SENG 426-Final Exam Review Questions

Exercise 1: Consider the following program segment and answer these questions:

1. What percentage statement coverage do tests 1 and 2 combined achieve?
2. What percentage branch coverage do tests 1 and 2 combined achieve?
3. What percentage path coverage do tests 1 and 2 combined achieve?
4. Specify any additional test cases required to achieve 100% **path** coverage (you don't need to document a full specification, just the inputs).

```
void prog(int X, int Y) {
        x = X; y = Y;
        if (x > 100) {
                x = x – 100;
        }
        else {
                x = x + 100;
        }

        if (x <= y) {
                y = 2*y;
        }
        else {
                if (x > 200) {
                        x = x - 100;
                }
                else {
                        x = x*x;
                }
        }
        printf("x=%d. y=%d",x, y);
}
```

Test 1	Input: X=10, Y=100
Test 2	Input: X=101, Y=100

Exercise 2: Consider a safety device responsible for monitoring a chemical processing plant. The device monitors various parameters of the plant, and according to the values of these parameters, it generates an *alarm* and/or *shutdown* the plant. The parameters monitored include the *pressure, the temperature*, and the maximum *volume of toxic gas* generated. For each of these parameters, the device monitors their levels, which according to their values, fluctuate between *high* and *low*. The operational status of the device is controlled by an *on/off* signal labeled *deviceCtl.* The following truth table describes the operational rules of the device (*high/on=1,low/off=0*).
Derive a test suite for the alarm function (*Z*) using the variable negation test strategy.

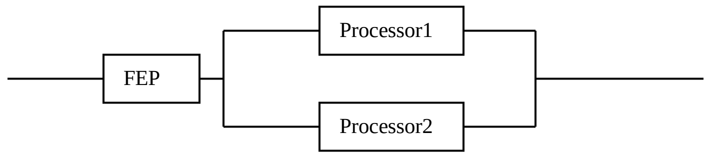
Input vector Number	Conditions				Actions	
	DeviceCtl (A)	**TempLevel (B)**	**PressureLevel (C)**	**GasVolume (D)**	**Alarm (Z)**	**ShutdownCtl (W)**
0	0	0	0	0	0	0
1	1	0	0	0	0	0
2	0	1	0	0	0	0
3	1	1	0	0	1	0
4	0	0	1	0	0	0
5	1	0	1	0	1	0
6	0	1	1	0	0	0
7	1	1	1	0	1	1
8	0	0	0	1	0	0
9	1	0	0	1	1	1
10	0	1	0	1	0	0
11	1	1	0	1	1	1
12	0	0	1	1	0	0
13	1	0	1	1	1	1
14	0	1	1	1	0	0
15	1	1	1	1	1	1

Exercise 3: A space vehicle will use three identical onboard computers, each with an independently designed and developed program that performs the identical function. The overall reliability goal for this vehicle is 0.999 for a 100-hr mission. Development costs are estimated as follows for different alternatives.

Program		Computer	
Reliability (100 hr)	**Cost**	**Reliability (100 hr)**	**Cost**
0.91	$ 3M	0.91	$ 300 K
0.92	$ 3.25 M	0.93	$ 400 K
0.93	$ 3.5 M	0.95	$ 500 K
0.94	$ 3.75 M	0.96	$ 600 K
0.95	$ 4 M	0.97	$ 800 K
0.97	$ 4.5 M	0.98	$ 1.5 M
0.99	$ 5 M	0.99	$ 5 M

The computers operate simultaneously. Outputs are subjected to a sanity check. If any output passes, the system is considered operational. The sanity checking routine is considered to have a reliability of 1. Note that if a computer fails, the sanity checker cannot run and the output for that machine is considered to have failed. How should we allocate reliability goals to the computer and the program to minimize total development cost?

Exercise 4: A computing center has reliability objective of 0.90 for an 8-hr shift for its interactive service. The system requirement is simply that service be provided, regardless of the response time involved. All reliabilities are measured with respect to this shift. It has a dual processor configuration fed by a front-end processor (FEP), as shown in the figure below. The front-end processor has a reliability of 0.99 and its operating system, 0.95. The reliability of each mainframe processor is 0.98. What must the reliability of the mainframe operating system be to meet the overall reliability objective?



Exercise 5: The Soft Landing software service company has won a service contract to provide recovery service for a patient control and billing system. The service is provided to doctors in a region that owns personal computers. It has a failure intensity of 1 failure/100 CPU hr. The average system runs 10 CPU hr/week and there are 600 doctors to be serviced. The average service person can make 2 service calls daily, including travel and allowing for enough spare personnel to prevent excessive service backlogs from building up.
How many service personnel do we need? Assuming a cost of $200/call, what annual contract fee must we charge each doctor to achieve 20 percent profit on sales?

Exercise 6: Assume that a safety-critical program used in a pacemaker will experience 100 failures in infinite time. The failure intensity at the beginning of testing is 10 failures/CPU hr. The program undergoes an extensive testing process, with the objective of achieving a failure intensity of 0.0004 failure/CPU hr before releasing the software.

1. Calculate the number of failures the test team needs to identify to achieve this objective.
2. Calculate the corresponding execution time.
3. Calculate the overall cost of the testing effort assuming that 6 person hr of effort (per failure) is required for failure identification and correction. Assume also that computer time costs $50 per CPU hr, and the hourly wage for a test engineer is $100.
4. Assuming a 5 days/week–8hr/day work schedule and a 4 persons test team, how long will the testing last?

