SE 333 Software Testing

Assignment: White Box Testing

Due date: May 11, 2020, 11:59pm

Late penalties: 1% late penalty per day for up to 7 days. No late submission will be accepted after that.

1. Given the following method in Java, which finds the maximum value in an array of integers…

```
public static int maximumValue(int[] a) {
  if (a != null && a.length > 0) {
    int max = a[0];
    for (int i = 1; i < a.length; i++) {
      if (a[i] > max) {
        max = a[i];
      }
    }
    return max;
  }
  return Integer.MIN_VALUE;
}
```

   a) Design a test suite with the fewest number of test cases that satisfies the statement test criterion.

   Define the test suite as one or more literal arrays:

   T1 = [x, y, …]

   where the values in the array are actual integer values.

   [] can be used to indicate an empty array
   The word 'null' by itself (not inside brackets) can be used to indicate … null

   b) Design a test suite with the fewest number of test cases that satisfies the branch-condition test criterion.

   Define this test suite using the same technique as above

2. Consider the following loop statement in C:

```
Int max_size = 10;
for (n = 0;  n < max_size && (c = getc(yyin)) != EOF && c != ‘\n’; ++n)
        buf[n] = (char) c;
```

a) Derive a set of test cases that satisfy the compound condition adequacy criterion with respect to the loop. Document your test cases by extending the table below

		A	B	C	
ID	input	n < max_size	c = getc(yyin) != EOF	c != ‘\n’	Result
1	n = 5; c = ‘X’	True	True	True	True
2					
. . .					

b) Derive a subset of test cases from the table above that satisfy the modified condition (MC/DC) adequacy criterion with respect to the loop. Express these test cases using their IDs and the clause that they test:

A = 1 and 3 (means test case 1 and test case 3 combine to satisfy MC/DC test adequacy for condition A)
B = . . .

3. Let us consider the following if statement in Java

```
// assume parseArray.length is 4
if ( pos < parseArray.length &&
      (parseArray[pos] == ‘{‘ ||
       parseArray[pos] == ‘}‘ ||
       parseArray[pos] == ‘|‘) ) {
     continue;
  }
```

Derive a set of test cases and show that it satisfies the modified condition (MC/DC) adequacy criterion for this statement. For brevity, abbreviate each of the basic condition as follows:

Room	pos < parseArray.length
Open	parseArray[pos] == ‘{‘
Close	parseArray[pos] == ‘}‘
Bar	parseArray[pos] == ‘|‘

Provide the answer in 2 parts:

1. a table showing the test cases you have designed, similar to this one:

ID	input	Room	Open	Close	Bar	Result
1	pos=1; pa[pos]='a'	True	False	False	False	False
2						
. . .						

2. List of pairs of test cases similar to what you did in question 2:

   Room = 1 and 3

You may want to create a table for compound condition adequacy first but you do not have to deliver it if you don't want to.

Deliverable:
A document containing your solutions to these problems in one of these formats:
PDF
Word
Plain text
Excel