ASSIGNMENT - 3

Ques 1 : Write a standalone procedure to compute and display the sum of odd number from the 1 to n.

CODE :

```
CREATE OR REPLACE PROCEDURE SUM_ODD(n in number, sm out number)

AS

BEGIN

   for i in 1..n
   LOOP
      IF mod(i, 2) <> 0 THEN
         sm := sm + i;
      END IF;
   END LOOP;
END;
/
```

OUTPUT :

```
Procedure created.
```

Ques 2 : Write a procedure to display the number employee who are living in ‘Chennai’.

CODE :

```
CREATE TABLE emp
(sno number(2), emp_name varchar(20), city varchar(20));

INSERT INTO emp
VALUES (1, 'Ramesh', 'Delhi');
INSERT INTO emp
VALUES (2, 'Kaushik', 'Mumbai');
INSERT INTO emp
VALUES (3, 'Siman', 'Chennai');
INSERT INTO emp
VALUES (4, 'Sanju', 'Gujrat');
INSERT INTO emp
VALUES (5, 'Ram', 'Kanpur');
INSERT INTO emp
VALUES (6, 'Yash', 'Chennai');
INSERT INTO emp
VALUES (7, 'Sukhweer', 'Patna');
INSERT INTO emp
VALUES (8, 'Balvinder', 'Chennai');


CREATE OR REPLACE PROCEDURE COUNT_NUM (all_emp OUT number)

AS

BEGIN

   SELECT COUNT(*) into all_emp FROM emp WHERE city = 'Chennai';

END;


DECLARE
   total number;

BEGIN
   COUNT_NUM(total);
   dbms_output.put_line('Total employee lives in Chennai are ' || total);
END;
```

OUTPUT :

```
Table created.

1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.

Procedure created.

Statement processed.
Total employee lives in Chennai are 3
```

Ques 3 : Write a function to compute the first 10 Fibonacci numbers.

CODE :

```
DECLARE
   n number := 8;
   first number := 0;
   last number := 1;
   i number;
   mid number;
   a number;

FUNCTION fibonacci(n number)
RETURN number
IS
   mid number;

BEGIN
 for i in 1..n
   LOOP
      mid := first + last;
      dbms_output.put(' '||mid);
      first := last;
      last := mid;
   END LOOP;
   dbms_output.new_line;
  RETURN mid;
END;

BEGIN

   dbms_output.put('Fibonacci Series : '||first||' '||last);
   a := fibonacci(n);

END;
/
```

OUTPUT:

```
Statement processed.
Fibonacci Series : 0 1 1 2 3 5 8 13 21 34
```

Ques 4 : Write a function find the student name with maximum CGPA.

CODE :

```
CREATE TABLE student
(rno number(2), student_name varchar(20), cgpa number(2));

INSERT INTO student
VALUES (1, 'Ramesh', 8);
INSERT INTO student
VALUES (2, 'Kaushik', 7);
INSERT INTO student
VALUES (3, 'Siman', 6);
INSERT INTO student
VALUES (4, 'Sanju', 8);
INSERT INTO student
VALUES (5, 'Ram', 7);
INSERT INTO student
VALUES (6, 'Yash', 9);
INSERT INTO student
VALUES (7, 'Sukhweer', 6);
INSERT INTO student
VALUES (8, 'Balvinder', 7);


CREATE OR REPLACE FUNCTION MAX_MARKS
RETURN varchar
AS
   name varchar(20);

BEGIN

   SELECT student_name into name FROM student WHERE cgpa = (SELECT MAX(cgpa)
FROM student);
   RETURN name;

END;


DECLARE
   name varchar(20);

BEGIN
   name := MAX_MARKS();
   dbms_output.put_line('Student Name with Maximum CGPA : ' || name);
END;
```

OUTPUT:

```
Table created.

1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.

Function created.

Statement processed.
Student Name with Maximum CGPA : Yash
```

Ques 5 : Write a function to find the sum of the given series upto ‘n’ terms
0 ,1, 1, 4, 9, 25 ,64 ,…….

CODE :

```
DECLARE
   n number := 5;
   first number := 0;
   last number := 1;
   i number;
   mid number;
   b number;

FUNCTION fun(n number)
RETURN number
IS
   s number := (first**2) + (last**2);

BEGIN
 for i in 1..n
   LOOP
      mid := first + last;
      s := s + (mid*mid);
      first := last;
      last := mid;
   END LOOP;
  RETURN s;
END;

BEGIN

   b := fun(n);
   dbms_output.put_line('Sum of Series : '||b);

END;
/
```

OUTPUT :

```
Statement processed.
Sum of Series : 104
```