1) Write a PL/SQL program to arrange the number of two variable in such a way that the

small number will store in num\_small variable and large number will store in num\_large variable.

**DECLARE**

**num\_small NUMBER := 8;**

**num\_large NUMBER := 5;**

**num\_temp NUMBER;**

**BEGIN**

**IF num\_small > num\_large THEN**

**num\_temp := num\_small;**

**num\_small := num\_large;**

**num\_large := num\_temp;**

**END IF;**

**DBMS\_OUTPUT.PUT\_LINE ('num\_small = '||num\_small);**

**DBMS\_OUTPUT.PUT\_LINE ('num\_large = '||num\_large);**

**END;**

**/**

2) Write a PL/SQL procedure to calculate the incentive on a target achieved and display the message either the record updated or not.

**DECLARE**

**PROCEDURE test1 (**

**sal\_achieve NUMBER,**

**target\_qty NUMBER,**

**emp\_id NUMBER**

**)**

**IS**

**incentive NUMBER := 0;**

**updated VARCHAR2(3) := 'No';**

**BEGIN**

**IF sal\_achieve > (target\_qty + 200) THEN**

**incentive := (sal\_achieve - target\_qty)/4;**

**UPDATE emp**

**SET salary = salary + incentive**

**WHERE employee\_id = emp\_id;**

**updated := 'Yes';**

**END IF;**

**DBMS\_OUTPUT.PUT\_LINE (**

**'Table updated? ' || updated || ', ' ||**

**'incentive = ' || incentive || '.'**

**);**

**END test1;**

**BEGIN**

**test1(2300, 2000, 144);**

**test1(3600, 3000, 145);**

**END;**

**/**

3) Write a PL/SQL program to check whether a number is even or odd.

**DECLARE**

**n1 NUMBER := 54;**

**BEGIN**

**-- test if the number provided by the user is even**

**IF MOD(n1,2) = 0 THEN**

**DBMS\_OUTPUT.PUT\_LINE ('The number. '||n1||**

**' is even number');**

**ELSE**

**DBMS\_OUTPUT.PUT\_LINE ('The number '||n1||' is odd number.');**

**END IF;**

**DBMS\_OUTPUT.PUT\_LINE ('Done Successfully');**

**END;**

**/**

Output: The number. 54 is even number

Done Successfully

4) Write a PL/SQL program to check whether a date falls on weekend i.e. SATURDAY or SUNDAY.

**DECLARE**

**dt1 DATE := TO\_DATE('05-07-2023', 'DD-MM-YYYY');**

**get\_day VARCHAR2(15);**

**BEGIN**

**get\_day := RTRIM(TO\_CHAR(dt1, 'DAY'));**

**IF get\_day IN ('SATURDAY', 'SUNDAY') THEN**

**dbms\_output.new\_line;**

**DBMS\_OUTPUT.PUT\_LINE**

**('The day of the given date is '||get\_day||' and it falls on weekend');**

**ELSE**

**dbms\_output.new\_line;**

**DBMS\_OUTPUT.PUT\_LINE ('The day of the given date is '||get\_day||' and it does not fall on the weekend');**

**END IF;**

**DBMS\_OUTPUT.PUT\_LINE ('Execution done successfully.');**

**END;**

**/**

Output: The day of the given date is WEDNESDAY and it does not fall on the weekend

5) Write a PL/SQL program to check whether a given number is positive, negative or zero.

**DECLARE**

**num1 NUMBER := -5;**

**BEGIN**

**IF num1 < 0 THEN**

**DBMS\_OUTPUT.PUT\_LINE ('The number '||num1||' is a negative number');**

**ELSIF num1 = 0 THEN**

**DBMS\_OUTPUT.PUT\_LINE ('The number '||num1||' is equal to zero');**

**ELSE**

**DBMS\_OUTPUT.PUT\_LINE ('The number '||num1||' is a positive number');**

**END IF;**

**END;**

**/**

Output: The number -5 is a negative number

6) Write a program in PL/SQL to insert records from one table to another.

