1. Consider the following relational schema:

Reader(RDNR, Surname, Firstname, City, Birthdate)
Book(ISBN, Title, Author, NoPages, PubYear, PublisherName)
Publisher(PublisherName, PublisherCity)
Category(CategoryName, BelongsTo)
Copy(ISBN, CopyNumber, Shelf, Position)
Loan(ReaderNr, ISBN, Copy, ReturnDate)
BookCategory(ISBN, CategoryName)

Enforce that a reader can only borrow up to 20 books. Give a solution using Trigger.

2. Enforce AC→B for every insertion in the relation R (A, B, C, D, E) with an SQL trigger.

3. Enforce AC→B for every update in the relation R (A, B, C, D, E) with an SQL trigger.

4. Consider the simple relation Employee(ID, salary) storing the employee Ids and salaries, where ID is a key. Consider the following two triggers over this relation:

Trigger T1:

```
CREATE TRIGGER T1
AFTER INSERT ON Employee
REFERENCING NEW ROW as New_Emp
FOR EACH ROW
UPDATE Employee
SET salary = 1.1 * (SELECT max(salary)
                    FROM Employee)
WHERE ID = New_Emp.ID;
```

Trigger T2:

```
CREATE TRIGGER T2
AFTER INSERT ON Employee
REFERENCING NEW TABLE AS New_Emp
FOR EACH STATEMENT
UPDATE Employee
SET salary = 1.1 * (SELECT max(salary)
                    FROM Employee)
WHERE ID IN (SELECT ID FROM New_Emp);
```

Assume that relation Employee has no tuple initially. Suppose that we had inserted the following four rows into the Employee table as the result of a single SQL statement:

1	1000
2	2000
3	3000
4	4000

(i) Show the final database state after trigger execution if only trigger T1 is defined.

(ii) Show the final database state after trigger execution if only trigger T2 is defined.