1) Consider the trigger languages of the SQL-99 standard and suppose that the following triggers have been defined on a table T1 with schema T1(A,B,C).

Time 10: CREATE TRIGGER Trig1 AFTER UPDATE ON T1 OF A
FOR EACH STATEMENT .....

Time 20: CREATE TRIGGER Trig2 AFTER UPDATE ON T1 OF A FOR EACH ROW .....

Time 30: CREATE TRIGGER Trig3 AFTER INSERT ON T1 FOR EACH ROW .....

Time 40: CREATE TRIGGER Trig4 BEFORE UPDATE ON T1 OF A,B FOR EACH ROW .....

Time 50: CREATE TRIGGER Trig5 BEFORE UPDATE ON T1 OF A
FOR EACH STATEMENT .....

(a) Suppose that the following UPDATE command is executed:
   UPDATE T1 SET A=A+10;
   (i) (pt. 5) Determine which triggers are activated for this command.
   (ii) (pt. 10) The execution order of the activated triggers.

(b) Suppose that the following UPDATE command is executed:
   UPDATE T1 SET A=A+10, B=B-20;
   (i) (pt. 5) Determine which triggers are activated for this command.
   (ii) (pt. 10) The execution order of the activated triggers.

2) Consider the trigger Default_Home_Ph (defined on slide 57 of the slides set on Active Database System used in class). Assume that the table Employees has the following schema Employees(Emp#, Name, D#, Salary, Office_P, Home_P).

Suppose that the table is empty and the following tuples are inserted (? Denotes null values):
(100, 'Bob Smith', 10, 150000, 765-496-2388, ?)
(200, 'Jon Red', 20, 120000, 765-475-3999, 765-237-2189)
(300, 'Mary Brown', 30, 145000, 765-211-9087, ?)

Write the content of table Employees after the insertion of the above tuples.

3) Write a trigger in ORACLE that monitors the updates on the salary column of the table Employees, whose schema is given in point (2) above. The trigger should check that whenever a salary is modified, the new value is greater than the old value, and if this is the case, the trigger restores the salary to the old value.

4) Consider the table Employee given in point (2) above. Suppose that, as employees may move to different departments, we want to keep track of the departments where each employee has been. To do so we create a new table Past-Dept(Emp#, D#) that keep track of such history. Write a trigger in ORACLE that monitors the table Employees and whenever an employee changes department, it automatically updates the Past-Dept table.
5) (pt. 20) Consider the following trigger defined in SQL-99 on a table TAB1 of schema TAB1(A,B,C)

```
CREATE TRIGGER Trig7
    AFTER UPDATE ON Tab1 OF A
    REFERENCING OLD TABLE AS old-t, NEW TABLE AS new-t
    FOR EACH STATEMENT
    BEGIN ATOMIC
        ......
    END;
```

Assume that TAB1 has the following content:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>200</td>
<td>X</td>
</tr>
<tr>
<td>200</td>
<td>250</td>
<td>X</td>
</tr>
<tr>
<td>400</td>
<td>450</td>
<td>Y</td>
</tr>
<tr>
<td>900</td>
<td>930</td>
<td>Z</td>
</tr>
</tbody>
</table>

Suppose that the following update is executed:

```
UPDATE TAB1 SET A=A+200 WHERE C='X';
```

Write the schema and the contents of the transition tables specified in the trigger after the update has been executed.