Home assignment #3 (452 &352),
Due date 4/22 5:30PM (total points = 100)

Q1. (10 points) Below is a sample of searched CASE statement. It finds the average salary of the employees in the emp table, using $1000 as the lowest salary possible.

```
Column  Average_Salary format $99,999.99

SELECT  AVG (CASE  WHEN sal > 1000 THEN sal
                        ELSE 1000
                        END )   Average_Salary
   FROM emp ;

OUTPUT

AVERAGE_SALARY
--------------
  $2,091.07
```

Please write a PL/SQL block to do the same job as above: calculate then print out the average salary of that company - using table emp. In your program, use cursor to retrieve each employee's sal(ary), if sal < 1000, count this sal as 1,000; if sal >= 1,000, then leave the sal as is.

Q2. (10 points)
First, please create a temporarily table called temp_emp, as

```
CREATE TABLE Temp_EMP AS SELECT * FROM emp;
```

You may need to run "drop table temp_emp" if that table already exists.

Write an anonymous PL/SQL block. Give a 10 % raise of the salary to each employee who works in department 10; delete the employees who work in the department with deptno = 20. Using SQL%ROWCOUNT, display how many rows are affected for each of the two statements.

Q3. (5 points) Run the following block. The output of code below will be "`No, cursor is NOT found`". Please edit the codes, your new codes will make the output be "`Yes, cursor is found`".

```
DECLARE
      CURSOR c1 IS
         SELECT last_name FROM employees;
      ename     employees.last_name%TYPE;

BEGIN
      OPEN c1;
      IF c1%FOUND THEN
            DBMS_OUTPUT.PUT_LINE('Yes, curse c1 is found');
      ELSE
            DBMS_OUTPUT.PUT_LINE('No, curse c1 is NOT found');
      END IF;
      FETCH c1 INTO ename;
```

```
      CLOSE c1;
END;
```

Q4. (10 points)

Please edit the codes, using cursor for loop ("For index IN cursor_name LOOP"). The output of your new codes should be the same as the original program.

```
DECLARE
    CURSOR cemp  IS
      SELECT   ename, sal
      FROM     emp
      WHERE    deptno = 10
      ORDER BY sal DESC;
    Emp_name      emp.ename%TYPE;
    salary        emp.sal%TYPE;
BEGIN
    OPEN cemp;
    LOOP
      FETCH cemp INTO Emp_name, salary;
      EXIT WHEN cemp%NOTFOUND;
      DBMS_OUTPUT.put_line
       ('current row number is: '|| cemp%ROWCOUNT ||
         ':  ' || RPAD (emp_name, 10) || ': ' ||salary ||'.');
    END LOOP;
    CLOSE cemp;
END;
```

Q5. (10 points) Write an anonymous PL/SQL block, based on table employees, the program will print out the employee ID, full name, salary of those employees whose salary are among the top five. In this question, you are required to define one ROWTYPE variable (for the fetch into statement)

(You may use either the attribute %RWOCOUNT, or define a counter)

(This clause may help: order by salary desc)

Q6. (5 points) Same request as question #5 above. But this time, No ROWTYPE variable will be used, instead you define the column level variables.

Q7. (20 points)

Assume that the company has decided a one-time bonus for the employees in department 80 and 30 only, the amount is decided as below.

For employees that get commission_pct equal to or great than .2:

if salary > = 10000 then bonus := 500

if salary < 10000 and salary >= 8000 then bonus := 600

if salary >= 7000 and salary < 8000 then bonus := 700

If salary < 7000 then bonus := 800

For employees that get commission_pct less than .2 but > 0:

if salary > = 10000 then bonus := 400
if salary < 10000 and salary >= 8000 then bonus := 500
if salary >= 7000 and salary < 8000 then bonus := 600
If   salary < 7000 then bonus := 700

For employees that do not get commission_pct
if salary > = 10000 then bonus := 300
if salary < 10000 and salary >= 3000 then bonus := 400
if salary >= 2600 and salary < 3000 then bonus := 500
If   salary < 2600 then bonus := 600

Write an anonymous PL/SQL program, retrieve the info needed for each employee (department 30 and 80 only), calculate the amount of bonus he/she should get. Accumulate the bonus by all employees together, print out this amount at the end of the program.

Q8. (15 points) This question is based on the table employees.
Write a program, declare a cursor that has a parameter. The cursor will report employee ID, first name and last name in a special department, the department ID should be defined as cursor parameter (IN mode). In the PL/SQL block, open the cursor two times with different department id, first time 20, second time 90. For each department (20 and 90), print out the employee first name and last name in that department.

Q9. (15 points) Cursor for update, based on table employees. Write an anonymous PL/SQL program, it will increase the salary of 5% for those employees whose salary is less than 2,499 and without commission.
Your program should declare a cursor with “FOR UPDATE” request, thus it can request the system to lock those records retrieved and change these records later.
After update, print out those affect employees, display their ID, last name, old salaries (before this time increase) and new salary.
It is better to rollback after the program.