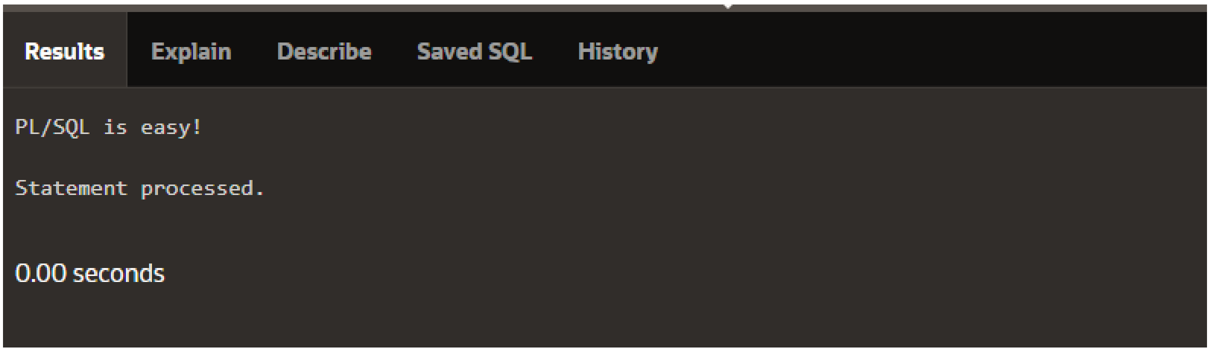
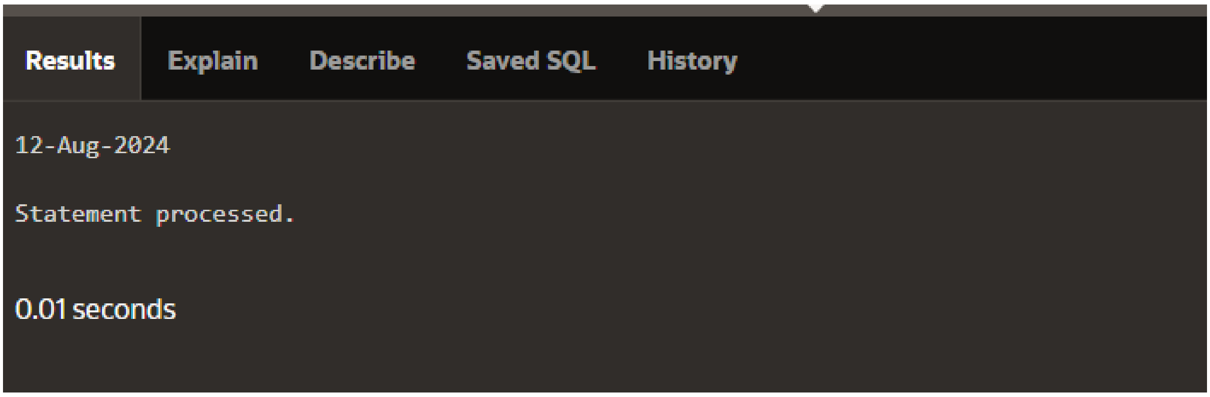
```
BEGIN
        DBMS_OUTPUT.PUT_LINE('PL/SQL is easy!');
END;
```



```
PL/SQL is easy!

Statement processed.

0.00 seconds
```

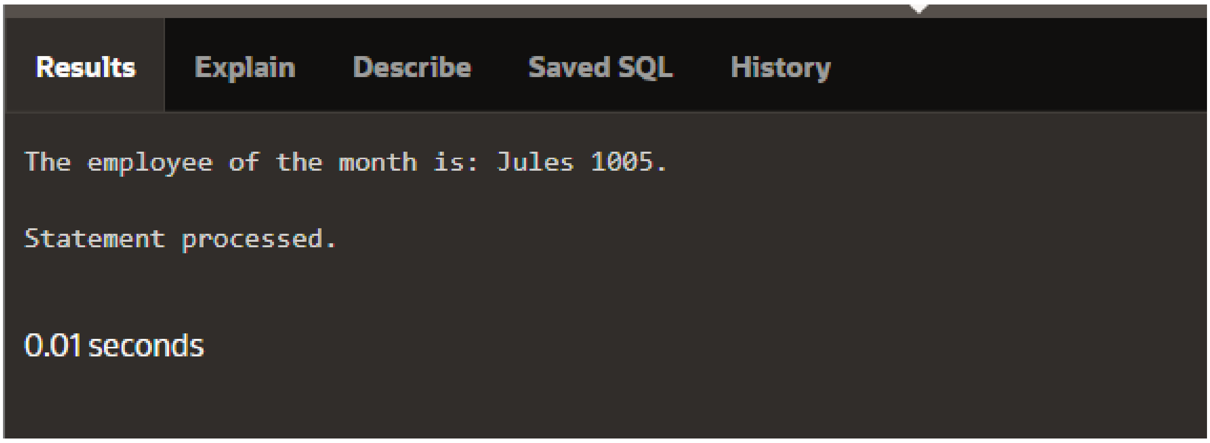
```
DECLARE
currentdate DATE := SYSDATE;
BEGIN
DBMS_OUTPUT.PUT_LINE(currentdate);
END;
```



```
12-Aug-2024

Statement processed.

0.01 seconds
```

