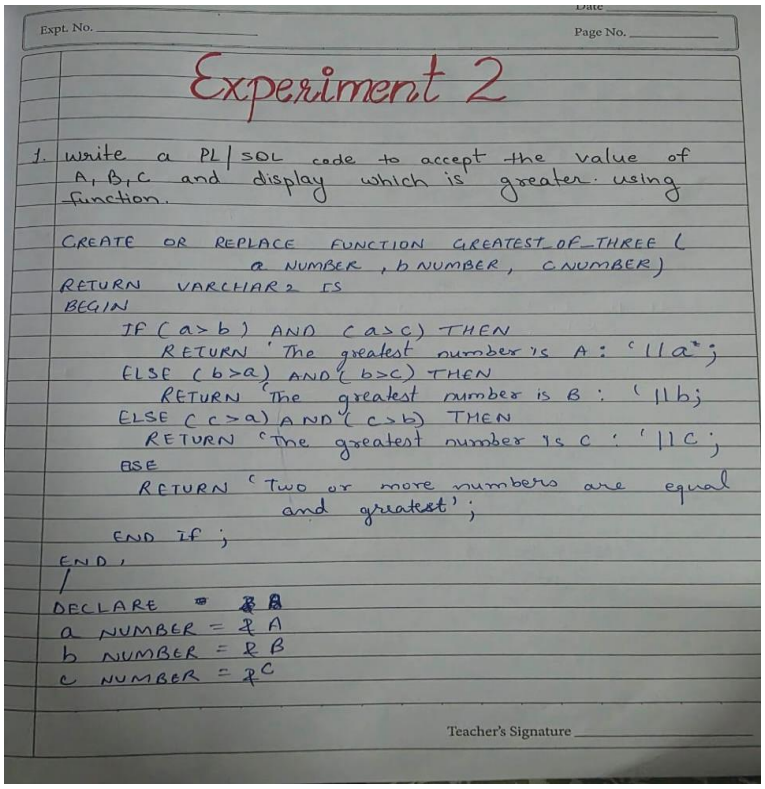
EXPERIMENT 2

1. Write a PL/SQL code to accept the value of A, B & C display which is greater using function and procedure.



Expt. No. Page No.

Experiment 2

1. write a PL/SQL code to accept the value of A, B, C and display which is greater using function.

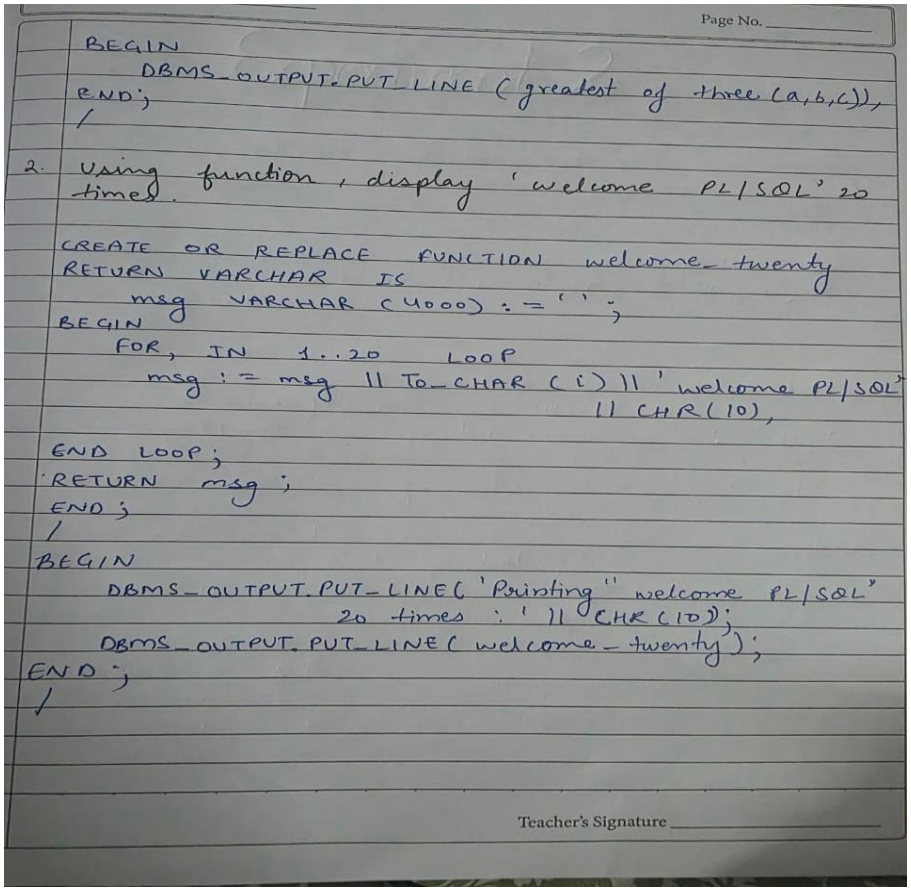
```
CREATE OR REPLACE FUNCTION GREATEST_OF_THREE (
        a NUMBER , b NUMBER , C NUMBER)
RETURN VARCHAR2 IS
BEGIN
    IF (a>b) AND (a>c) THEN
      RETURN 'The greatest number is A : '||a;
    ELSE (b>a) AND (b>c) THEN
      RETURN 'The greatest number is B : '||b;
    ELSE (c>a) AND (c>b) THEN
      RETURN 'The greatest number is C : '||C;
    ELSE
      RETURN 'Two or more numbers are equal
              and greatest';
    END IF;
END;
/
DECLARE
a NUMBER = &A
b NUMBER = &B
c NUMBER = &C
```

