SAT: Structural Testing and Code Coverage

Consider Java's implementation of the LinkedList's `computeIfPresent()` method:

```java
public V computeIfPresent(K key,
                    BiFunction<? super K, ? super V, ? extends V> rf) {
  if (rf == null) {
    throw new NullPointerException();
  }
  Node<K,V> e;
  V oldValue;
  int hash = hash(key);
  e = getNode(hash, key);
  oldValue = e.value;
  if (e != null && oldValue != null) {
    V v = rf.apply(key, oldValue);
    if (v != null) {
      e.value = v;
      afterNodeAccess(e);
      return v;
    } else {
      removeNode(hash, key, null, false, true);
    }
  }
  return null;
}
```

1. What is the minimum number of tests needed for 100% (and why):

a) line coverage?

b) branch coverage?

c) branch+condition coverage?

d) path coverage?

e) MC/DC?

Consider the expression (A & B) || C with the following truth table:

Test case	A	B	C	(A & B) | C
1	T	T	T	T
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	T
6	F	T	F	F
7	F	F	T	T
8	F	F	F	F

2. What test suite(s) achieve(s) 100% MC/DC?

3. Draw the truth table for the expression A & (A||B). What test suite(s) achieve 100% MC/DC? What can you say about this piece of code?