list2.size()) {
      if(list1.get(index1).compareTo(list2.get(index2)) < 0) {
        result.add(list1.get(index1));
        index1 += 1;
      }
      else {
        result.add(list2.get(index2));
        index2 += 1;
      }
    }
    while(index1 < list1.size()) {
      result.add(list1.get(index1));
      index1 += 1;
    }
    while(index2 < list2.size()) {
      result.add(list2.get(index2));
      // change index1 below to index2 to fix test
      index1 += 1;
    }
    return result;
  }


}
~
~
~
~
~
~
~
"ListExamples.java" 50L, 1435C
```

```
// key pressed:
1. vim<space>List<tab>.java<enter>
2. /index1<enter><n><n><n><n><n><n><n><n><n><i><right><right><right><right>><right><right><backspace>2<esc>:wq<enter>
```

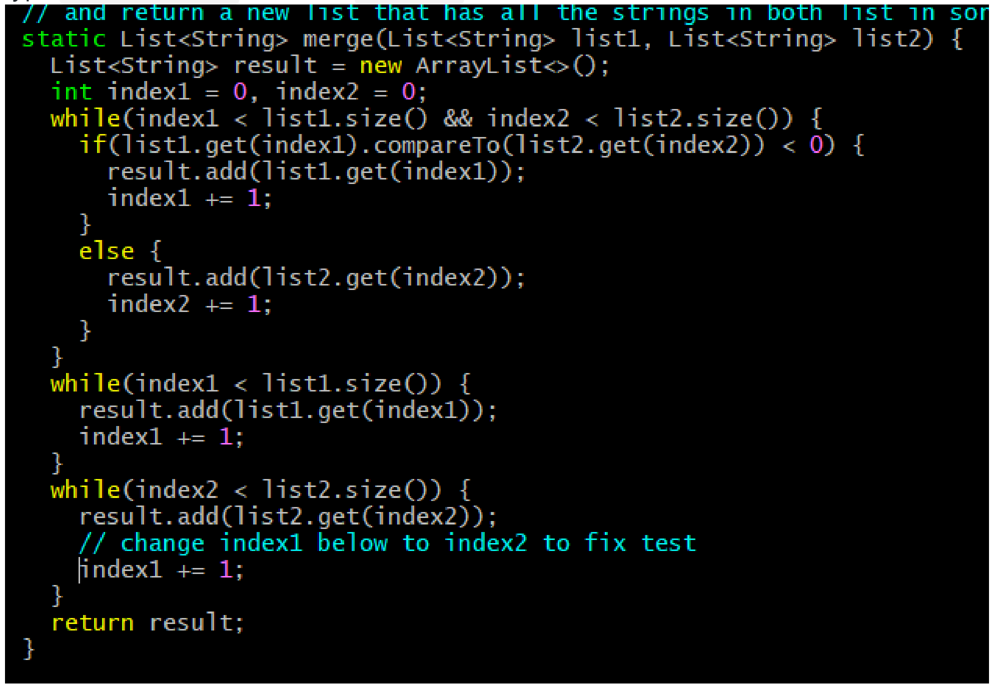
- `vim` : open document by vim editor.
- type `vim` and then type `List<tab>.java` to open `ListExamples.java` in vim. `<tab>` autofill ListExamples.



```
  // Takes two sorted list of strings (so "a" appears before "b" and so on),
  // and return a new list that has all the strings in both list in sorted order.
  static List<String> merge(List<String> list1, List<String> list2) {
    List<String> result = new ArrayList<>();
    int index1 = 0, index2 = 0;
    while(index1 < list1.size() && index2 < list2.size()) {
      if(list1.get(index1).compareTo(list2.get(index2)) < 0) {
        result.add(list1.get(index1));
        index1 += 1;
      }
      else {
        result.add(list2.get(index2));
        index2 += 1;
      }
    }
    while(index1 < list1.size()) {
      result.add(list1.get(index1));
      index1 += 1;
    }
    while(index2 < list2.size()) {
      result.add(list2.get(index2));
      // change index1 below to index2 to fix test
      index1 += 1;
    }
    return result;
  }