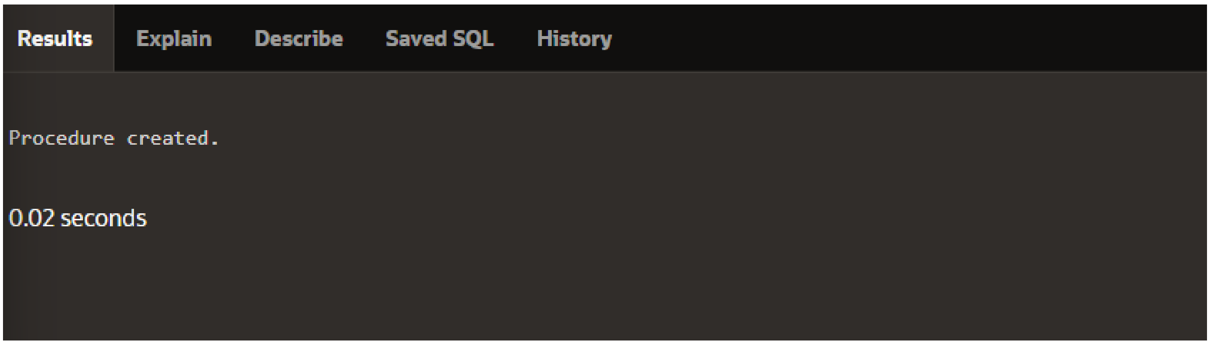
```
DECLARE
  v_first_name VARCHAR2(25);
  v_id INT;
BEGIN
  SELECT emp_name, emp_id
  INTO v_first_name, v_id
  FROM EMPLOYEE
  WHERE emp_name = 'Jules';
```

```
   DBMS_OUTPUT.PUT_LINE('The employee of the month is: '
       || v_first_name || ' ' || v_id || '.');
EXCEPTION
   WHEN TOO_MANY_ROWS THEN
       DBMS_OUTPUT.PUT_LINE('Your select statement retrieved '
           || 'multiple rows. Consider using a cursor or changing '
           || 'the search criteria.');
END;
```

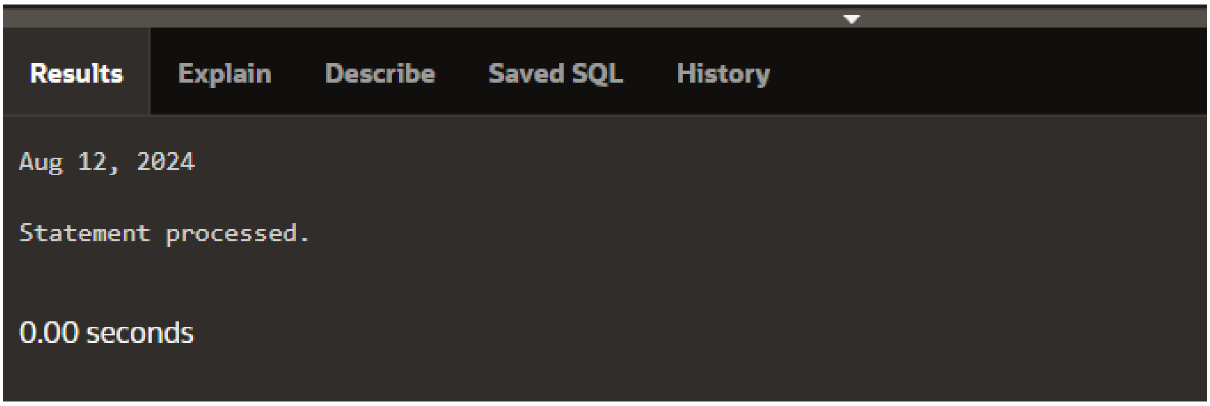


```
The employee of the month is: Jules 1005.

Statement processed.

0.01 seconds
```

```
CREATE OR REPLACE PROCEDURE print_date IS
v_date VARCHAR2(30);
BEGIN
SELECT TO_CHAR(SYSDATE,'Mon DD, YYYY')
INTO v_date
FROM DUAL;
DBMS_OUTPUT.PUT_LINE(v_date);
END;
BEGIN
            PRINT_DATE;
 END;
```

