* **PL &SQL Array concept**

DECLARE

type namesarray IS VARRAY(5) OF VARCHAR2(10);

type grades IS VARRAY(5) OF INTEGER;

names namesarray;

marks grades;

total integer;

BEGIN

names := namesarray('Kavita', 'Pritam', 'Ayan', 'Rishav', 'Aziz');

marks:= grades(98, 97, 78, 87, 92);

total := names.count;

dbms\_output.put\_line('Total '|| total || ' Students');

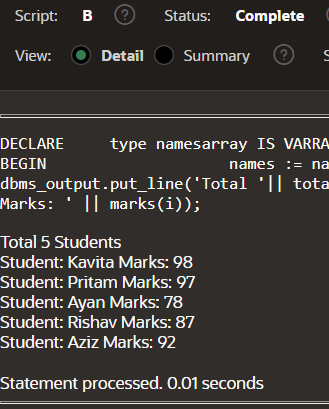
FOR i in 1 .. total LOOP

dbms\_output.put\_line('Student: ' || names(i) || '

Marks: ' || marks(i));

END LOOP;

END;



* **Count the even and odd**

DECLARE

v\_number NUMBER := 23146579;

v\_odd\_count NUMBER := 0;

v\_even\_count NUMBER := 0;

v\_digit NUMBER;

BEGIN

WHILE v\_number > 0 LOOP

v\_digit := MOD(v\_number, 10);

IF MOD(v\_digit, 2) = 0 THEN

v\_even\_count := v\_even\_count + 1;

ELSE

v\_odd\_count := v\_odd\_count + 1;

END IF;

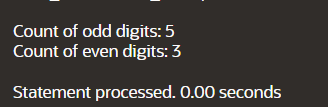
v\_number := TRUNC(v\_number / 10);

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Count of odd digits: ' || v\_odd\_count);

DBMS\_OUTPUT.PUT\_LINE('Count of even digits: ' || v\_even\_count);

END;



* **Find Minimum**

DECLARE

a number;

b number;

c number;

PROCEDURE findMin(x IN number, y IN number, z OUT number) IS

BEGIN

IF x < y THEN

z:= x;

ELSE

z:= y;

END IF;

END;

BEGIN

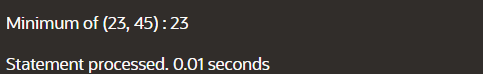
a:= 23;

b:= 45;

findMin(a, b, c);

dbms\_output.put\_line(' Minimum of (23, 45) : ' || c);

END;



* **MAXIMUM NUMBER**

DECLARE

a number;

b number;

c number;

FUNCTION findMax(x IN number, y IN number)

RETURN number

IS

z number;

BEGIN

IF x > y THEN

z:= x;

ELSE

Z:= y;

END IF;

RETURN z;

END;

BEGIN

a:= 23;

b:= 45;

c := findMax(a, b);

dbms\_output.put\_line(' Maximum of (23,45): ' || c);

END;



**FIBONACCI SERIES**

DECLARE

i NUMBER;

FUNCTION fibonacci(n NUMBER) RETURN NUMBER

IS

BEGIN

IF n <= 0 THEN

RETURN 0;

ELSIF n = 1 THEN

RETURN 1;

ELSE

RETURN fibonacci(n-1) + fibonacci(n-2);

END IF;

END fibonacci;

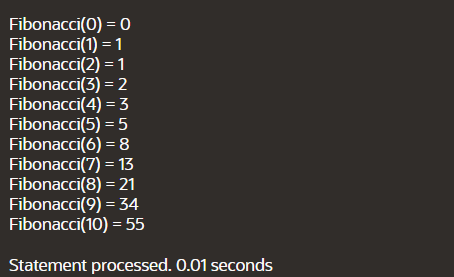
BEGIN

FOR i IN 0..10 LOOP

DBMS\_OUTPUT.PUT\_LINE('Fibonacci(' || i || ') = ' || fibonacci(i));

END LOOP;

END;



* **FACTORIAL USING RECURTION**

DECLARE

num number;

factorial number;

FUNCTION fact(x number)

RETURN number

IS

f number;

BEGIN

IF x=0 THEN

f := 1;

ELSE

f := x \* fact(x-1);

END IF;

RETURN f;

END;

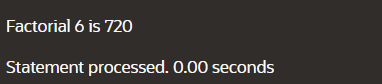
BEGIN

num:= 6;

factorial := fact(num);

dbms\_output.put\_line(' Factorial '|| num || ' is ' || factorial);

END;



* **FIBONACCI**

DECLARE

n1 NUMBER := 0;

n2 NUMBER := 1;

n3 NUMBER;

counter NUMBER := 2;

terms NUMBER := 10;

BEGIN

DBMS\_OUTPUT.PUT\_LINE(n1);

DBMS\_OUTPUT.PUT\_LINE(n2);

WHILE counter < terms LOOP

n3 := n1 + n2;

DBMS\_OUTPUT.PUT\_LINE(n3);

n1 := n2; -- Update n1 to the next number

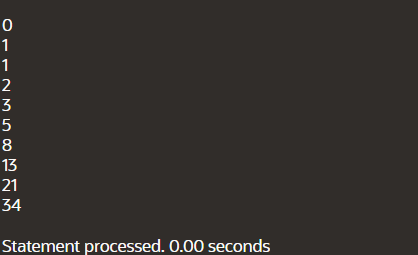
n2 := n3; -- Update n2 to the next number

counter := counter + 1;

END LOOP;

END;

/

]

**IMPLICITY**

DECLARE

total\_rows number(2);

BEGIN

UPDATE COP3

SET AGE = AGE + 5;

IF sql%notfound THEN

dbms\_output.put\_line('no customers selected');

ELSIF sql%found THEN

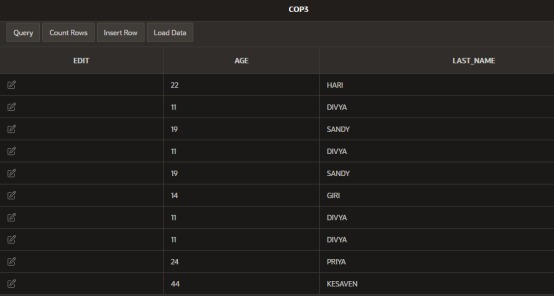
total\_rows := sql%rowcount;

dbms\_output.put\_line( total\_rows || ' customers selected ');

END IF;

END;

BEFORE TABLE:



AFTER TABLE:

