Q)The Pesticide Paradox of testing says that --> **methods to find bugs leave subtler bugs**
Q)The Complexity Barrier of testing says that --> **complexity of software grows to the limit of managerial ability**
Q)The probability of showing that the software works --> **decreases as testing increases**
Q)According to phase 0 of tester's mental life --> **testing and debugging are same**
Q)According to phase 4 of tester's mental life --> **testing is a mental discipline**
Q)According to phase 1 of tester's mental life --> **testing shows software works**
Q)The primary goal of testing is --> **bug detection**
Q)Tight code for optimizing the performance --> **destroys quality**
Q)Which of the following is a static analysis method? --> **Type checking**
Q)SQA stands for --> **Software quality assurance**
Q)Which of the following is true with regard to testing? --> **testing starts with known conditions**
Q)Which of the following is true with regard to debugging? --> **debugging doesn't have a robust theory**
Q)Which of the following is false with regard to debugging? --> **debugging has a robust theory**
Q)Which of the following is a correct sequence from a low-level to a high-level testing? --> **unit, component, and system testing**
Q)Programming-in-the-large means --> **writing programs with many components**
Q)Testing starts with --> **known conditions**
Q)Debugging is a --> **deductive process**
Q)Structural testing does look at --> **implementation details**
Q)A module is a --> **discrete, well-defined component**
Q)System testing does --> **test the system as a whole**
Q)Sadism suffices state that --> **sadistic streak is sufficient to remove bugs**
Q)Control bug dominance says that --> **errors in control dominate bugs**
Q)Bug locality hypothesis means --> **bug in a component is local**
Q)Silver bullets means --> **there exists X granting immunity from bugs**
Q)Angelic tester means --> **tester who is better at test design**
Q)A program's environment is the --> **hardware and software to make it run**
Q)Initialization bugs arise out of --> **improper initial values to variables**
Q)Subtle bugs have no --> **definable pattern**
Q)Acceptance test is done --> **to know whether to accept a system or not**
Q)Integration testing does test --> **components while integrating them**
Q)Which of the following is the correct order of dealing with the nightmare list for bugs? --> **listing of nightmares, converting consequences into cost, sorting them, ranking them**
Q)Which among the following lists of examples represent mild, catastrophic, and infectious consequences of bugs respectively? --> **misaligned output, automatic shutdown, spreading erratic code**
Q)The first step of the nightmare approach of finding consequence of bugs is --> **listing of worst software nightmares**
Q)The importance of bugs depends on --> **correction cost, installation cost, consequences, frequency**
Q)Consequences of bugs range from --> **mild to catastrophic**
Q)Correction cost of a bug depends on --> **cost of discovery and correction**

Q)Bug importance can be given by the formula --> **importance ($) = frequency*(correction _cost + installation _cost + consequential _cost)**
Q)Example of a mild consequence of a bug --> **misspelled output**
Q)An example for an infectious consequence of a bug could be --> **spreading of malfunctioning code**
Q)Flexible severity rather than absolutes says that it is better to have --> **flexibility in dealing with bugs**
Q)Which among the following lists of examples correspond to processing bugs, logical bugs, and control bugs respectively? --> **spaghetti code, improper negation, arithmetic bug**
Q)Improper layout of cases, superfluous initial values, and negative zero are examples of which categories of bugs respectively? --> **logical bugs, initialization bugs, processing bugs**
Q)Uninitialized variable attempting to use a variable before it exists is a --> **dataflow bug**
Q)Arithmetic bugs come under the section of --> **processing bug**
Q)Syntax errors comes under the category of --> **coding bugs**
Q)Data bugs arise out of --> **specification of data objects**
Q)Which among the following is a logical bug? --> **improper usage of "greater than"**
Q)A data dictionary contains --> **database documentation**
Q)TQA stands for --> **test quality assurance**
Q)Documentation bugs lead to incorrect --> **maintenance**
Q)Sequence of steps in complete testing includes --> **exercise every path, exercise every statement, and exercise every branch and case**
Q)Which of the following statements is true with regard to flowchart? --> **Flowchart doesn't have fixed rules for design.**
Q)Which of the following statements is true regarding control flowgraphs? --> **In control flowgraphs, details of the process block are not shown.**
Q)Logic errors are inversely proportional to the probability of --> **path execution**
Q)A path segment basically consists of a succession of --> **consecutive links**
Q)Path testing is the name given to a family of test techniques based on --> **selecting a set of test paths**
Q)A control flowgraph is a graphical representation of a program's --> **control structure**
Q)Complete coverage is given by --> **C1 + C2**
Q)A junction generally is denoted by the symbol --> **circle**
Q)PDL stands for --> **program design language**
Q)Correct Predicate: X: = 7 ---- IF Y > 0 THEN
Buggy Predicate: X: = 7 ---- IF X + Y > 0 THEN
This is an example of --> **Assignment blindness**
Q)Correct Predicate: Y: = 2 ---- IF X + Y > 3
Buggy Predicate: IF Y = 2 THEN---- IF X > 1 THEN----
This is an example of --> **Equality blindness**
Q)Correct Predicate: X: = A --- IF X - 1 > 0 THEN ---
Buggy Predicate: X: = A ---- IF X + Y - 2 > 0 THEN
This is an example of --> **Self blindness**
Q)ABCD + ABC + AB + A can be reduced to --> **A**
Q)ABCD + BCD + CD + AB can be reduced to --> **CD + AB**
Q)A predicate is a --> **logical function**
Q)A path predicate is a --> **predicate associated with a path**