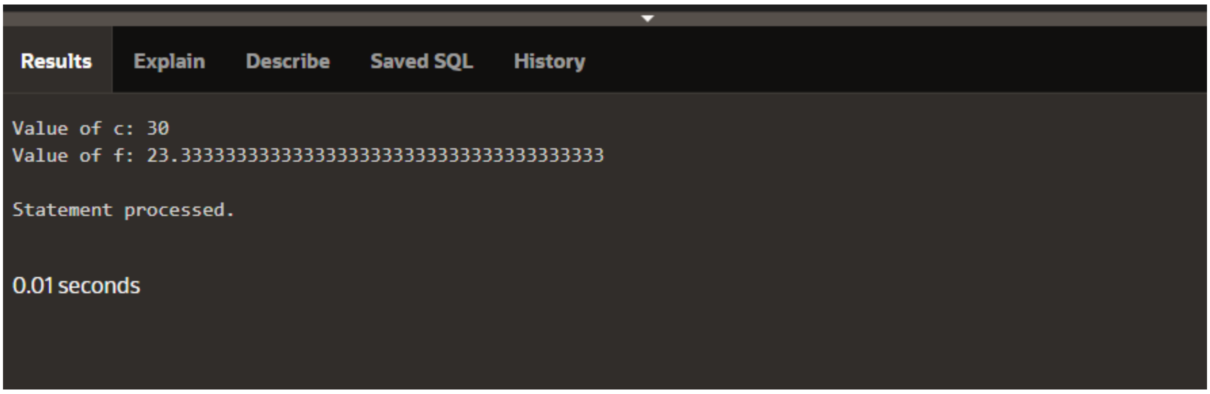


```
Procedure created.

0.02 seconds
```



```
Aug 12, 2024

Statement processed.

0.00 seconds
```

```
DECLARE
   a integer :=10;
   b integer :=20;
   c integer;
   f real;
BEGIN
   c:= a+b;
   dbms_output.put_line('Value of c: ' || c);
  f := 70.0/3.0;
  dbms_output.put_line('Value of f: ' || f);
END;
```



```
Value of c: 30
Value of f: 23.33333333333333333333333333333333333333

Statement processed.

0.01 seconds
```

CREATE OR REPLACE FUNCTION tomorrow (p_today IN DATE)

RETURN DATE IS

v_tomorrow DATE;

BEGIN

SELECT p_today + 1 INTO v_tomorrow

FROM DUAL;

RETURN v_tomorrow;

END;

SELECT TOMORROW(SYSDATE) AS "TomorrowDate"

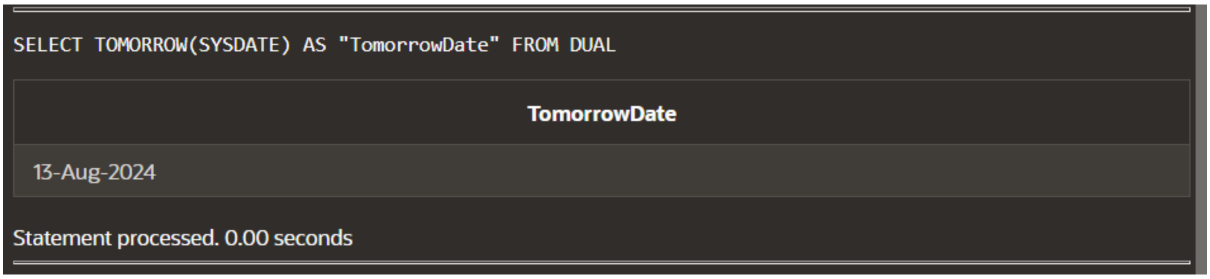
FROM DUAL;

or

BEGIN

DBMS_OUTPUT.PUT_LINE(TOMORROW(SYSDATE));

