1. **Given R(C, D, E, F, G), and the following functional dependencies: F → E F → G C → D We decompose R into two relations R1(C, D) and R2 (E, F, G, C). Does this decomposition have the lossless join property?**

   Solution:

   Step # 1) Create initial Matrix:

	C	**D**	**E**	**F**	**G**
R1	b11	b12	b13	b14	b15
R2	b21	b22	b23	b24	b25

   Step # 2) Mark Attributes Present in Relations

	C	**D**	**E**	**F**	**G**
R1	a1	a2	b13	b14	b15
R2	a1	b22	a3	a4	a5

   Step # 3) Apply Functional Dependencies

   - Using F → E:
     In initial matrix, we have different symbol corresponding to F in both relations. Hence, this cause no change in Matrix. (Unchanged)
   - Using F → G:
     Again, we have different symbol corresponding to F in both relations. Hence, this cause no change in Matrix. (Unchanged)
   - Using C → D:
     Given that R1 and R2 have the same symbol for C, D in R2 is derived from R1 :

	C	**D**	**E**	**F**	**G**
R1	a1	a2	b13	b14	b15
R2	a1	~~b22~~ a2	a3	a4	a5

     Looking at the final matrix, the row R2 is entirely filled with attribute symbols (a). Hence, this shows that it is decomposition have the lossless join property.

2. **Decompose R into 3rd normal form with both dependency preservation property and loss-less join property.**

   R (a, b, c, d, e, f, g)

   Fd1: a → {b, c}
   Fd2: d → {e, f}

   Solution:

   Step -1) Find Minimal Cover:

a → b
a → c
d → e
d → f

Step-2) Produce relations:

R1 (a, b, c)

R2 (d, e, f)

Step-3)

In above Relation we have b, c, e and f as non-essential attributes and a, d, g are essential attributes.

Closure of {a, d, g} = {a, b, c, d, e, f, g}

Hence, Candidate key is **(a, d, g).**

Given that this key has not been covered by the existing relation schemas, we create a new schema:

R3 (a, d, g)

Step-4) Eliminate Redundant Relations:
No redundant relations.

Hence, resulted 3NF is R1 (a, b, c), **R2** (d, e, f), **R3** (a, d, g).

3. **Given R(x, y, c, z, e, f, g). There are two keys: (x, y) and z. Given the following functional dependency: F = {{x, y} → {c, z, e, f, g}, z → {x, y, c, e, f, g}, f→ x}. Decompose R into BCNF.**

Solution:

Step#1) Finding all candidate keys:

Closure of z = (x, y, c, z, e, f, g)
Closure of (x, y) = (x, y, c, z, e, f, g)
Closure of (f, y) = (x, y, c, z, e, f, g)

Hence,
Prime Attributes: x, y, z, f
Non-Prime Attributes: c, e, g

Step # 2) Determine if each Functional Dependency violate BCNF or not:

- (x, y) → (c, z, e, f, g): It is in BCNF. It does not have any partial and transitive dependencies. Also, (x, y) is a super key. So, no violation to BCNF.
- z → (x, y, c, e, f, g): It is in BCNF. It does not have any partial and transitive dependencies. Also, (x, y) is a super key. So, no violation to BCNF.
- f→ x: This violate BCNF property. Left part of a functional dependency must be a super key. Hence, R will be decomposed to two relations.

R1(f, x) and R2(c, e, f, g, y, z) are resulted BCNF decomposition.