**DROP TABLE emp\_temp;**

**CREATE TABLE emp\_temp (**

**emp\_id NUMBER,**

**emp\_email VARCHAR2(40)**

**);**

**DECLARE**

**number\_of\_emp NUMBER;**

**BEGIN**

**SELECT COUNT(employee\_id) INTO number\_of\_emp**

**FROM employees;**

**FOR i IN 1..number\_of\_emp LOOP**

**INSERT INTO emp\_temp (emp\_id, emp\_email)**

**VALUES(i, 'not available now');**

**END LOOP;**

**END;**

**/**

7) Write a PL/SQL block to differentiate between CHAR and VARCHAR2 datatype.

**SET SERVEROUTPUT ON;**

**DECLARE**

**f\_name CHAR(15 CHAR);**

**l\_name VARCHAR2(15 CHAR);**

**BEGIN**

**f\_name := 'Allen ';**

**l\_name := 'Munra ';**

**DBMS\_OUTPUT.PUT\_LINE('\*' || f\_name || '\*');**

**DBMS\_OUTPUT.PUT\_LINE('\*' || l\_name || '\*');**

**END;**

**/**

Output:

\*Allen \*

\*Munra \*

8) Write a PL/SQL block to retrieve the first 3 characters of each employee's last name in the employees table.

**DECLARE**

**v\_first\_three\_chars VARCHAR2(3);**

**BEGIN**

**FOR emp IN (SELECT last\_name FROM employees) LOOP**

**v\_first\_three\_chars := SUBSTR(emp.last\_name, 1, 3);**

**DBMS\_OUTPUT.PUT\_LINE('First three characters of last name: ' || v\_first\_three\_chars);**

**END LOOP;**

**END;**

**/**

9) Write a PL/SQL block that prompts the user to enter a string and removes any leading and trailing zeros from it. Display the trimmed string.

**DECLARE**

**v\_input\_string VARCHAR2(100);**

**v\_trimmed\_string VARCHAR2(100);**

**BEGIN**

**v\_input\_string := '00000test string00000';**

**v\_trimmed\_string := TRIM('0' FROM v\_input\_string);**

**DBMS\_OUTPUT.PUT\_LINE('Trimmed String: [' || v\_trimmed\_string || ']');**

**END;**

Output: Trimmed String: [test string]

10) Write a PL/SQL program to display the names of all countries.

**DECLARE**

**v\_country\_name country.country\_name%TYPE;**

**CURSOR c\_countries IS SELECT country\_name FROM country;**

**BEGIN**

**OPEN c\_countries;**

**FETCH c\_countries INTO v\_country\_name;**

**WHILE c\_countries%FOUND LOOP**

**DBMS\_OUTPUT.PUT\_LINE(v\_country\_name);**

**FETCH c\_countries INTO v\_country\_name;**

**END LOOP;**

**CLOSE c\_countries;**

**END;**

11) Write a program in PL/SQL to find the number of rows effected by the use of SQL%ROWCOUNT attributes of an implicit cursor

**DROP TABLE emp\_temp;**

**CREATE TABLE emp\_temp AS**

**SELECT employee\_id, first\_name, last\_name,email**

**FROM employees;**

**BEGIN**

**UPDATE emp\_temp**

**SET email = 'not available'**

**WHERE first\_name LIKE 'B%';**

**dbms\_output.Put\_line('Number of record updated: '**

**||To\_char(SQL%rowcount));**

**END;**

**/**

12) Write a program in PL/SQL to display a cursor based detail information of employees from employees table.