Q)The act of symbolic substitution of operations along the path in order to express the predicate in terms of input vector is called --> **predicate interpretation**
Q)If a variable's value can change as a result of the processing the variable then, the variable is --> **process dependent**
Q)If every combination of values of two variable values cannot be independently specified then they are said to be --> **correlated**
Q)A heuristic procedure for sensitizing paths checks for predicate coverage in following order --> **independent uncorrelated, correlated, dependent, correlated dependent**
Q)Which of the following set of steps form the correct sequence of steps for heuristic procedure for sensitizing paths? --> **identifying variables, classifying predicates, and path selection**
Q)Which of the following sets correspond to the given options respectively?
i) X + Y = 3Z
ii) IF case = 1 DO A1 ELSe (IF case = 2 DO A2 ELSE DO A3 ENDIF) ENDIF --> **path predicate expression and multiway branch**
Q)The first step in the heuristic procedure for sensitizing paths is --> **identifying the variables that affect decisions**
Q)The last step in the heuristic procedure for sensitizing paths is --> **satisfying the inequalities**
Q)If the solution can be found, then the given path is said to be --> **achievable**
Q)The act of finding a set of solution to the path predicate expression is called --> **path sensitization**
Q)ABC + BCD + CDE + EFG is said to be in the form of --> **sum-of-products**
Q)If the solution cannot be found to any of the sets of inequalities, then the path is said to be --> **unachievable**
Q)When some of the necessary conditions are satisfied and the desired outcome is achieved for wrong reason, then such a situation is called --> **coincidental correctness**
Q)Which among the following sets correspond well to the instrumentation methods such as bitmap method, hash code scheme, and symbolic debugger respectively? --> **single bit per link, check sum, dummy subroutine**
Q)Execution of every statement in order and recording of the intermediate values of all calculations is called a/an --> **interpretive trace program**
Q)Which among the following is not a method used to instrument paths? --> **configuring test markers**
Q)Which among the following is a method that uses a dummy subroutine to instrument paths? --> **symbolic debugger**
Q)Link markers and link counters come under the category of --> **instrumentation probes**
Q)The outcome of a test is --> **what we expect to happen**
Q)The process of confirming that the outcome was achieved by the intended path is known as --> **path instrumentation**
Q)A link marker is also known as a --> **traversal marker**
Q)A name given to a link is known as --> **link marker**
Q)A variable value that increments whenever a link is traversed is called a --> **link counter**
Q)Stubs are used where it is clear that the bug potential for the stub is --> **lower than that of the called component**
Q)The bugs in the rehosting process will be in the --> **translation algorithm**
Q)The test suite and the associated outcomes becomes the _ _ _ _ _ for the rehosted software. --> **specification**

