1. Pictorial representation of an expression is called  
a) Expression tree  
b) Operator tree  
c) Expression flow  
d) Expression chart

Answer: b  
Explanation: The operator tree has a tree like format where the evaluation starts from root of the tree.

2. The results of each intermediate operation are created and then are used for evaluation of the next-level operations. This is called  
a) Materialized evaluation  
b) Expression evaluation  
c) Tree evaluation  
d) Tree materialization

Answer: a  
Explanation: The cost of a materialized evaluation is not simply the sum of the costs of the operations involved.

3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ allows the algorithm to execute more quickly by performing CPU activity in parallel with I/O activity.  
a) Buffering  
b) Double buffering  
c) Multiple buffering  
d) Double reading

Answer: a  
Explanation: Double buffering using two buffers, with one continuing execution of the algorithm while the other is being written out.

4. Pipelines can be executed in  
a) 4  
b) 3  
c) 2  
d) 5

Answer: c  
Explanation: Demand driven and producer driven pipelines are the two ways.

5. In a \_\_\_\_\_\_\_\_\_ the system makes repeated requests for tuples from the operation at the top of the pipeline.  
a) Demand-driven pipeline  
b) Producer-driven pipeline  
c) Demand pipeline  
d) All of the mentioned

Answer: a  
Explanation: Each time that an operation receives a request for tuples, it computes the next tuple (or tuples) to be returned, and then returns that tuple.

6. In a \_\_\_\_\_\_\_\_\_\_\_\_\_ operations do not wait for requests to produce tuples, but instead generate the tuples eagerly.  
a) Demand-driven pipeline  
b) Producer-driven pipeline  
c) Demand pipeline  
d) All of the mentioned

Answer: b  
Explanation: Each operation in a producer-driven pipeline is modeled as a separate process or thread within the system that takes a stream of tuples from its pipelined inputs and generates a stream of tuples for its output.

7. Each operation in a demand-driven pipeline can be implemented as an \_\_\_\_ that provides the following functions: open(), next(), and close().  
a) Demand  
b) Pipeline  
c) Iterator  
d) All of the mentioned

Answer: c  
Explanation: After a call to open(), each call to next() returns the next output tuple of the operation.

8. The iterator maintains the \_\_\_\_\_\_\_\_\_\_ of its execution in between calls so that successive next() requests receive successive result tuples.  
a) State  
b) Transition  
c) Rate  
d) Block

Answer: a  
Explanation: The function close() tells an iterator that no more tuples are required.

9. Tuples are generated \_\_\_\_\_\_\_\_\_\_\_ in producer-driven pipelining, they are generated \_\_\_\_\_\_\_\_ on demand, in demand-driven pipelining.  
a) Lazily, Eagerly  
b) Eagerly, Lazily  
c) Slowly, Eagerly  
d) Eagerly, Slowly

Answer: b  
Explanation: Producer-driven pipelining is very useful in parallel processing systems.

10. When two inputs that we desire to pipeline into the join are not already sorted it is the \_\_\_\_\_\_\_\_\_\_\_\_\_ technique.  
a) Hash join  
b) Buffer join  
c) double-pipelined hash join  
d) double-pipelined join

Answer: d  
Explanation: When hash indices are used on tuples, the resultant algorithm is called the double-pipelined hash-join technique.