Teacher's Signature

```plsql
-- Function to return greatest among 3 numbers
CREATE OR REPLACE FUNCTION find_max(a IN NUMBER, b IN NUMBER, c IN NUMBER)
RETURN NUMBER
IS
    greatest NUMBER;
BEGIN
    IF a >= b AND a >= c THEN
        greatest := a;
    ELSIF b >= a AND b >= c THEN
        greatest := b;
    ELSE
        greatest := c;
    END IF;
    RETURN greatest;
END;
/
-----------------------------------------------------
```

```
18
19 -- Procedure to accept input values and display result
20 CREATE OR REPLACE PROCEDURE show_max(a IN NUMBER, b IN NUMBER, c IN NUMBER)
21 IS
22     result NUMBER;
23 BEGIN
24     result := find_max(a, b, c);
25     DBMS_OUTPUT.PUT_LINE('The greatest number is: ' || result);
26 END;
27 /
28 ------------------------------------------------------
29
30 -- Run the procedure (example)
31 BEGIN
32     show_max(10, 25, 15);
33 END;
34 /
35
```

2. Using PL/SQL Statements create a simple loop that display message “Welcome to PL/SQL Programming” 20 times using function and procedure



Page No.

```
BEGIN
    DBMS_OUTPUT.PUT_LINE (greatest of three (a,b,c));
END;
/
```

2. Using function, display 'welcome PL/SQL' 20 times.

```
CREATE OR REPLACE FUNCTION welcome_twenty
RETURN VARCHAR IS
    msg VARCHAR (4000) := '';
BEGIN
    FOR, IN 1..20 LOOP
    msg := msg || TO_CHAR (i) || 'welcome PL/SQL'
                || CHR(10);

END LOOP;
RETURN msg;
END;
/
BEGIN
    DBMS_OUTPUT.PUT_LINE('Printing '' welcome PL/SQL''
                20 times :' || CHR(10));
    DBMS_OUTPUT.PUT_LINE( welcome_twenty);
END;
/
```

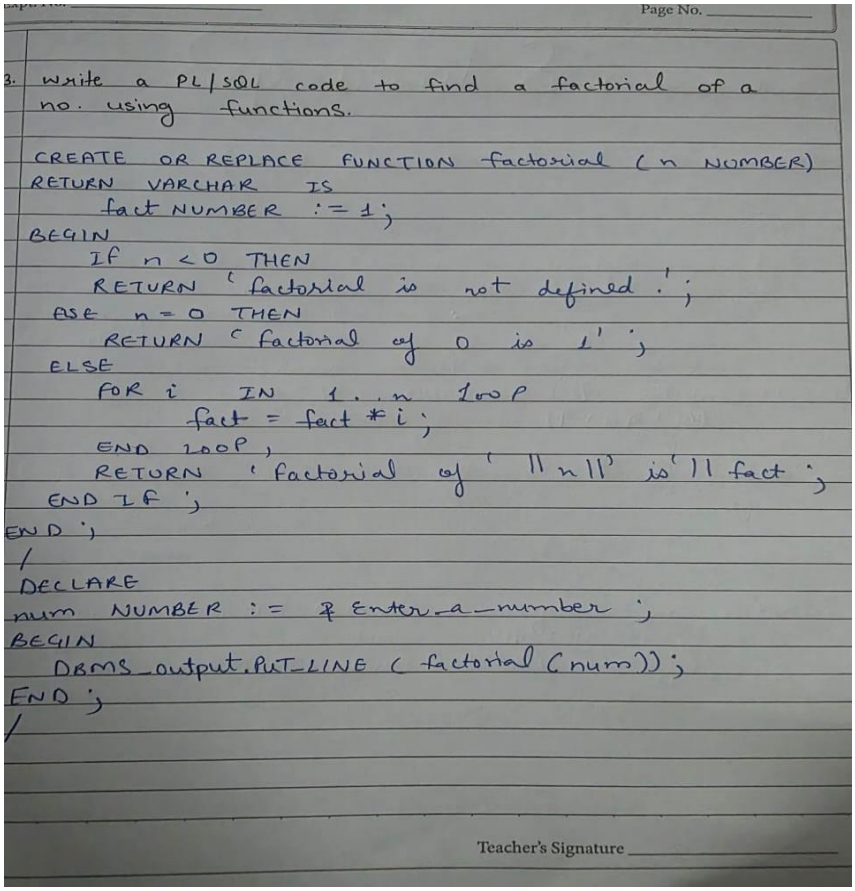
Teacher's Signature

```
SET SERVEROUTPUT ON;

-----------------------------------------------------
-- Function to return the welcome message
-----------------------------------------------------
CREATE OR REPLACE FUNCTION get_message
RETURN VARCHAR2
IS
BEGIN
   RETURN 'Welcome to PL/SQL Programming';
END;
/
-----------------------------------------------------
-- Procedure to display the message 20 times
-----------------------------------------------------
CREATE OR REPLACE PROCEDURE show_message
IS
   i NUMBER := 1;
```

```
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
```

