Question 1: (20 points)
a) Why is randomization important? (5 points)

b) List 5 main features that should be randomized while testing the DUT. (5 points)
c) Do the following: (10 points)

```typescript
class Test;
  // two numbers: logic [2:0]   num1, num2;
  // declare the 2 numbers and constraints that allows
  // all values of num1 to be created one at a time.
  // If num1 > 3, then num2 should be
  //   a) between and including 4 and 6 with 30%
  //      probability
  //   b) between and including 1 and 3 with 50% probability
  //   c) any other number with the remaining probability

  // For all other values of num1
  //   num2 should be between and including 2 and 4.

endclass
```
Question 2 (20 points)

a) Randomization Probability: (10 points)

```cpp
class Test5Probability;
    typedef enum {AND, OR, NOT, XOR} operation;
    typedef enum {R1, R2, R3, R4} registers;

    rand operation op1;
    rand registers register_op1;

    constraint c_op1 {
        (op1 == AND) -> (register_op1 == R2);
        (op1 == NOT) -> (register_op1 inside {R2, R3});
        (op1 == XOR) -> (register_op1 inside {R1, R2, R4});
    }

    // CASE A
    solve op1 before register_op1;

    // CASE B
    solve register_op1 before op1;
}
```

What are probabilities of the different cases of op1 and lengths for CASE A and CASE B.

CASE A:
CASE B:

b) What is the probability of “s” being true for both the cases (10 points)

```markdown
class B;
    rand bit s;
    rand bit [22:0] r;
    constraint c1{s->r==0;solve r before s}
endclass

class B;
    rand bit s;
    rand bit [22:0] r;
    constraint c1{s->r==0;solve s before r}
endclass
```

Case 1:

Case 2:
Q3) Please answer the following: (10 points)

a) If the test space for verification is small you will use ____________________________ and for a wider test space you will use ____________________________ [1]

b) It is important to test the design for corner conditions and error handling to see how:
   1. ____________________________ and
   2. ____________________________ [1]

b) The four things you would randomize in your testbench are:
   1. ____________________________
   2. ____________________________
   3. ____________________________
   4. ____________________________ [2]

   a) You achieve faster coverage with ____________________________ vs ____________________________ [1]

d) List the steps to achieve coverage convergence for a complex DUT:
   ___________________________________________________________________________________
   ___________________________________________________________________________________
   ___________________________________________________________________________________ [2]

   e) For the testbench to be reusable and easy to modify it should be ____________________________ and for it to achieve coverage quickly it should be ____________________________ [1]

   f) ____________________________ simplify the verification of reusable IP and capture the designer intent. [1]

   g) In a verification environment the stimulus is applied by ____________________________ to the ____________________________ [1]
Q4) Please answer the following: (10 points)

```c
int q[,] = `{1, 2, 3, 4, 5};
int b[,] = `{3, 4, 4, 5};
int tq[,] = `{};
int j, k;

repeat(2) begin
    repeat (2) begin
        k = b.pop_front;
        q.push_front(k);
    end
    k = q.pop_back;
    b.push_back(k);
end
```

i) What are the values of q and b and k?
Q5) Please answer the following: (20 points)

<table>
<thead>
<tr>
<th>OPCODE 1</th>
<th>DATA1</th>
<th>OPCODE 2</th>
<th>DATA 2</th>
</tr>
</thead>
</table>

Core 1 (16 bits) Core 2 (16 bits)

Imagine that you are modeling an Instruction Cache for a dual core engine. The instruction is shown in figure above. Each Instruction word is divided into CORE 1 and CORE 2, which are 16 bits long. Each CORE contains two fields, OPCODE and DATA, each of which is 8 bits wide.

a) Designer A declares a datatype Instr in the following manner. Calculate the memory required (in bits) to store 1 instruction in the Cache. [2]

```c
bit [7:0] Instr [2][2];
```

b) Designer B declares a datatype Instr in the following manner. Calculate the memory required (in bits) to store 1 instruction in the Cache. [2]

```c
bit [1:0] [7:0] Instr [2];
```
c) Which designer’s code is more efficient in terms of memory consumption? Why? Do you think there is an even more efficient way to declare Instr with minimum memory consumption? If yes, declare a datatype named Instr which can model the above cache and be able to address efficiently at the OPCODE, DATA, CORE and INSTRUCTION level with minimum memory.

[3]

d) Set the value of most significant 3 bits of OPCODE 2 in the first Instruction to 101 (using the declaration in part c).

[2]

e) Write a code to set the value of least significant 4 bits of OPCODE 1 in the first Instruction to 1110 (using the declaration in part a).

[2]
f) How would you model the instruction cache if the size of the cache is not known at run time. [3]

g) Allocate ten elements to the memory you designed in part (f) [3]

h) Initialize the opcode field of core 1 of all the elements to 8'h42 [3]
Q6) Please answer the following: (20 points)

Let us assume that the memory array shown in the figure above is called “Mem”. The memory is readable along the dimensions shown.

a) Write the declaration for this memory

[5]
b) Initialize the array using one foreach loop to hold the sum of dimensions for each memory element. For instance location [1][1][3] should contain 5 and location [3][4][5] should contain 12.

\[5\]

c) Display the contents of the memory at the location marked in grey. (Write pseudo code)
d) Use nested for loops to display each content at the level of a,b,c under dimension x=6->7, y=4, z=4 of the memory. (This means display a,b,c for both x=6 & 7 in the array)