Part 1.1:

1. R = ABC
   Functional Dependencies = AB →C, AC→B, BC→A
   Minimal Cover - YES
   BCNF = YES - where AB, BC, AC are candidate keys for R.

2. R = ABCD
   Functional Dependencies = AB →C, AC→B, BC→A, B→D
   Minimal Cover = YES
   BCNF = NO, decomposing it to BCNF as ABC, BD

3. R = ABCEG
   Functional Dependencies = AB →C, AC→B, BC→A, E→G
   Minimal Cover = YES
   BCNF = NO, decomposing it to BCNF as
   R1(ACB) with FDs: AC → B
   R2(ADEBG) with FDs: AD → E, E → G

4. R = DCEGH
   Functional Dependencies = E→G
   Minimal Cover = YES
   BCNF = NO, decomposing it to BCNF as DCEH, EG

5. R = ACEH
   Functional Dependencies =AB → C, AC → B, AD → E, B → D,BC → A, E → G
   Minimal Cover - YES
   BCNF = YES

Part 1.2:

1. R = ABCDEG

   We have following functional dependencies:
   AB->C
   BC->A
   Because of which Decomposition {AB, BC, ABDE, EG} is lossy.
   Functional Dependencies AB->C OR AC->B are not preserved by the decomposition.

2. R = ABCDEG
   The decomposition {ABC, ACDE, ADG}: lossless and not dependency preserving.

Part 2: R (A, B,C,D)

1. B->C, D->A; Decomposes into BC and AD.
   Candidate Key: BD
   Decomposition: Lossy, hence decomposition is poor.
   If we try to join AD and BC then we get result greater than ABCD.

2. AB->C, C->A, C->D; decomposes into ACD and BC.
   Candidate Key: AB
   BC decomposition is lossless – because ACD intersects BD where C->ACD.
   As C is the Key to ACD, the projection on ACD includes C->A, C->D and the projection on BC does not produce any non-trivial dependency.

3. A-> BC, C->AD, decomposes into ABC and AD
   Candidate Key: A, C there is no need to decompose as it is already in BCNF.

4. A->B, B->C, C->D; decomposes into AB and ACD.
   Candidate Key: A
   Decomposition: Lossless – because A is the key, and it does not preserve dependency on B->C