}
~
~
~
~
~
~
~
/index1
```

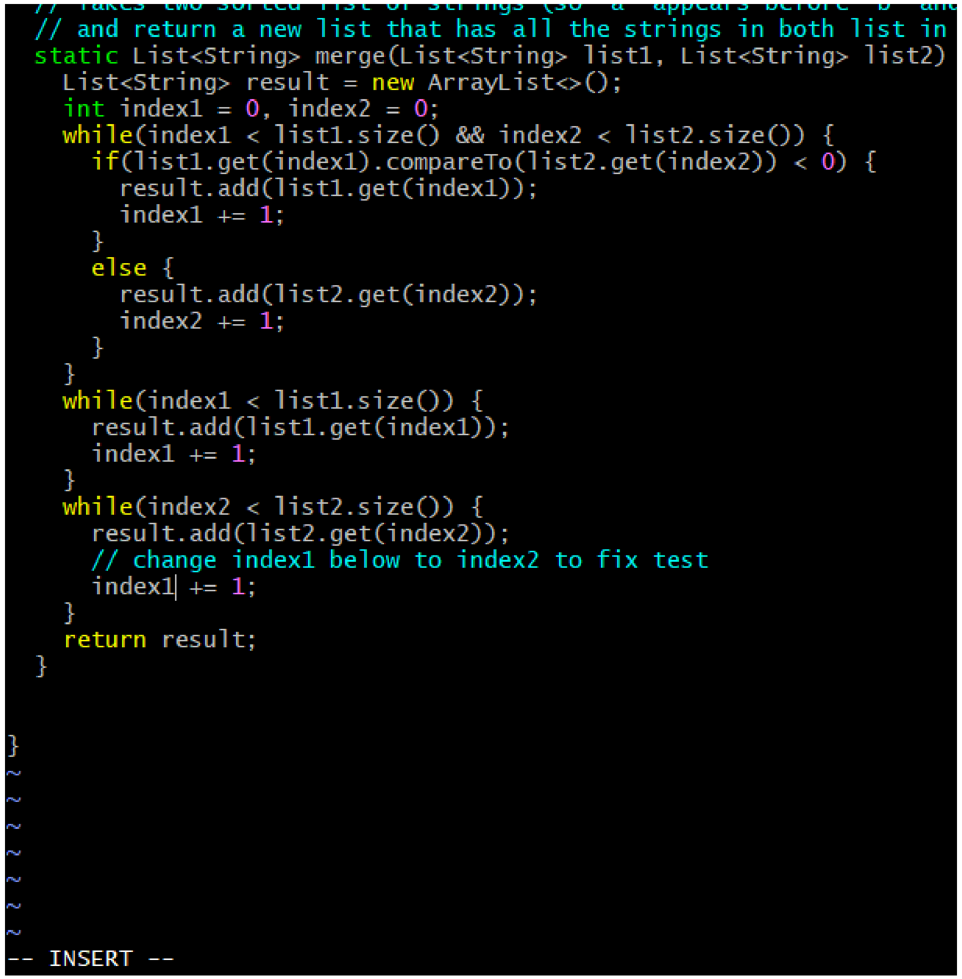
- `/text` : search a text in document in vim. The cursor will move to the first character of the first matching word.

- type `/index1<enter>` to search `"index1"` text in the file. It will move the cursor to the first `"i"` of the first matching word `"index1"`.



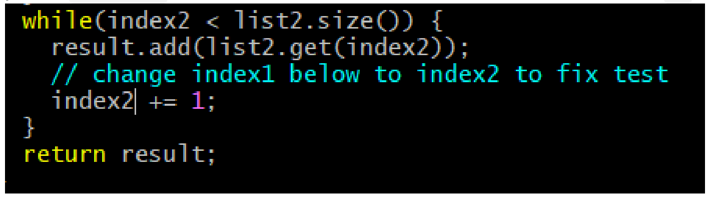
```
// and return a new list that has all the strings in both list in sor
static List<String> merge(List<String> list1, List<String> list2) {
  List<String> result = new ArrayList<>();
  int index1 = 0, index2 = 0;
  while(index1 < list1.size() && index2 < list2.size()) {
    if(list1.get(index1).compareTo(list2.get(index2)) < 0) {
      result.add(list1.get(index1));
      index1 += 1;
    }
    else {
      result.add(list2.get(index2));
      index2 += 1;
    }
  }
  while(index1 < list1.size()) {
    result.add(list1.get(index1));
    index1 += 1;
  }
  while(index2 < list2.size()) {
    result.add(list2.get(index2));
    // change index1 below to index2 to fix test
    index1 += 1;
  }
  return result;
}
```

- type `<n>` 9 times, `<n>` will move the cursor to the next matching word. so the cursor move to the `"index1"` we want to fix after 9 times.



