**Individual Project:  
Encode Utility -- Deliverable 3**

For the last deliverable of this project, your goals are to

* evaluate your current set of test cases on an alternative implementation of encode and
* possibly extend the test set in a black-box fashion to improve the coverage they achieve.

**Details**

In this deliverable, you will be provided with a precompiled, alternative version of the encode utility that was developed by one of your colleagues in parallel to yours. The goal of this task is to use this version of encode to evaluate the tests that you developed for the previous deliverables. To complete this task you must (1) run the tests you defined (class MyMainTest) against the provided version of encode and review the report on their results and (2) extend your set of tests to find bugs in this version of encode. There are 10 bugs in the provided encode utility, none of which are revealed by the test cases **we** provided. You must find **at least eight unique bugs** to get full points for this task, and finding the two remaining bugs will qualify you for up to 10 extra points. To remove ambiguity in identifying the bugs, **each valid bug will throw an SDPBugException with a message labeling it as a discovered bug** and a unique identifying number (i.e., if another test outputs the same identifying bug number, it is the same bug).

**Detailed instructions**

**Setup**

1. Download archive [individualProject-d3.tar.gz](https://drive.google.com/open?id=1As2v_VHpShPexTxNDYH4lLZOOuPb59E3)
2. Unpack the archive in the root directory of the **personal GitHub repo that we assigned to you**. After unpacking, you should see the following files under <root>/IndividualProject/D3 (we refer to this directory as <dir> hereafter):
   * **compileAndRunTests.bat** and **compileAndRunTests.sh**

These scripts for Windows and Unix/Mac, respectively, will compile and run against the provided version of encode the set of tests in <dir>/testsrc/edu/.../MyMainTest.java and save the console output for your review.

* + **lib/\***

These are various libraries used to run the tests. You can safely ignore these files.

* + **testsrc/edu/.../MyMainTest.java**

This is a placeholder that you will have to replace with your own version of this test class.

* + **testclasses/\***

This is the directory where the compiled version of your tests (i.e., class MyMainTest) will be saved. You can safely ignore this directory.

**Testing**

1. Before beginning this task you should make sure that the provided files work as expected, by doing the following:
   * Open a command shell
   * Go to directory <dir>
   * Run ./compileAndRunTests.sh  
     (if on Windows, run the corresponding bat file, compileAndRunTests.bat).
   * Check the content of file report.txt, which should be as

follows (time may vary):

JUnit version 4.12

..............................

Time: 0.263

OK (6 tests)

* + If you have not set up java for use from the command line, you will need to do so.
  + If some of these steps do not work as expected, please post a public question on Piazza, as other student may have solved similar issues and may be able to help.
  + The jar and scripts should work with Java 7 or 8. Java 8 is recommended.

1. Copy your latest version of MyMainTest.java to <dir>/testsrc/edu/.../encode/MyMainTest.java, thus replacing the placeholder file currently there.
2. Run your test suite by executing compileAndRunTests (.sh or .bat, depending on your platform).
3. Rename your initial junit report from report.txt to report-initial.txt in directory <dir>, so that it will remain saved.
4. Review the junit report. If one of your test cases fails, it could be for one of several reasons:
   * Your test case hits a corner case that is not defined in the

requirements and for which your version of encode makes different assumptions than the version we provided. You should (1) modify the test case (**in this copy of MyMainTest.java only**) so that it **passes** and (2) add the following comment right before the @test annotation for that test: “// Failure Type: Corner Case”.

* + Your test behaves incorrectly, which means that there is a bug

in your version of encode that was not caught by our test suite (class MainTest). You should (1) modify the test case (**in this copy of MyMainTest.java only**) so that it passes and (2) add the following comment right before the @test annotation for that test: “// Failure Type: Test Failure”. **Note**: You will not be penalized for this, so there is no need to fix the bug in your own encode implementation; we are actually interested in seeing what kind of bugs our test suite missed in your code.

* + Your test triggered a bug in our version of encode and caused

an SDPBugException: good job! In this case, you should leave the test case as is but add the following comment right before the @test annotation for that test: “// Failure Type: BUG <short explanation of the failure and what you think it’s the corresponding bug>”. Do not worry too much about the explanation and just make your best guess as you will not be penalized for getting a reasonable, but incorrect, explanation.

As an example, if the output of your test was:

edu.gatech.seclass.encode.SDPBugException: You

Arguments used: [-r, 3, -n, -c, “a”, -d, “aeiou”,

C:\Temp\junit23859131\junit44022402542.tmp]

File: n/a

Encode result: throws an

ArrayIndexOutOfBoundsException

You might label the test with:

// Failure Type: BUG: encode fails when passed

more than 3 options, probably due to the storing

of the options in a fixed-size array.

**Note**: This example is not a hint.

* + If you found at least eight unique bugs in the version of encode we provided, you are done.
  + Otherwise, you should add test cases to your test suite (<dir>/testsrc/edu/.../encode/MyMainTest.java) to make our version of encode fail and throw an SDPBugException. If one of the new tests you add does cause a failure in our version of encode, you should add the following comment right before the @test annotation for that test: “// New Failure Type: BUG: <short explanation of the failure and what you think it’s the corresponding bug>”.  
    Note: To extend your test suite, you should add new tests. You may use test cases from the test suite we provided (MainTest), but our tests will not reveal any bug in our code.
  + This task will be completed when either your tests cause at least eight failures or you are stuck and cannot find further bugs. (In this latter case, you will still get partial credit depending on how much you accomplished.)
  + Generate the junit report for the final version of your test suite and save it as report-final.txt in directory <dir>. **All tests in this version should pass, except for the labelled bugs (which should all fail with an uncaught SDPBugException).**

1. Commit and push the following files:
   * <dir>/testsrc/edu/gatech/seclass/encode/MyMainTest.java
   * <dir>/report-initial.txt
   * <dir>/report-final.txt

There is no need to push any other file, as none of the other existing files under directory <dir> is supposed to be modified.

1. Paste this last commit ID on Canvas, as usual.