HW7

1. Provide reachability conditions, infection conditions, propagation conditions, and test case values to kill mutants 2, 4, 5, and 6 in Figure 9.1.

Original Method		With Embedded Mutants
`int Min (int A, int B)`		`int Min (int A, int B)`
`{`		`{`
`  int minVal;`		`  int minVal;`
`  minVal = A;`		`  minVal = A;`
`  if (B < A)`	Δ1	`  minVal = B;`
`  {`		`  if (B < A)`
`    minVal = B;`	Δ2	`  if (B > A)`
`  }`	Δ3	`  if (B < minVal)`
`  return (minVal);`		`  {`
`} // end Min`		`    minVal = B;`
	Δ4	`    Bomb();`
	Δ5	`    minVal = A;`
	Δ6	`    minVal = failOnZero (B);`
		`  }`
		`  return (minVal);`
		`} // end Min`

Figure 9.1. Method Min and six mutants.

3. Answer questions (a) through (d) for the mutant on line 6 in the method `sum()`.

```
/**
 * Sum values in an array
 *
 * @param x array to sum
 *
 * @return sum of values in x
 * @throws NullPointerException if x is null
 */
1. public static int sum(int[] x)
2. {
3.     int s = 0;
4.     for (int i=0; i < x.length; i++) }
5.     {
6.        s = s + x[i];
6'.    // s = s - x[i]; //AOR
7.     }
8.     return s;
9.  }
```

(a) If possible, find test inputs that do **not** reach the mutant.

(b) If possible, find test inputs that satisfy reachability but **not infection** for the mutant.

(c) If possible, find test inputs that satisfy reachability and infection, but **not propagation** for the mutant.

(d) If possible, find test inputs that strongly **kill** the mutants.