```
 // Takes two sorted list of strings (so "a" appears before "b" and
 // and return a new list that has all the strings in both list in
 static List<String> merge(List<String> list1, List<String> list2)
  List<String> result = new ArrayList<>();
  int index1 = 0, index2 = 0;
  while(index1 < list1.size() && index2 < list2.size()) {
   if(list1.get(index1).compareTo(list2.get(index2)) < 0) {
    result.add(list1.get(index1));
    index1 += 1;
   }
   else {
    result.add(list2.get(index2));
    index2 += 1;
   }
  }
  while(index1 < list1.size()) {
   result.add(list1.get(index1));
   index1 += 1;
  }
  while(index2 < list2.size()) {
   result.add(list2.get(index2));
   // change index1 below to index2 to fix test
   index1 += 1;
  }
  return result;
 }


}
~
~
~
~
~
~
~
-- INSERT --
```

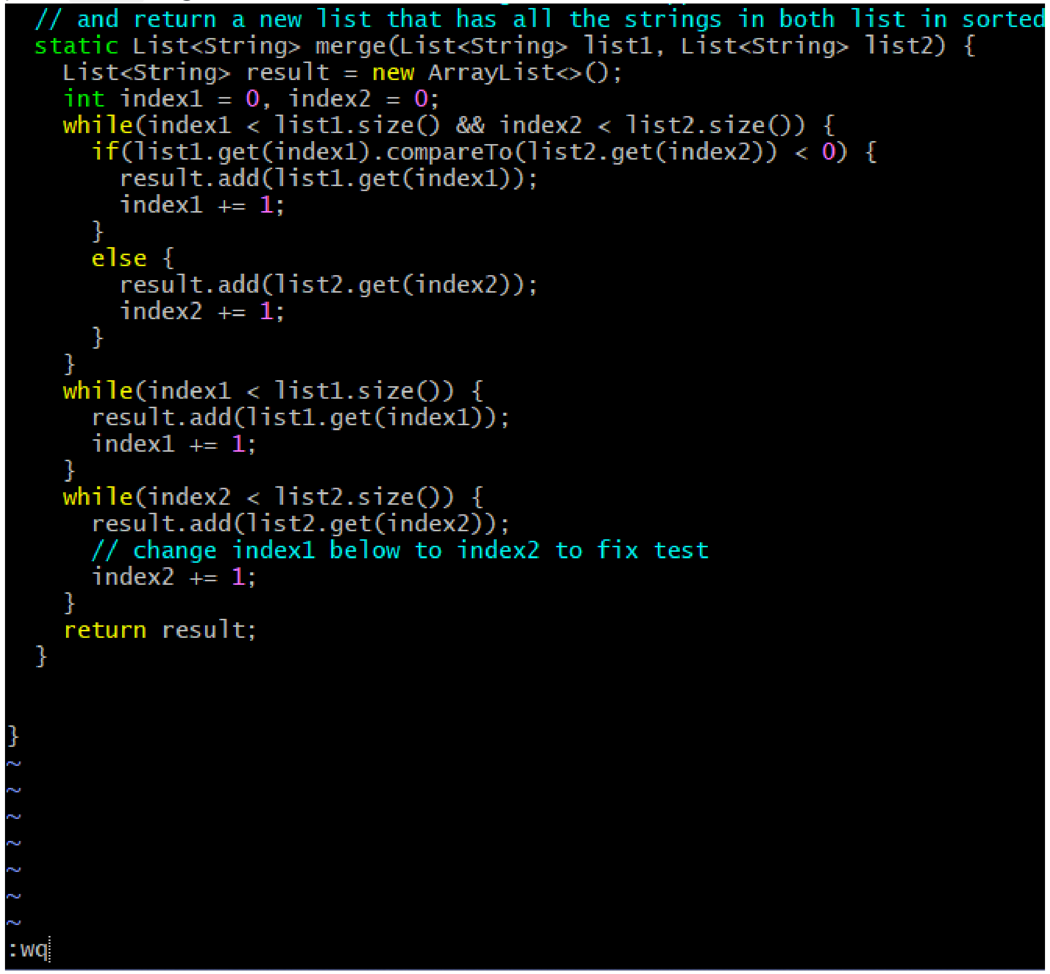
- type `<i>` change to insert mode, where we can insert and edit text.
- `<right>` : move the cursor one position to the right.
- press `<right>` 6 times to move the cursor on the character `1` .



