**7. PL/SQL Program To Add Two Numbers (All PL/SQL program: using function or procedure )**

CREATE OR REPLACE FUNCTION Addnumber(

num1 IN NUMBER,

num2 IN NUMBER

) RETURN NUMBER IS

result NUMBER;

BEGIN

result := num1 + num2;

RETURN result;

END;

DECLARE

result NUMBER;

BEGIN

result := Addnumber(100,10);

DBMS\_OUTPUT.PUT\_LINE('THE ADDITION OF TWO NO IS ' || result);

END;

**8. PL/SQL Program for Prime Number**

CREATE OR REPLACE FUNCTION Primeno(

num IN NUMBER

) RETURN BOOLEAN IS

i NUMBER;

BEGIN

IF num <= 1 THEN

RETURN FALSE;

END IF ;

FOR i IN 2..TRUNC(num/2) LOOP

IF MOD(num,i) = 0 THEN

RETURN FALSE;

END IF;

END LOOP;

RETURN TRUE;

END;

DECLARE

num NUMBER := 23;

result BOOLEAN ;

BEGIN

result := Primeno(num);

IF result THEN

DBMS\_OUTPUT.PUT\_LINE(num || ' is a prime no ');

ELSE

DBMS\_OUTPUT.PUT\_LINE(num || ' is not a prime no');

END IF;

END;

**9. PL/SQL Program to Find Factorial of a Number**

CREATE OR REPLACE FUNCTION Fact(

num IN NUMBER

)RETURN NUMBER IS

fact NUMBER := 1;

i NUMBER;

BEGIN

IF num < 0 THEN

DBMS\_OUTPUT.PUT\_LINE(' THIS IS A NEGETIVE NO WHOSE FACTORIAL DOESNT EXISTS ');

END IF;

FOR i in 1..num LOOP

fact := fact \*i;

END LOOP;

RETURN fact;

END;

DECLARE

result NUMBER;

num NUMBER := 5;

BEGIN

result := Fact(num);

DBMS\_OUTPUT.PUT\_LINE('THE FACTORIAL OF' || num || ' is ' || result);

END;

CREATE OR REPLACE FUNCTION Reverse(

num IN NUMBER

) RETURN NUMBER IS

reverse NUMBER := 0;

remender NUMBER;

BEGIN

IF num < 0 THEN

num := ABS(num);

END IF;

WHILE num > 0 LOOP

remender := MOD(num,10);

reverse := (reverse \* 10 ) + remender;

num := num /10;

END LOOP;

RETURN reverse ;

END;

DECLARE

num NUMBER := 654321;

reverse NUMBER;

BEGIN

reverse := Reverse(num);

DBMS\_OUTPUT.PUT\_LINE(' THE REVRESE OF THE NO ' || num || ' is' || reverse);

END;

**10. PL/SQL Program for Reverse of a Number**

CREATE OR REPLACE FUNCTION Rev(

num IN NUMBER

)RETURN VARCHAR2 IS

reversed VARCHAR2(4000) := '';

original VARCHAR2(4000);

BEGIN

original := TO\_CHAR(num);

FOR i IN REVERSE 1..LENGTH(original) LOOP

reversed := reversed || SUBSTR(original,I,1);

END LOOP;

RETURN reversed;

END;

DECLARE

num NUMBER := 654321;

result VARCHAR2(4000);

BEGIN

result := Rev(num);

DBMS\_OUTPUT.PUT\_LINE('THE REVERSED NUMBER IS ' || result);

END;

**11. PL/SQL Program for Fibonacci Series**

CREATE OR REPLACE FUNCTION Fiboseries(

n IN NUMBER

)RETURN VARCHAR2 IS

fibosequence VARCHAR2(4000):= '';

num1 NUMBER:=0;

num2 NUMBER:=1;

next NUMBER;

BEGIN

IF n <=0 THEN

RETURN 'INVALID INPUT PLEASE PROVIDE POSITIVE NUMBER';

END IF;

IF n >=1 THEN

fibosequence := fibosequence || num1;

END IF;

IF n >= 2 THEN

fibosequence := fibosequence ||' , ' || num2;

END IF;

FOR i IN 3..(n) LOOP

next := num1 + num2;

fibosequence := fibosequence || ' , ' || next;

num1 := num2;

num2 := next;

END LOOP;

RETURN fibosequence;

END;

DECLARE

n NUMBER:= 8;

result VARCHAR2(4000);

BEGIN

result := Fiboseries(n);