END;



```
SELECT TOMORROW(SYSDATE) AS "TomorrowDate" FROM DUAL
```

TomorrowDate
13-Aug-2024

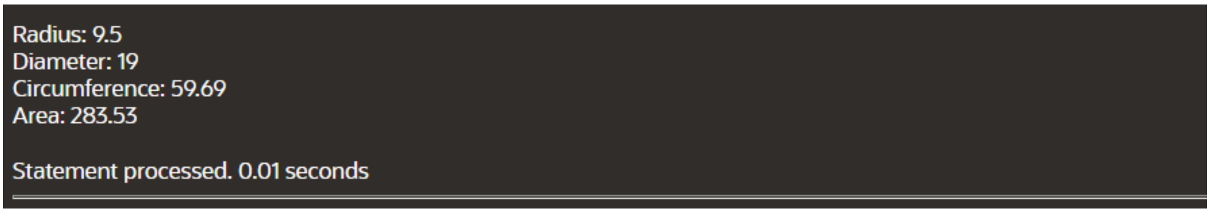
Statement processed. 0.00 seconds

DECLARE

   -- constant declaration

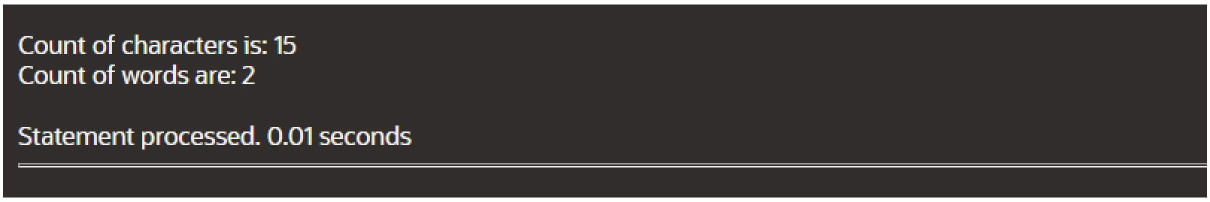
   pi constant number := 3.141592654;

```
  -- other declarations
  radius number(5,2);
  dia number(5,2);
  circumference number(7, 2);
  area number (10, 2);
BEGIN
  -- processing
  radius := 9.5;
  dia := radius * 2;
  circumference := 2.0 * pi * radius;
  area := pi * radius * radius;
  -- output
  dbms_output.put_line('Radius: ' || radius);
  dbms_output.put_line('Diameter: ' || dia);
  dbms_output.put_line('Circumference: ' || circumference);
  dbms_output.put_line('Area: ' || area);
END;
```



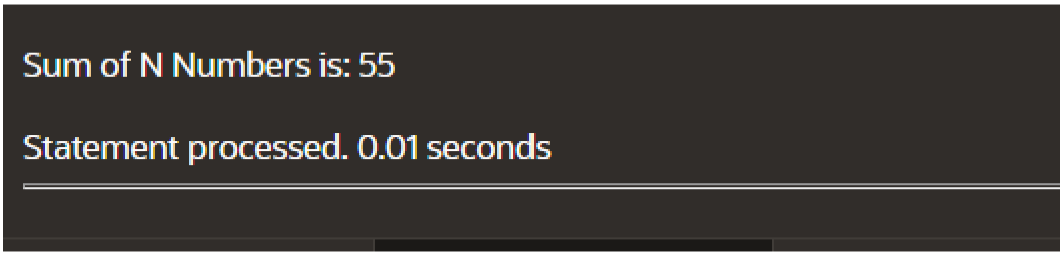
```
Radius: 9.5
Diameter: 19
Circumference: 59.69
Area: 283.53

Statement processed. 0.01 seconds
```

```
DECLARE
    str VARCHAR2(40) := 'Tutorials Point';
    nchars NUMBER(4) := 0;
    nwords NUMBER(4) := 1;
    s CHAR;
BEGIN
  FOR i IN 1..Length(str) LOOP
    s := Substr(str, i, 1);
    nchars:= nchars+ 1;
    IF s = ' ' THEN

    nwords := nwords + 1;

    END IF;

END LOOP;

dbms_output.Put_line('count of characters is:'

  ||nchars);


dbms_output.Put_line('Count of words are: '

  ||nwords);

END;
```



```
Count of characters is: 15
Count of words are: 2

Statement processed. 0.01 seconds
```