```
while(index2 < list2.size()) {
 result.add(list2.get(index2));
 // change index1 below to index2 to fix test
 index2 += 1;
}
return result;
```

- press `<backspace>` to delete `1` and then type `2` .

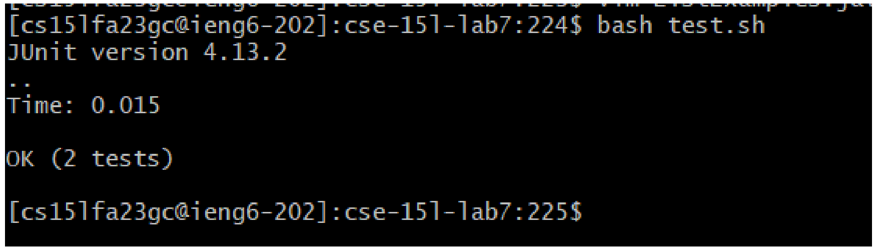
- press `<esc>` to go back normal mode.



```
  // and return a new list that has all the strings in both list in sorted
  static List<String> merge(List<String> list1, List<String> list2) {
    List<String> result = new ArrayList<>();
    int index1 = 0, index2 = 0;
    while(index1 < list1.size() && index2 < list2.size()) {
      if(list1.get(index1).compareTo(list2.get(index2)) < 0) {
        result.add(list1.get(index1));
        index1 += 1;
      }
      else {
        result.add(list2.get(index2));
        index2 += 1;
      }
    }
    while(index1 < list1.size()) {
      result.add(list1.get(index1));
      index1 += 1;
    }
    while(index2 < list2.size()) {
      result.add(list2.get(index2));
      // change index1 below to index2 to fix test
      index2 += 1;
    }
    return result;
  }


}
~
~
~
~
~
~
~
:wq
```

- type `:wq<enter>` to save and quit vim.

Run the tests, demonstrating that they now succeed


```
[cs15lfa23gc@ieng6-202]:cse-15l-lab7:224$ bash test.sh
JUnit version 4.13.2
..
Time: 0.015

OK (2 tests)

[cs15lfa23gc@ieng6-202]:cse-15l-lab7:225$
```

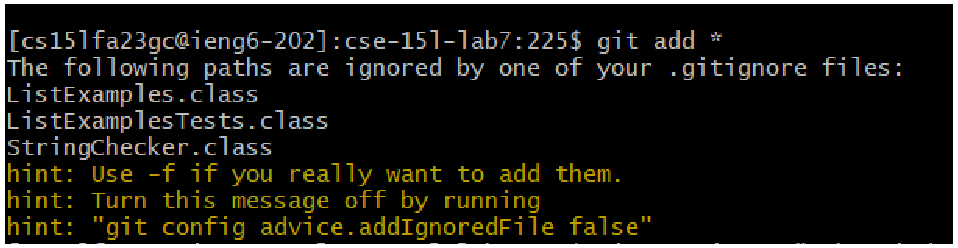
```
// key pressed:
<up><up><enter>
```

- type `<up><up>`: `bash test.sh` command was 2 up in the search history, so I used up arrow to access it.
- press `<enter>` to run the test and get success.

Commit and push the resulting change to your Github account (you can pick any commit message!)

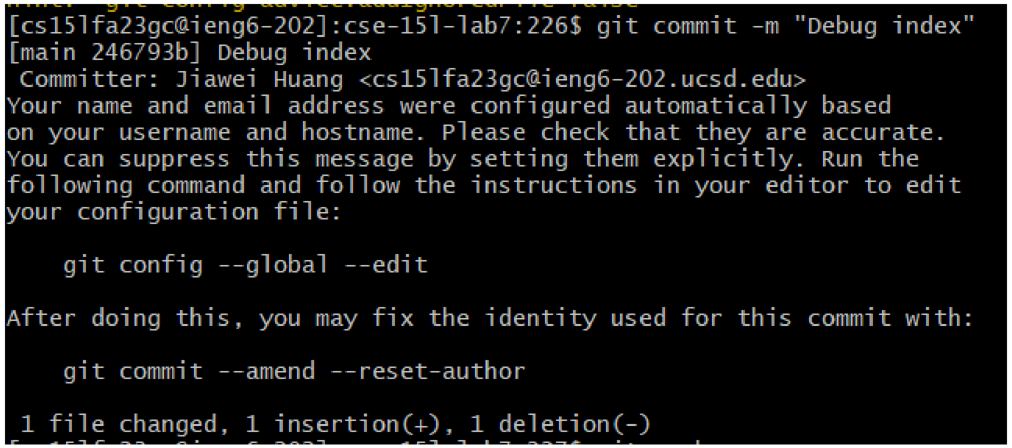
```
// key pressed:
1. git<space>add<space>*<enter>
2. git<space>commit<space><->m<space>"Debug<space>index"<enter>
3. git<space>push<enter>
```

git add


