Homework 4

EE 363 (Fall 2018)

Department of Electrical and Computer Engineering
Clarkson University

Instructions

Please read the instructions carefully before submitting your work.

Note: There are 2 questions in this HW for a total of 70 points.

Note: Solve all problems and upload your answers to Moodle. Whenever you write solutions on paper, you need to scan all documents and upload the files to Moodle.

Note: user stands for your login ID on Polaris (`polaris.clarkson.edu`). This should be the exact same as your CU ID.

Note: Make sure any code you write works on Polaris before uploading your files. You will likely lose many points if your code doesn't compile.

Note: Do not upload any executable or intermediate files as answers to problems, unless specifically asked to do so.

1. [30 points] In *C*, provide an implementation of the function specified below in Listing 1:

Listing 1: Specification of `maxptr`

```
/*@ requires \valid(p) && \valid(q);
    ensures *p <= *q;
    ensures (*p == \old(*p) && *q == \old(*q) ||
           *q == \old(*p) && *p == \old(*q));
*/
void maxptr(int* p, int* q);
```

Write your answer in file `user_maxp.c` and also provide a `main` to test your implementation. You must include a `README` file that explains how to run your program on Polaris.

Deliverable: Upload `user_maxp.c` and the README file to Moodle.

2. [40 points] Download files related to the IntSet type from Moodle.

   Using JUnit, write five unit tests for the IntSet type in a class named IntSetTest. Try to design the tests such that each one checks something different about IntSets. You must include a README file that describes how to run your tests on Polaris.

   Note: For ease of reference, the specification for the IntSet type is included in Listing 2.

   Note: JUnit tests for the Rational type that we discussed in class are on Moodle. Included with those files is a small script that compiles and runs the tests on Polaris.

   Deliverable: Upload IntSetTest.java and the README file to Moodle.

Listing 2: IntSet specification

```
// overview: IntSets are mutable, unbounded sets of integers.
//          A typical IntSet is {x_1, x_2, ..., x_n}.


public class IntSet {


//[note: assignable is clause often omitted from constructor
   specs]
//@ ensures (* _this_ object is initialized as an empty
   intSet *);
public IntSet();


[//optional constructor]
//@ ensures (* _this_ object is initialized to the values in
   arr *);
public IntSet(int[] arr)

//@ assignable \everything;
//@ ensures (* adds x to the elements of _this_, i.e. _this_
   = \old(this) U {x} *);
public void insert (int x);


//optional method
//@ assignable \nothing;
//@ ensures (* \result = array representing all values in
   _this_ *);
public int[] getAll();


//@ assignable \everything;
//@ ensures (* removes x from _this_ such that _this_ =
   \old(_this_) - {x} *);
public void remove (int x);


//@ ensures (* if x is in _this_, \result = true;  otherwise
   \result = false *);
//@ assignable \nothing;
public boolean isIn (int x);
//@ ensures (* \result = the number of elments in _this_ *);
//@ assignable \nothing;
public int size();


/*@ normal_behavior
  @ requires (* _this_ is nonempty *);
  @ assignable \nothing;
  @ ensures (* \result = an arbitrary element of _this_ *);
  @ also
  @ exceptional_behavior
  @ requires  (* _this_ is empty *);
  @ assignable \nothing;
  @ signals (EmptyException e) true;
  @ signals_only EmptyException;
*/
public int choose();

}
```

End