3. Write a PL/SQL code block to find the factorial of a number using function and procedure.



```
3. write a PL/SQL code to find a factorial of a
no. using functions.

CREATE OR REPLACE FUNCTION factorial (n NUMBER)
RETURN VARCHAR IS
    fact NUMBER := 1;
BEGIN
    If n < 0 THEN
    RETURN 'factorial is not defined.';
ESE n = 0 THEN
    RETURN 'factorial of 0 is 1';
ELSE
    FOR i IN 1..n loop
        fact = fact * i;
    END loop;
    RETURN 'factorial of ' ||n||' is' || fact;
END IF;
END;
/
DECLARE
num NUMBER := &Enter_a_number;
BEGIN
   DBMS_output.PUT_LINE (factorial (num));
END;
/
```

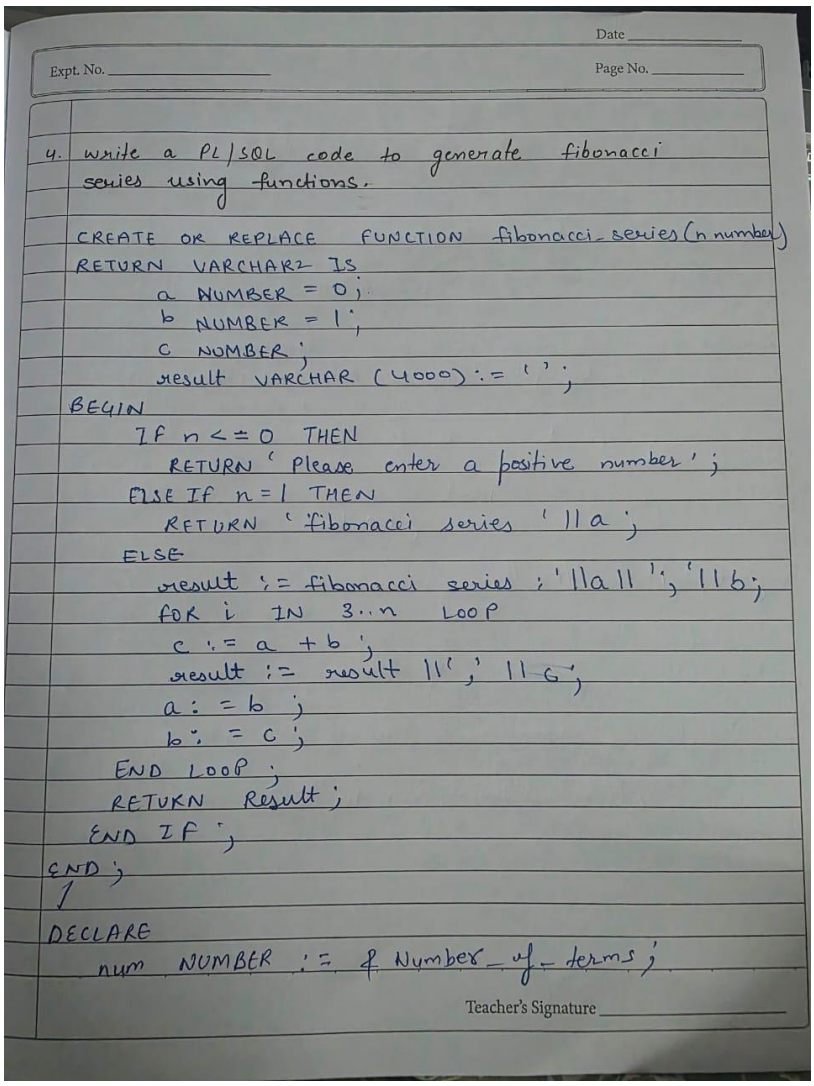
Teacher's Signature

```sql
-- Function to find factorial
CREATE OR REPLACE FUNCTION find_factorial(n IN NUMBER)
RETURN NUMBER
IS
    fact NUMBER := 1;
BEGIN
    IF n < 0 THEN
        RETURN NULL;  -- factorial not defined for negative numbers
    ELSIF n = 0 THEN
        RETURN 1;     -- factorial of 0 is 1
    ELSE
        FOR i IN 1..n LOOP
            fact := fact * i;
        END LOOP;
        RETURN fact;
    END IF;
END;
/
```

```
20
21 -- Procedure to display factorial
22 CREATE OR REPLACE PROCEDURE show_factorial(n IN NUMBER)
23 IS
24     result NUMBER;
25 BEGIN
26     result := find_factorial(n);
27
28     IF result IS NULL THEN
29         DBMS_OUTPUT.PUT_LINE('Factorial not defined for negative numbers.');
30     ELSE
31         DBMS_OUTPUT.PUT_LINE('Factorial of ' || n || ' is ' || result);
32     END IF;
33 END;
34 /
```

```
Factorial of 5 is 120
```