```
[cs15lfa23gc@ieng6-202]:cse-15l-lab7:225$ git add *
The following paths are ignored by one of your .gitignore files:
ListExamples.class
ListExamplesTests.class
StringChecker.class
hint: Use -f if you really want to add them.
hint: Turn this message off by running
hint: "git config advice.addIgnoredFile false"
```

- `git add *` to add all change in the working directory to the staging area.
- The files methioned in `.gitignore` file would be ignore when add.

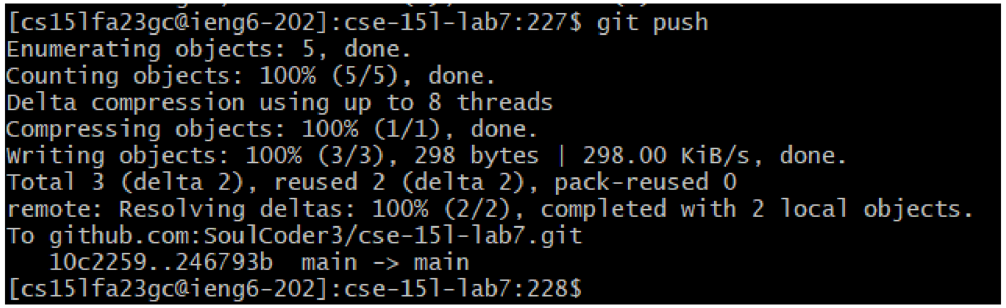
git commit



```
[cs15lfa23gc@ieng6-202]:cse-15l-lab7:226$ git commit -m "Debug index"
[main 246793b] Debug index
 Committer: Jiawei Huang <cs15lfa23gc@ieng6-202.ucsd.edu>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+), 1 deletion(-)
```

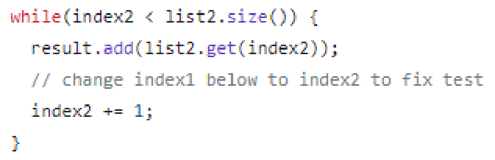
- `git commit -m "Debug index"` : submit changes in the staging area to the local repository, and add message `"Debug index"` .

git push



```
[cs15lfa23gc@ieng6-202]:cse-15l-lab7:227$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (1/1), done.
Writing objects: 100% (3/3), 298 bytes | 298.00 KiB/s, done.
Total 3 (delta 2), reused 2 (delta 2), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:SoulCoder3/cse-15l-lab7.git
   10c2259..246793b  main -> main
[cs15lfa23gc@ieng6-202]:cse-15l-lab7:228$
```

- `git push` : submit changes in local repository to remote repository.



```
while(index2 < list2.size()) {
  result.add(list2.get(index2));
  // change index1 below to index2 to fix test
  index2 += 1;
}
```

- double check the change in remote repository.