1. Bitmap indices are a specialized type of index designed for easy querying on \_\_\_\_\_\_\_\_\_\_\_  
a) Bit values  
b) Binary digits  
c) Multiple keys  
d) Single keys

Answer: c  
Explanation: Each bitmap index is built on a single key.

2. A \_\_\_\_\_\_\_ on the attribute A of relation r consists of one bitmap for each value that A can take.  
a) Bitmap index  
b) Bitmap  
c) Index  
d) Array

Answer: a  
Explanation: A bitmap is simply an array of bits.

3.

SELECT \*

FROM r

WHERE gender = ’f’ AND income level = ’L2’;

In this selection, we fetch the bitmaps for gender value f and the bitmap for income level value L2, and perform an \_\_\_\_\_\_\_\_ of the two bitmaps.  
a) Union  
b) Addition  
c) Combination  
d) Intersection

Answer: d  
Explanation: We compute a new bitmap where bit i has value 1 if the ith bit of the two bitmaps are both 1, and has a value 0 otherwise.

4. To identify the deleted records we use the \_\_\_\_\_\_\_\_\_\_\_\_\_\_  
a) Existence bitmap  
b) Current bitmap  
c) Final bitmap  
d) Deleted bitmap

Answer: a  
Explanation: The bitmaps which are deleted are denoted by 0.

5. Bitmaps can be used as a compressed storage mechanism at the leaf nodes of \_\_\_\_\_\_\_\_ for those values that occur very frequently.  
a) B-trees  
b) B+-trees  
c) Bit trees  
d) Both B-trees and B+-trees

Answer: b  
Explanation: Bitmaps are combined and stored in a B+ tree.

6. Bitmaps can be combined with regular B+-tree indices for relations where a few attribute values are extremely common, and other values also occur, but much less frequently.  
a) Bitmap, B-tree  
b) Bitmap, B+tree  
c) B-tree, Bitmap  
d) B+tree, Bitmap

Answer: b  
Explanation: Bitmaps are combined and stored in a B+ tree.

7. In a B+-tree index \_\_\_\_\_\_ for each value, we would normally maintain a list of all records with that value for the indexed attribute.  
a) Leaf  
b) Node  
c) Root  
d) Link

Answer: a  
Explanation: Bitmaps are combined and stored in a B+ tree.

8. A tablespace is further broken down into \_\_\_\_\_\_\_\_  
a) Tablespace  
b) Segments  
c) Extents  
d) Blocks

Answer: b  
Explanation: Segment names are used in create table and create index commands to place tables or indexes on specific database devices.

9. In ordered indices the file containing the records is sequentially ordered, a \_\_\_\_\_\_\_\_\_\_\_ is an index whose search key also defines the sequential order of the file.  
a) Clustered index  
b) Structured index  
c) Unstructured index  
d) Nonclustered index

Answer: a  
Explanation: Clustering index are also called primary indices; the term primary index may appear to denote an index on a primary key, but such indices can in fact be built on any search key.

10. Indices whose search key specifies an order different from the sequential order of the file are called \_\_\_\_\_\_\_\_\_\_\_ indices.  
a) Nonclustered  
b) Secondary  
c) All of the mentioned  
d) None of the mentioned

Answer: c  
Explanation: Nonclustering index are also called secondary indices.