4. Write a PL/SQL program to generate Fibonacci series using function and procedure.



Date _____

Expt. No. _____ Page No. _____

4. write a PL/SQL code to generate fibonacci series using functions.

```
CREATE OR REPLACE FUNCTION fibonacci_series(n number)
RETURN VARCHAR2 IS
    a NUMBER := 0;
    b NUMBER := 1;
    c NUMBER;
    result VARCHAR(4000) := '';
BEGIN
    IF n <= 0 THEN
      RETURN 'Please enter a positive number';
    ELSE IF n = 1 THEN
      RETURN 'fibonacci series ' || a;
    ELSE
      result := fibonacci series : ' || a || ' ' || b;
      FOR i IN 3..n LOOP
      c := a + b;
      result := result || ' ' || c;
      a := b;
      b := c;
    END LOOP;
    RETURN Result;
  END IF;
END;
/
DECLARE
   num NUMBER := &Number_of_terms;
```

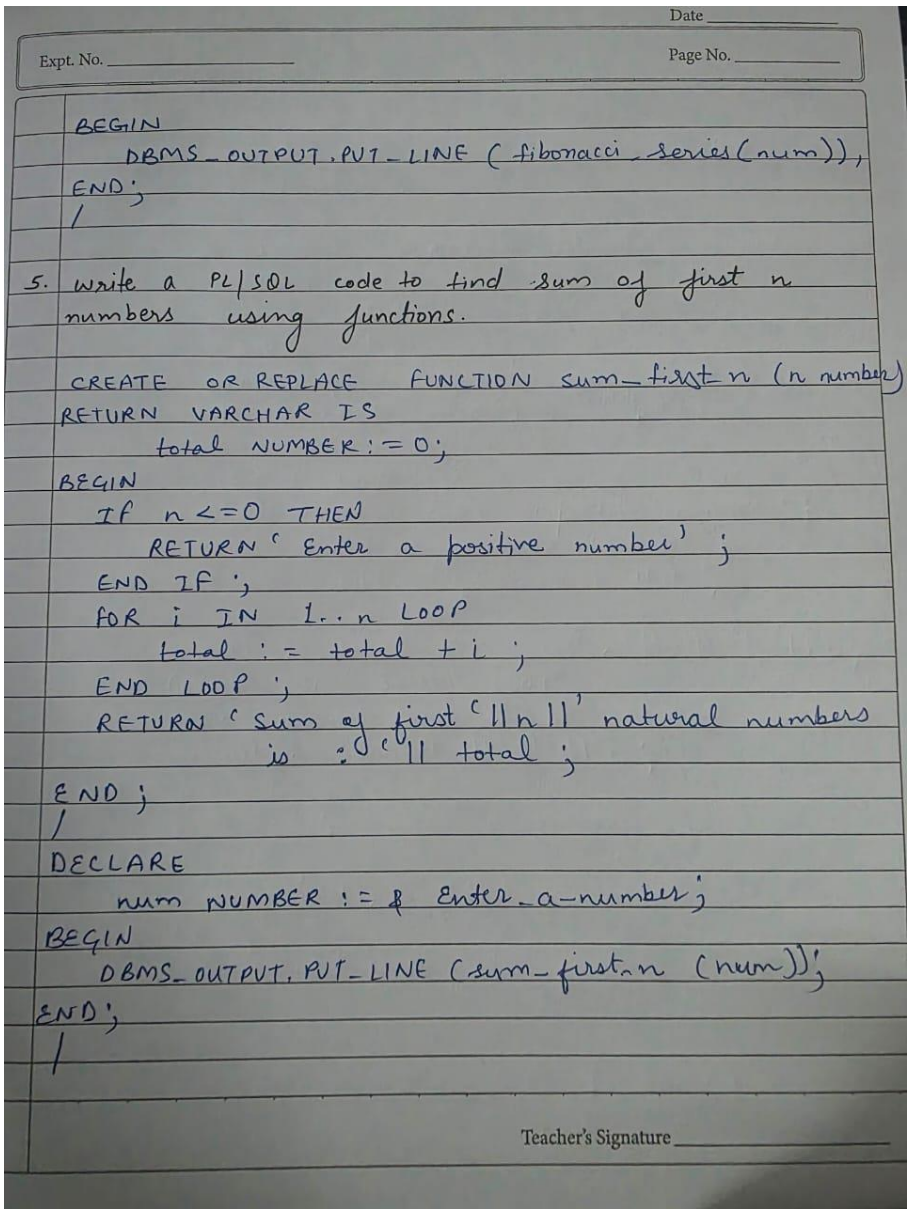
Teacher's Signature _____

```
      FOR i IN 2..n LOOP
         temp := a + b;
         a := b;
         b := temp;
      END LOOP;
      RETURN b;
   END IF;
END;
/
-------------------------------------------------------
-- Procedure to display Fibonacci series
-------------------------------------------------------
CREATE OR REPLACE PROCEDURE show_fib(num IN NUMBER)
IS
BEGIN
   FOR i IN 0..num LOOP
      DBMS_OUTPUT.PUT_LINE('Fibonacci(' || i || ') = ' || fib(i));
   END LOOP;
```

```
-------------------------------------------------------
CREATE OR REPLACE PROCEDURE show_fib(num IN NUMBER)
IS
BEGIN
   FOR i IN 0..num LOOP
      DBMS_OUTPUT.PUT_LINE('Fibonacci(' || i || ') = ' || fib(i));
   END LOOP;
END;
/
-------------------------------------------------------