DBMS\_OUTPUT.PUT\_LINE(' THE FIBONACHI SERIES IS ' || result);

END;

**12. PL/SQL Program to Check Number is Odd or Even**

CREATE OR REPLACE FUNCTION Checkoe(

num IN NUMBER

)RETURN VARCHAR IS

BEGIN

IF MOD(num,2)=0 THEN

RETURN 'EVEN';

ELSE

RETURN 'ODD';

END IF;

END;

DECLARE

num NUMBER := 50;

result VARCHAR2(4000);

BEGIN

result := Checkoe(num);

DBMS\_OUTPUT.PUT\_LINE(' THE GIVEN NO ' || num || ' IS ' || result);

END;

**13. PL/SQL Program to Reverse a String**

CREATE OR REPLACE FUNCTION Revstr(

str IN VARCHAR2

) RETURN VARCHAR IS

reversedstr VARCHAR2(4000);

BEGIN

IF str IS NULL THEN

RETURN NULL;

END IF;

FOR i IN REVERSE 1..LENGTH(str) LOOP

reversedstr := reversedstr || SUBSTR(str,i,1);

END LOOP;

RETURN reversedstr;

END;

DECLARE

str VARCHAR2(4000) := 'DATABASE';

result VARCHAR2(4000);

BEGIN

result := Revstr(str);

DBMS\_OUTPUT.PUT\_LINE('THE REVERSED STRING OF '|| str ||' IS ' || result);

END;

**14. Pl/SQL Program for Palindrome Number**

CREATE OR REPLACE FUNCTION Ispalindrome(

num IN NUMBER

)RETURN BOOLEAN IS

reversedno VARCHAR2(100);

originalno VARCHAR2(100) ;

BEGIN

originalno := TO\_CHAR(num);

reversedno := '';

FOR i IN REVERSE 1..length(originalno) LOOP

reversedno := reversedno ||SUBSTR(originalno,i,1);

END LOOP;

RETURN originalno = reversedno;

END;

DECLARE

num NUMBER := 1281;

result BOOLEAN;

BEGIN

result := Ispalindrome(num);

IF result THEN

DBMS\_OUTPUT.PUT\_LINE(' THE GIVEN NUMBER IS A PALINDROME');

ELSE

DBMS\_OUTPUT.PUT\_LINE(' THE GIVEN NUMBER IS NOT A PALINDROME ');

END IF;

END;

**15. PL/SQL Program to Swap two Numbers**

CREATE OR REPLACE PROCEDURE SwapNumbers(

num1 IN OUT NUMBER,

num2 IN OUT NUMBER

)

IS

temp NUMBER;

BEGIN

temp := num1;

num1 := num2;

num2 := temp;

DBMS\_OUTPUT.PUT\_LINE('After swapping - Number 1: ' || num1 || ', Number 2: ' || num2);

END;

DECLARE

number1 NUMBER := 10;

number2 NUMBER := 20;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Before swapping - Number 1: ' || number1 || ', Number 2: ' || number2);

SwapNumbers(number1, number2);

END;

**17. PL/SQL Program to Find Greatest of Three Numbers**

CREATE OR REPLACE FUNCTION Great(

num1 IN NUMBER,

num2 IN NUMBER,

num3 IN NUMBER

) RETURN NUMBER IS

result NUMBER;

BEGIN

result := GREATEST(num1,num2,num3);

RETURN result;

END;

DECLARE

num1 NUMBER := 45;

num2 NUMBER := 70;

num3 NUMBER := 25;

result NUMBER;

BEGIN

result := Great(num1,num2,num3);

DBMS\_OUTPUT.PUT\_LINE(' THE GREATEST NUMBER BETWEEN THESE 3 NUMBERS IS ' || result);

END;

**18. PL/SQL procedure to check whether the current day is a weekend or weekday**

CREATE OR REPLACE PROCEDURE CheckWeekendOrWeekday IS

currentDay VARCHAR2(10);

BEGIN

-- Get the current day using the built-in function

currentDay := TO\_CHAR(SYSDATE, 'DAY');

-- Convert the day to uppercase for comparison

currentDay := UPPER(currentDay);

-- Check if the current day is a weekend or weekday

IF currentDay = 'SATURDAY' OR currentDay = 'SUNDAY' THEN

DBMS\_OUTPUT.PUT\_LINE('Today is a weekend!');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Today is a weekday.');

END IF;

END;

BEGIN

CheckWeekendOrWeekday;

END;