Exercise 7: A personal computer manufacturer wishes to know if the software vendor providing the operating system for a new machine has met the failure intensity objective of 0.1 failure/CPU hr. A series of tests are applied and failure data are recorded. The first failure occurs at 8 CPU hr; the second failure occurs at 19 CPU hr; the third failure occurs at 60 CPU hr. What can the manufacturer conclude at this stage of the testing? You may build a reliability demonstration chart to assist you in this decision-making, using the following parameters:

- Developer risk: α=10%
- Customer risk: β=10%
- Discrimination ratio: γ=2

Exercise 8: We are about to launch the third release of an inventory management software system.

- The size of the initial release was 50 KLOC, with a defect density of 2.0 def/KSSI.
- In the second release, 20KLOC of code was newly added or modified (20% of which are changed lines of code, with no deleted code).
- In the third release, 30KLOC of code was newly added or modified (20% of which are changed lines of code, with no deleted code).
- Assume that the LOC counts are based on source instructions (SSI/CSI).

    a. Considering that the quality goal of the company is to achieve 10% improvement in overall defect rates from release-to-release, calculate the total number of additional defects for the third release.
    b. What should be the maximum (overall) defect rate target for the third release to ensure that the number of total defects does not exceed that of the second release?

Exercise 9:
Consider a software program for operating an elevator system. An elevator can only run if the following invariants are met:

1. 0 < weight ≤ MAX_WEIGHT
2. LOWEST_FLOOR ≤ selectedFloor ≤ HIGHEST_FLOOR
3. selectedFloor != currentFloor
4. door.isClosed
5. !inEmergency

Generate sample test cases for the software program using domain matrix testing strategy.

Exercise 10: In a test project, the number of defects found in different phases of testing are: unit testing (UT), 163 defects; integration testing (IT), 186 defects; system testing (ST), 271 defects.

1. Calculate the overall defect removal effectiveness (DRE), assuming that by conducting fault seeding experimentation, 20 out of 25 seeded defects were detected by the sustaining testers in addition to the 55 (unseeded) reported during the product general availability (GA).
2. Calculate the value of the system test (ST) DRE.
3. Calculate the residual defect density assuming that the size of the release was 500 KLOC, and discuss whether the product was ready to be released from quality standpoint.

Exercise 11:
The following program computes the greatest common divisor of two natural numbers by Euclid’s algorithm.

Begin

```
        input(x,y);
        While(x>0 and y>0) Do
                If (x>y)
                        Then x:=x-y
                        Else y:=y-x
                Endif
        Endwhile
        output(x+y);
End
```

1. Build the corresponding control flow graph
2. Devise test sets (in the form of paths), and then test inputs (i.e., values for x and y) in order to achieve:
    a. statement coverage
    b. edge coverage
    c. condition coverage
    d. path coverage

Exercise 12:
Consider a program for the determination of the nature of roots of a quadratic equation. Its input is a triple of positive integers (say a, b, c) and values in the interval [0,100]. The program output may be one of the following:
[Not a quadratic equation; Real roots; Imaginary roots; Equal roots].

Design the boundary value test cases and robust test cases for the program.

Exercise 13:
Assume a program will experience a total of 200 failures. Initial failure intensity is 16 failure/ CPU hr. It has now experienced 50 failures. Determine the following using the basic execution model:

a. Current failure intensity
b. Decrement of failure intensity
c. Failure intensity at 100 CPU hr.

Exercise 14:
Choose the correct or best alternative in the following:

1. All the modules of the system are integrated and tested as complete system in the case of
   (A) Bottom up testing (B) Top-down testing
   (C) Sandwich testing (D) Big-Bang testing

2. The level at which the software uses scarce resources is
   (A) reliability (B) efficiency
   (C) portability (D) all of the above

3. Modifying the software to match changes in the ever changing environment is called
   (A) adaptive maintenance (B) corrective maintenance
   (C) perfective maintenance (D) preventive maintenance

4. Alpha and Beta Testing are forms of
   (A) Acceptance testing (B) Integration testing
   (C) System Testing (D) Unit testing

5. Changes made to the system to reduce the future system failure chances is called
   (A) Preventive Maintenance (B) Adaptive Maintenance
   (C) Corrective Maintenance (D) Perfective Maintenance

6. The problem that threatens the success of a project but which has not yet happened is a
   (A) bug (B) error
   (C) risk (D) failure

7. The main purpose of integration testing is to find
   (A) design errors (B) analysis errors
   (C) procedure errors (D) interface errors

8. For a function of two variables, boundary value analysis yields
   (A) 4n + 3 test cases (B) 4n + 1 test cases
   (C) n + 4 (D) None of the above

9. Site for Alpha Testing is
   (A) Software Company (B) Installation place
   (C) Anywhere (D) None of the above

10. Which is not a size metric?
    (A) LOC (B) Function count
    (C) Program length (D) Cyclomatic complexity

11. As the reliability increases, failure intensity
    (A) decreases (B) increases
    (C) no effect (D) none of the above

12. Software deteriorates rather than wears out because
    (A) software suffers from exposure to hostile environments.
    (B) defects are more likely to arise after software has been used often.
    (C) multiple change requests introduce errors in component interactions.
    (D) software spare parts become harder to order.

13. Which of these terms is a level name in the Capability Maturity Model?
    (A) Ad hoc (B) Repeatable
    (C) Reusable (D) Organized

14. The ISO quality assurance standard that applies to software engineering is
    (A) ISO 9000 (B) ISO 9001
    (C) ISO 9002 (D) ISO 9003

15. What is the normal order of activities in which software testing is organized?
    (A) unit, integration, system, validation
    (B) system, integration, unit, validation
    (C) unit, integration, validation, system
    (D) none of the above

16. The goal of quality assurance is to provide management with the data needed to determine which software engineers are producing the most defects.
    (A) True (B) False

17. Units and stubs are not needed for unit testing because the modules are tested independently of one another
    (A) True (B) False