-- Anonymous block to execute
-------------------------------------------------------
BEGIN
   show_fib(10);   -- Change 10 to how many terms you want
END;
/
```

```
Fibonacci(1) = 1
Fibonacci(2) = 1
Fibonacci(3) = 2
Fibonacci(4) = 3
Fibonacci(5) = 5
Fibonacci(6) = 8
Fibonacci(7) = 13
Fibonacci(8) = 21
Fibonacci(9) = 34
Fibonacci(10) = 55
```

5. Write a PL/SQL code to fund the sum of first N numbers using function and procedure.



Date

Expt. No. Page No.

```
BEGIN
   DBMS_OUTPUT.PUT_LINE (fibonacci_series(num));
END;
/
```

5. write a PL/SQL code to find sum of first n numbers using functions.

```
CREATE OR REPLACE FUNCTION sum_first_n (n number)
RETURN VARCHAR IS
      total NUMBER := 0;
BEGIN
   IF n <=0 THEN
      RETURN 'Enter a positive number';
   END IF;
   FOR i IN 1..n LOOP
      total := total + i;
   END LOOP;
   RETURN 'Sum of first '||n||' natural numbers
         is :'||total;
END;
/
DECLARE
   num NUMBER := &Enter_a_number;
BEGIN
   DBMS_OUTPUT.PUT_LINE (sum_first_n (num));
END;
/
```

Teacher's Signature

```sql
-- Function to calculate the sum of first n numbers
CREATE OR REPLACE FUNCTION sum_first_n(n IN NUMBER) RETURN NUMBER IS
    total_sum NUMBER := 0;
BEGIN
    FOR i IN 1..n LOOP
        total_sum := total_sum + i;
    END LOOP;
    RETURN total_sum;
END;
/

-- Procedure to display the sum
CREATE OR REPLACE PROCEDURE display_sum(n IN NUMBER) IS
    result NUMBER;
BEGIN
    result := sum_first_n(n);
    DBMS_OUTPUT.PUT_LINE('Sum of first ' || n || ' numbers is: ' || result);
END;
```

```
Sum of first 10 numbers is: 55
```