Q)The cost of rehosting process is comparable to the cost of --> **rewriting the software**
Q)Rehosting process can be made powerful and effective by the usage of --> **automatic structural test generators**
Q)A simulator of lower-level components that is presumably more reliable than the actual component is called a --> **stub**
Q)The integration strategy that is used in implementation of path testing is --> **bottom-up approach**
Q)The process of changing the operating environment to support old software is --> **rehosting**
Q)Path testing with C1 + C2 coverage is a powerful tool for rehosting --> **old software**
Q)For maintenance purpose, path testing is first used on the --> **modified components**
Q)Which is the best way to deal with absorption? --> **follow the predator as the primary flow**
Q)Which of the following is the correct way to deal with biosis? --> **follow the parent flow from beginning to the end**
Q)Which of the following pairs forms a correct sequence of transaction flows? --> **accept input, validate, acknowledge, record transaction**
Q)Which of the following forms a set of troublesome cases? --> **biosis, mitosis, absorption, conjugation**
Q)Which of the following forms the correct set of births? --> **biosis-mitosis**
Q)A unit of work seen from a system user's point of view is called --> **transaction**
Q)TCB stands for --> **transaction control block**
Q)MIMD stands for --> **multi-instruction multiple data**
Q)Which of the following pairs comes under mergers? --> **absorption-conjugation**
Q)Predator versus prey is an example of --> **absorption**
Q)Which of the following methods is useful in the case of inadequate resources in transaction flow testing? --> **mistune**
Q)Which of the following is true with regard to transaction flow instrumentation? --> **counters are not useful**
Q)Usually, the percentage of transaction-flow sensitization that can achieved ranges from --> **80 %-95 %**
Q)The effort required for design and maintenance of test databases in the design of transaction flow tests ranges from --> **30 %-40 %**
Q)Which of the following methods is useful where hard-to-sensitize paths begin? --> **use breakpoints**
Q)OCL stands for --> **object constraint language**
Q)The objective of a PDL representation is --> **to have a trace of actions**
Q)Which of the following are offpaths? --> **exception conditions**
Q)Which of the following is a technique to go through the transaction flows? --> **walkthrough**
Q)Which of following is used for representation of transaction flows? --> **petrinet**
Q)Which of the following is a PDL language example? --> **PAR DO: CSL: =COSL END PAR:**
Q)Which of the following is a PDL statement? --> **PAR DO: CSL: = COSL, CSC: = COSC, SNL: =SINL END PAR**
Q)Which of the following is the correct sequence of Von Neumann microinstruction sequence? --> **fetch, interpret, process, store**
Q)In the context of data object state and usage d, k stand for --> **defined, killed**
Q)In the context of data object state and usage c, p stand for --> **calculation, predicate**

Q)The name given to a family of test strategies for selecting paths through the program's control flow to explore sequence of events related to the status of data objects --> **data-flow testing**
Q)Unreasonable things that can happen to data that can explored by data-flow testing area --> **data-flow anomalies**
Q)An interchangeable storage of instructions and data in the same memory units is supported by an architecture called --> **Von Neumann machine**
Q)A graph consisting of nodes and directed links is called --> **data-flow graph**
Q)MIMD architecture supports --> **parallel execution**
Q)A:=0; B = A;. These statements represent which of the following sequences with respect to the variable A. (d - defined, u - used) --> **du**
Q)With regard to the data object state and usage "kd" describes --> **object killed and redefined**
Q)Which of the following is a buggy sequence with regard to the lifetime of a variable? (k - killed, u-used, d - defined) --> **ku**
Q)Which of the following is a perfectly normal sequence wth regard to the lifetime of a variable? (k - killed, u -used, d - defined) --> **uu**
Q)Which of the following statements is buggy? (free() is used to kill the variable) --> **free(x); x=x+1;**
Q)Analysis done on source code without actually executing it is called --> **static analysis**
Q)Analysis done as the program is being executed is called --> **dynamic analysis**
Q)Syntax error detection is an archetypal result of --> **static analysis**
Q)Division by zero warning is the archetypal result of --> **dynamic analysis**
Q)Dummy nodes placed at the outgoing arrowhead of exit statements to complete a graph are called --> **exit nodes**
Q)Data-flow testing strategy in which for variable and every definition of that variable includes at least one definition-free path from the definition to every predicate use is called --> **all-p-uses/some-c-uses strategy**
Q)Data-flow testing strategy in which coverage is ensured by computational-use cases and if any definition is not covered by the previously selected paths, such predicate-use cases are assured is called --> **all-c-uses/some-p-uses strategy**
Q)In the context of data-flow testing strategies ADUP stands for --> **all-du-paths**
Q)Data-flow testing strategy in which every du path from every definition of every variable to every use of that definition be exercised under some test is called --> **all-du-paths strategy**
Q)Data-flow testing strategy in which at least one path segment from every definition to every use that can be reached by that definition is called --> **all-uses strategy**
Q)Path segments are also called --> **subpaths**
Q)A path segment for which every node is visited at most once is called --> **loop-free path segment**
Q)A path segment in which at most one node is visited twice is called --> **simple path segment**
Q)A connected sequence of links such that X (a variable) is defnied on the first link and not redefined or killed on any subsequent link of that segment is called --> **definition-clear path segment**
Q)A refinement of static slicing in which only statements on achievable paths to the statement in question are included is called --> **dynamic slicing**
Q)Which of the following sets of bugs come under domain errors? --> **double zero, float point zero**

