Procedure: Answer to Problem Statements

1. **Create Procedure to give names of instructors as o/p.**

   Routine File:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `show_name_of_ins`()
BEGIN
        select name
    from instructor;
END
```

   Actual SQL Script:

```
USE `university`;
DROP procedure IF EXISTS `show_name_of_ins`;

USE `university`;
DROP procedure IF EXISTS `university`.`show_name_of_ins`;
;

DELIMITER $$
USE `university`$$
CREATE PROCEDURE `show_name_of_ins`()
BEGIN
        select name
    from instructor;
END$$

DELIMITER ;
;
```

```
call show_name_of_ins();
```

---

2. **Create Procedure where you can pass an integer value and give output as name and salary of those instructors who have salary value greater than the passed value.**

Routine File:

```
CREATE PROCEDURE `ins_with_greater_sal`(
	ins_having_sal_gt int
)
BEGIN
	select name
	from instructor
	where salary > ins_having_sal_gt;
END
```

Actual SQL Script:

```
USE `university`;
DROP procedure IF EXISTS `ins_with_greater_sal`;

USE `university`;
DROP procedure IF EXISTS `university`.`ins_with_greater_sal`;
;

DELIMITER $$
USE `university`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `ins_with_greater_sal`(
	ins_having_sal_gt int
)
BEGIN
	select name, salary
   from instructor
   where salary > ins_having_sal_gt;
END$$

DELIMITER ;
;
```

```
call ins_with_greater_sal(20000);
```

---

3. **Create a SP which will take instructor ID as input and will give output the details of that particular instructor whose ID was passed.**

```
delimiter //
create procedure getIns (in insid int)
begin
select *
        from instructor
       where ID = insid;
end //
delimiter ;
```

```
call getIns (103);
```

---

4. **Create a SP to which will take instructor name as input and will give output the details of that particular instructor whose name was passed.**

```
delimiter //
create procedure get_name (in ins_name varchar(50))
begin
        select *
        from instructor
where name = ins_name;
end //
delimiter ;
```

```
call get_name('Lexi');
```

---

5. **Create a SP to check whether an instructor exists in the database if you know his name and ID.**

```
delimiter //
create procedure check_ins (in ins_id int, in ins_name varchar(50))
begin
        select *
        from instructor
        where ID = ins_id and
        name = ins_name;
end //
```

```
delimiter ;
```

```
call check_ins (100, 'Lexi');
```

---

6. **Create a SP to check whether a number is even or odd.**

```
delimiter //
create procedure check_even_odd(num int)
begin
	if num%2 = 0 then
		select 'Number is even' as Result;
	else
		select 'Number is odd' as Result;
	end if;
end //
delimiter ;
```

```
call check_even_odd(4);
```

OUT Parameter

7. **Create a SP where you can pass department name and a variable. The variable shall store number of instructors working under the passed department name. Output the value of the variable.**

```
delimiter //
create procedure count_ins_fromDept (in dname varchar(30), out totalIns int)
begin
	select count(ID) into totalIns
   from instructor
   where dept_name = dname;
end //
delimiter ;
```

```
call count_ins_fromDept ('Comp. Sci.', @countIns);
select @countIns as TotalIns;
```

IN OUT Parameter

8. **Create a procedure which will take a value through a variable and increase the variable's value by 5**

```
delimiter //
create procedure increase_valby5 (inout val int)
begin
        set val = val + 5;
end //
delimiter ;
```

```
set @val = 0;
call increase_valby5 (@val);
select @val as Value_of_val;
```
