ASSIGNMENT NO: PC – CS 692 / 09

1) Write a PL / SQL code block to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius and the corresponding values of calculated area in an empty table named AREAS with radius and area as attributes.

```
SQL> SET SERVEROUTPUT ON
SQL> DECLARE
  2  pi constant number(4, 2) := 3.14 ;
  3  radius number(5);
  4  area number(14, 2);
  5  BEGIN
  6  radius:= 3;
  7  WHILE RADIUS <= 7
  8  LOOP
  9  area := pi * power(radius, 2);
 10  INSERT INTO areas VALUES (radius, area);
 11  radius := radius + 1;
 12  END LOOP;
 13
 14  END;
 15  /

PL/SQL procedure successfully completed.

SQL> select * from areas;

    RADIUS        AREA
---------- ----------
         3       28.26
         4       50.24
         5        78.5
         6      113.04
         7      153.86
```

2) Write PL / SQL code to calculate sum of digits of a number.

```
SQL> SET SERVEROUTPUT ON
SQL> DECLARE
  2  given_number number(8);
  3  sum_of_digit number(8):=0;
  4  rem number(8);
  5  BEGIN
  6  given_number:= &given_number ;
  7  while given_number>0
  8  LOOP
  9  rem:=mod (given_number,10);
 10  sum_of_digit:=sum_of_digit+rem;
 11  given_number:=trunc(given_number / 10);
 12  END LOOP;
 13  dbms_output.put_line('The sum of digit is: '||sum_of_digit);
 14  END;
 15  /
Enter value for given_number: 1252
old   6: given_number:= &given_number ;
new   6: given_number:= 1252 ;
The sum of digit is: 10

PL/SQL procedure successfully completed.
```

3) Write PL / SQL code to calculate sum of natural series.

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2    term NUMBER(10) := &input_term;
  3    sum1 NUMBER(9);
  4  BEGIN
  5    sum1 := 0;
  6    FOR i IN 1..term LOOP
  7      sum1 := sum1 + i;
  8    END LOOP;
  9    dbms_output.put_line('sum = ' || sum1);
 10  END;
 11  /
Enter value for input_term: 121
old   2:   term NUMBER(10) := &input_term;
new   2:   term NUMBER(10) := 121;
sum = 7381

PL/SQL procedure successfully completed.
```

4)Write PL / SQL code for inverting a number 8975 to 5798.

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2    given_number varchar(5) := '8975';
  3    str_length number(2);
  4    inverted_number varchar(5) := ''; -- Initialize as empty string
  5  BEGIN
  6    str_length := length(given_number);
  7    FOR cntr IN REVERSE 1..str_length  -- Fixed: changed 'l' to '1' and 'I' to '1'
  8    LOOP
  9      inverted_number := inverted_number || substr(given_number, cntr, 1);
 10    END LOOP;
 11    dbms_output.put_line('The Given number is ' || given_number);
 12    dbms_output.put_line('The Inverted number is ' || inverted_number);
 13  END;
 14  /
The Given number is 8975
The Inverted number is 5798

PL/SQL procedure successfully completed.
```

5) Write PL/SQL code to find Factorial of first 10 Prime Numbers.

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2      num         NUMBER := 2;
  3      prime_count NUMBER := 0;
  4      is_prime    BOOLEAN;
  5      factorial   NUMBER;
  6  BEGIN
  7      WHILE prime_count < 10 LOOP
  8          is_prime := TRUE;
  9
 10          FOR i IN 2 .. TRUNC(SQRT(num)) LOOP
 11              IF MOD(num, i) = 0 THEN
 12                  is_prime := FALSE;
 13                  EXIT;
 14              END IF;
 15          END LOOP;
 16
 17          IF is_prime THEN
 18              factorial := 1;
 19              FOR j IN 1 .. num LOOP
 20                  factorial := factorial * j;
 21              END LOOP;
 22
 23              DBMS_OUTPUT.PUT_LINE('Prime: ' || num || ' -> Factorial: ' || factorial);
 24              prime_count := prime_count + 1;
 25          END IF;
 26
 27          num := num + 1;
 28      END LOOP;
 29  END;
 30  /
Prime: 2 -> Factorial: 2
Prime: 3 -> Factorial: 6
Prime: 5 -> Factorial: 120
Prime: 7 -> Factorial: 5040
Prime: 11 -> Factorial: 39916800
Prime: 13 -> Factorial: 6227020800
Prime: 17 -> Factorial: 355687428096000
Prime: 19 -> Factorial: 121645100408832000
Prime: 23 -> Factorial: 25852016738884976640000
Prime: 29 -> Factorial: 8841761993739701954543616000000

PL/SQL procedure successfully completed.
```

6) Write PL/SQL code to find a user input number is Petersen Number or not.

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2      given_number NUMBER(8);
  3      original_number NUMBER(8);
  4      sum_of_factorials NUMBER(8) := 0;
  5      digit NUMBER(8);
  6      factorial NUMBER(8);
  7
  8      FUNCTION calc_factorial(n IN NUMBER) RETURN NUMBER IS
  9          result NUMBER := 1;
 10      BEGIN
 11          FOR i IN 1..n LOOP
 12              result := result * i;
 13          END LOOP;
 14          RETURN result;
 15      END;
 16
 17  BEGIN
 18      given_number := &given_number;
 19      original_number := given_number;
 20
 21      WHILE given_number > 0 LOOP
 22          digit := MOD(given_number, 10);
 23          factorial := calc_factorial(digit);
 24          sum_of_factorials := sum_of_factorials + factorial;
 25          given_number := TRUNC(given_number / 10);
 26      END LOOP;
 27
 28      IF sum_of_factorials = original_number THEN
 29          DBMS_OUTPUT.PUT_LINE(original_number || ' is a Peterson Number.');
 30      ELSE
 31          DBMS_OUTPUT.PUT_LINE(original_number || ' is NOT a Peterson Number.');
 32      END IF;
 33  END;
 34  /
Enter value for given_number: 145
old  18:     given_number := &given_number;
new  18:     given_number := 145;
145 is a Peterson Number.

PL/SQL procedure successfully completed.
```