Q)If at least two supposedly distinct domains overlap then they are said to be --> **contradictory**
Q)In the context of domain dimensionality, which of the following pair matches to plane and volumes respectively. --> **line and points, planes**
Q)Double zero representation involves a --> **domain error**
Q)If the union of specified domains is incomplete, then the domain is said to be --> **ambiguous**
Q)A testing strategy that is usually applied to one input variable or to simple combinations of two variables based on specification is called --> **domain testing**
Q)A domain basically is a --> **set**
Q)If the points on the boundary belong to the domain, then it is said to be --> **closed**
Q)If the boundary point belong to another domain, then the domain boundary is said to be --> **open**
Q)A testing strategy that divides a programs input space into domains such that all inputs within a domain are equivalent is called --> **partition testing**
Q)Domain ambiguities are --> **holes in the input space**
Q)Three generic cases of complex domains are --> **disconnected pieces, holes, and concavities**
Q)f(X) > = k is an example for _ _ _ _ _ in the context of nice and ugly domains --> **systematic boundary**
Q)Nice domain boundaries are complete, in that, they span the number space from --> **plus to minus infinity in all dimensions**
Q)Which of the following is a correct statement with regard to nice and ugly domains? --> **nice domains are convex, ugly domains are not**
Q)Input vectors for which no path is specified are called --> **incomplete domains**
Q)Input vectors having at least two contradictory specifications over the same segment of the input space are called --> **inconsistent domains**
Q)If every inequality of boundary set V is perpendicular to every inequality of boundary set U, the U and V are said to be --> **orthogonal**
Q)Nice domains are --> **convex**
Q)Nice domains are --> **simply connected**
Q)Using 3x+7y>17 when 7x+3y >17 was intended is an example of --> **tilted boundary**
Q)Using x+y>=17 when x+y>=7 was intended is an example of --> **shifted boundary**
Q)A missing boundary is created by --> **leaving a boundary predicate**
Q)Two types of contradictions possible with ugly domains are --> **overlapped domain specifications and overlapped closure specifications**
Q)Using a wrong operator, for example, x>=l when x>k was intended leads to a --> **closure bug**
Q)A point in the domain such that all points within epsilon neighborhood are also in the domain is called --> **interior point**
Q)A point within an epsilon neighborhood where there are points both in the domain and not in the domain is called --> **boundary point**
Q)A point that does not lie between any two other arbitrary, but distinct points of a convex domain is called --> **extreme point**
Q)A point on the boundary is called --> **on point**
Q)A point near the boundary, but in the adjacent domain where the domain boundary is closed is called --> **off point**
Q)An interface test consists of exploring the correctness of the following mappings --> **caller unit test, integration test, called unit test**
Q)Which of the following forms a correct sequence of interface testing? --> **caller domain-**

caller range, caller range-called domain, called domain-called range

Q)In the context of domain span for domains and interface testing, for every input variable, we need at least --> **compatible domain spans and compatible closures**

Q)For subroutines that classify inputs as valid/invalid, the called routines domain span and closure is usually --> **caller's span and closure**

Q)To confirm compatibility of the caller's range and called routine's domain span, we need to --> **test every input variable independently**

Q)The set of output values produced by a function is called the --> **range**

Q)A set of numbers between the smallest value and the largest value for a single variable is called --> **domain span**

Q)A set of input values over which a function is defined is called --> **domain**

Q)Boundary inequalities related by a simple function such as a constant is called --> **systematic boundary**

Q)If two arbitrary points on any two different boundaries are taken, join them by a line and all points on that line lie within the figure then,comma the figure is called --> **convex**

Q)Which of the linearizing transformation methods yields an infinite polynomials --> **Taylor series**

Q)A canonical program sequence for domain testability can be --> **input, apply linearizing transforms, test, direct**

Q)Which of the following is an example for linearizing transformation for domain testability? --> **polynomials**

Q)The order of the procedure for n boundary equations for coordinate transformations of nice boundaries is --> **O(n square) to 0(n^2)**

Q)Linearization by substituting u= logx, v = logy, w = logz is called --> **logarithmic transform**

Q)Conversion of nonlinear boundaries to equivalent linear boundaries is called --> **linearizing transformations**

Q)When there is a simple pattern to the domain closure, then it is said to be --> **consistent**

Q)In the context of domain testing, the acronym COOOOI stands for --> **closed off outside, open off inside**

Q)If the boundaries span the number space from plus to minus infinity in all dimensions, they are said to be --> **complete**

Q)If a domain is convex, then it is --> **simply connected**