**DECLARE**

**CURSOR z\_emp\_info IS**

**SELECT employee\_id,**

**first\_name,**

**last\_name,**

**salary**

**FROM employees;**

**r\_emp\_info z\_emp\_info%ROWTYPE;**

**BEGIN**

**OPEN z\_emp\_info;**

**LOOP**

**FETCH z\_emp\_info INTO r\_emp\_info;**

**EXIT WHEN z\_emp\_info%NOTFOUND;**

**dbms\_output.Put\_line('Employees Information:: '**

**||' ID: '**

**||r\_emp\_info.employee\_id**

**||' Name: '**

**||r\_emp\_info.first\_name**

**||' '**

**||r\_emp\_info.last\_name);**

**END LOOP;**

**dbms\_output.Put\_line('Total number of rows : '**

**||z\_emp\_info%rowcount);**

**CLOSE z\_emp\_info;**

**END;**

**/**

13) Write a program in PL/SQL to declare a record datatype with same datatype of tables using %TYPE attribute.

**DECLARE**

**CURSOR cur\_emp\_detail IS**

**SELECT employee\_id,**

**first\_name,**

**last\_name,**

**salary**

**FROM employees;**

**TYPE type\_record\_type IS RECORD (**

**emp\_id employees.employee\_id%TYPE,**

**emp\_f\_name employees.first\_name%TYPE,**

**emp\_l\_name employees.last\_name%TYPE,**

**emp\_s\_salary employees.salary%TYPE );**

**emp\_rec\_type type\_record\_type;**

**BEGIN**

**OPEN cur\_emp\_detail;**

**LOOP**

**FETCH cur\_emp\_detail INTO emp\_rec\_type;**

**EXIT WHEN cur\_emp\_detail%NOTFOUND;**

**dbms\_output.Put\_line('Employees Information:: '**

**||' ID: '**

**||emp\_rec\_type.emp\_id**

**||'| Name: '**

**||emp\_rec\_type.emp\_f\_name**

**||' '**

**||emp\_rec\_type.emp\_l\_name**

**||'| Salary: '**

**||emp\_rec\_type.emp\_s\_salary);**

**END LOOP;**

**dbms\_output.Put\_line('Total number of Employees : '**

**||cur\_emp\_detail%rowcount);**

**CLOSE cur\_emp\_detail;**

**END;**

**/**

14) Write a program in PL/SQL to declare a record datatype with same datatype of tables using %TYPE attribute.

**DECLARE**

**CURSOR emp\_cur\_detail IS**

**SELECT department\_name,**

**d.department\_id,**

**first\_name,**

**last\_name,**

**job\_id,**

**salary**

**FROM departments d**

**join employees e**

**ON e.department\_id = d.department\_id**

**WHERE job\_id = 'ST\_CLERK'**

**AND salary > 3200;**

**BEGIN**

**FOR emp\_rec IN emp\_cur\_detail LOOP**

**dbms\_output.Put\_line('Name: '**

**||emp\_rec.first\_name**

**||' '**

**||emp\_rec.last\_name**

**||Chr(9)**

**||'Department Name: '**

**||emp\_rec.department\_name**

**||Chr(9)**

**||'Department ID: '**

**||emp\_rec.department\_id**

**||Chr(9)**

**||'Job ID: '**

**||emp\_rec.job\_id**

**||Chr(9)**

**||'Salary: '**

**||emp\_rec.salary);**

**END LOOP;**

**END;**

**/**

15) Write a program in PL/SQL to FETCH multiple records and more than one columns from different tables.

**DECLARE**

**CURSOR cur\_emp\_name IS**

**SELECT first\_name,**

**last\_name,**

**department\_name**

**FROM employees e,**

**departments d**

**WHERE d.department\_id = e.department\_id;**

**v\_emp\_rec cur\_emp\_name%ROWTYPE;**

**BEGIN**

**OPEN cur\_emp\_name;**

**LOOP**

**FETCH cur\_emp\_name INTO v\_emp\_rec;**

**exit WHEN cur\_emp\_name%NOTFOUND;**

**dbms\_output.Put\_line('Name: '**

**|| v\_emp\_rec.first\_name**

**|| ' '**

**||v\_emp\_rec.last\_name**

**|| ' :: department: '**

**|| v\_emp\_rec.department\_name);**

**END LOOP;**

**CLOSE cur\_emp\_name;**

**END;**

**/**