ECE 322

SOFTWARE TESTING AND MAINTENANCE

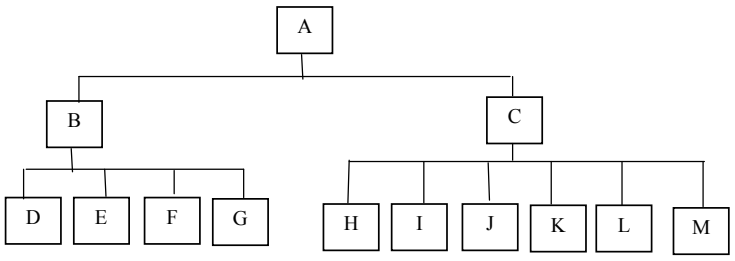
Fall 2015
SOLUTIONS

Assignment #5

Due date: Monday, November 23, 2015 by 3:00 PM
(return to the appropriate box- 2nd floor of ECERF building)

10 points

1. What strategy of integration testing would you recommend for the software system whose dependency tree for the modules is shown below? Justify your choice.



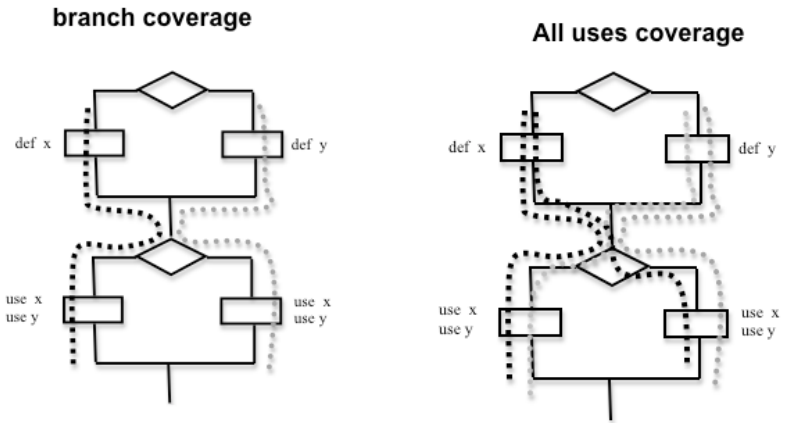
Solution
Bottom-up, considering that there is a stable environment. There are a significant number of modules at the lowest level, which can be tested independently.

10 points

2. Write a function in C such that the *all-uses* coverage criterion produces more test cases than *branch* coverage criterion.

Solution

A code whose control flow graph looks as follows



20 points

3.For the piece of code shown below do the following:
(a) draw a control flow graph
(b) show def-clear paths. Develop test cases that realize all c-use paths.

```
public static double ReturnAverage(int value[],
                           int AS, int MIN, int MAX){
  /*
   Function: ReturnAverage  Computes the  average
   of all  those  numbers in the  input array  in
   the  positive  range  [MIN, MAX]. The  maximum
   size  of the array is AS. But, the  array size
   could be smaller than AS in which case the end
   of input is represented by -999.
  */
     int i, ti, tv, sum;
     double av;
     i = 0; ti = 0; tv = 0; sum = 0;
     while (ti < AS && value[i] != -999) {
         ti++;
         if (value[i] >= MIN && value[i] <= MAX) {
            tv++;
            sum = sum + value[i];
         }
         i++;
     }
     if (tv > 0)
        av = (double)sum/tv;
     else
        av = (double) -999;
     return (av);
}
```

Solution
Having a control flow graph, enumerate def-clear paths for the individual variables and then enumerate all c-use paths.

10 points

4. Fault seeding was applied to two software modules M1 and M2. In both cases there were 70 faults seeded. The cumulative numbers of seeded and indigenous faults found in each module and reported in the same time instances are given below. We claim that there are 12 indigenous faults in M1 and 17 in M2.

M1 – seeded faults found	0	1	8	9	12	15	21	47	59
M2-seeded faults found	2	12	30	31	32	37	45	46	49

How does the confidence levels about the number of indigenous faults change over the course of testing? Plot the corresponding relationships. What percentage of seeded faults should you find to arrive at a reasonable level of confidence, say at least 0.7?

Solution

The expression for confidence, conf, is given as

$$\text{conf} = \frac{\binom{S}{s-1}}{\binom{S+N+1}{N+s}}$$

assuming that n $\leq$ N. Here S – number of faults seeded (70), s –number of detected seeded faults, N- number of actual faults.

Some selected values are included in the table below

	…	62	65	67	69
M1		0.197	0.347	0.498	0.710
M2		0.113	0.242	0.393	0.631

Confidence level of 0.7 requires that 69 out of 70 faults to be found for M1 and 70 for M2.