Q1) Please answer the following:

a) __________________ are user defined test cases, used to test corner cases, or areas with high bug probability. [2]

b) The __________________ creates a test scenario, the __________________ converts the test scenario into a valid DUT transaction and the __________________ puts it into the DUT interface. [2]

c) __________________ are self checking test-code, which are often directly tied to the design. These are often written by the designer themselves. [1]

d) __________________ computes the expected result which is used by the __________________ to see for discrepancies between expected and observed output. [2]

e) Lets say you are a verification engineer trying to achieve functional coverage in a design. What would you do when...

   i) .. you run Constraint random tests and they saturate at 70 % coverage?

   ii). you run Constraint random tests and get stuck at 98% coverage?
Q2) Please answer the following:

```c
int q[] = `{1, 2, 3, 4, 5};
int b[] = `{2{6}, 2{7}, {8}};
int tq[] = `{};
int j,k;

repeat(5) begin
    j = q.pop_front;
    b.push_front(j);
    j = b.pop_back;
    q.push_back(j);
end
```

i) What are the values of q and b at this point ?

```c
// Code continued from above.
repeat(5) begin
    k = q.pop_back;
    b.push_front(k);
    k = b.pop_back;
    q.push_front(k);
end
```

ii) What are the values of q and b at this point ? (Bonus Question)
// Overwriting the values of queues again.
q[] = `{1 2 3 4 5};
b[] = `{2{6}, 2{7}, {8}};

iii) Print the output at each step.

tq = q.find_index with (item < 4); [1]

j = b.sum with (item = 8); [2]

tq.delete(j); [1]

b.insert(tq[1],tq[0]) [2]
Q3) Please answer the following:

Imagine that you are modeling an Instruction Cache for a Triple-Core Engine which has the following instruction type (see figure). Each Instruction word is divided into CORE 1, CORE 2 and CORE3 each of which are 16 bits long. Each CORE contains two fields, OPCODE and DATA, each of which is 8 bits wide. Your Instruction Cache can buffer up to 16 such instructions.

a) Declare a datatype named Icache which can model the above cache and be able to address efficiently at the OPCODE, DATA, CORE and INSTRUCTION level with minimum memory consumption.

b) Write an operation to swap the DATA fields of the cores 3 and 2 for the 4th Instruction stored in the Icache (using the declaration in part a).
c) Write a code to set the value of most significant 3 bits of OPCODE 2 in the first Instruction to 101 (using the declaration in part a).

[1.5]

d) 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).

[1.5]

e) Declare a datatype `Icache` (using another method than the one used above) which can model the above cache and be able to address efficiently at the OPCODE, DATA, CORE and INSTRUCTION level with minimum memory consumption.

[4]

f) Using the declaration in part e write an operation to swap the DATA fields of the cores 3 and 2 for the 4th Instruction stored in the Icache.

[3]
g) Using the declaration in part e, write a code to set the value of most significant 3 bits of OPCODE 2 in the first Instruction to 101. [1.5]

h) Using the declaration in part e, write a code to set the value of least significant 4 bits of OPCODE 1 in the first Instruction to 1110. [1.5]

Q4) Please answer the following:

a) What type of array/data type are you likely to use when?
   i) Modeling a 16 GB memory space with would be accessed by only 1024 vectors.
   ii) Creating a FIFO transactor between the Generator and the Driver.

b) Name any two methods to reduce simulation memory usage. [2]
c) Consider the following program body.

```c
int a[];
int b[];
```

i) Allocate 3 elements to a[] and 5 new elements to b[].

ii) Initialize all elements of a[] and b[] to their index value.

iii) Copy a[] at the beginning of b[] and push the existing elements of b[] lower, without using any third variable.

Q5) Please answer the following:

a) Create an enumerated type calling it fsm by assigning values to bit type of size 2 bits as fetch: 11 decode: 10 execute: 01

b) Declare three variables fsmstate1, fsmstate2, fsmstate3.
c) Write a small program to step through
   i) All the states of the fsm
   
   ii) Step through fetch and decode only

   iii) Display the value and name of the state: display only execute

   iv) Display execute, decode, fetch (reverse order)

   d) What might be the potential bug in the above procedure (a,b,c). Change the enumerated type to remove the bug
Q6) Please answer the following:

Let us assume that the memory array is called “Mem”. Also the memory is readable at the a, b, and c dimensions.

a) Write the declaration of this memory

b) Use ONE foreach loop to display the contents of the entire memory along the x, y and z dimensions.
c) Use nested for loops to display each content at the level of a, b, c under dimension x = 3, y = 12 and z = 0 of the memory.