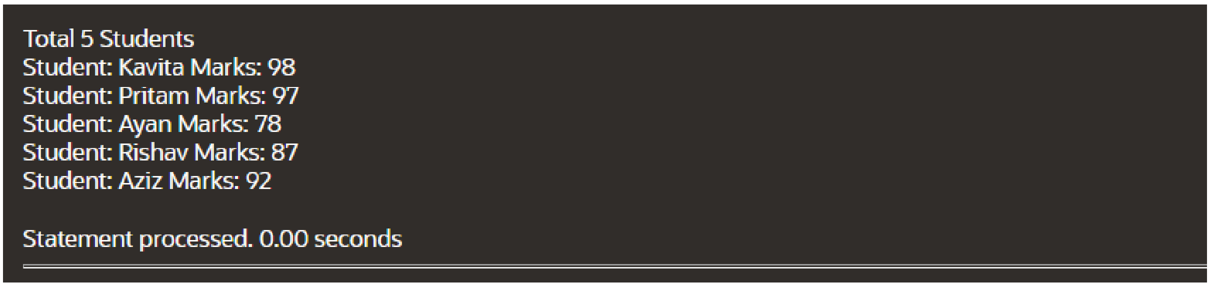
```
DECLARE

  total_sum NUMBER := 0;

  n NUMBER := 10;

BEGIN

  FOR i IN 1 .. n LOOP

    total_sum := total_sum + i;

  END LOOP;


  DBMS_OUTPUT.PUT_LINE('Sum of N Numbers is: ' || total_sum);

END;
```



```
Sum of N Numbers is: 55

Statement processed. 0.01 seconds
```

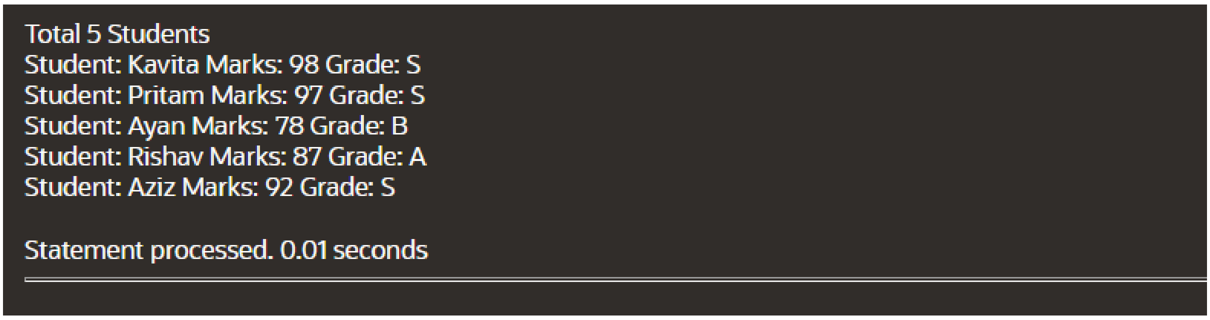
```
DECLARE
```

```
  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;
```



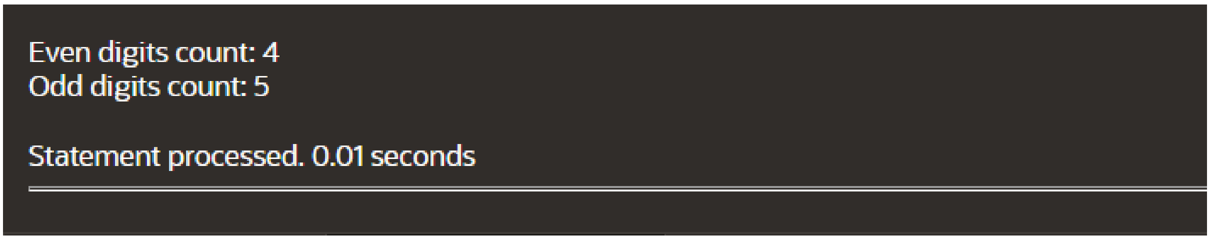
```
Total 5 Students
Student: Kavita Marks: 98
Student: Pritam Marks: 97
Student: Ayan Marks: 78
Student: Rishav Marks: 87
Student: Aziz Marks: 92

Statement processed. 0.00 seconds
```

```
DECLARE

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

  TYPE grades IS VARRAY(5) OF INTEGER;


  names namesarray;

  marks grades;

  total INTEGER;

  grade CHAR(1);

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
    IF marks(i) >= 90 THEN
      grade := 'S';
    ELSIF marks(i) >= 80 THEN
      grade := 'A';
    ELSIF marks(i) >= 70 THEN
      grade := 'B';
    ELSE
      grade := 'C';
    END IF;


    dbms_output.put_line('Student: ' || names(i) || ' Marks: ' || marks(i) || ' Grade: ' || grade);
  END LOOP;
END;
```



```
Total 5 Students
Student: Kavita Marks: 98 Grade: S
Student: Pritam Marks: 97 Grade: S
Student: Ayan Marks: 78 Grade: B
Student: Rishav Marks: 87 Grade: A
Student: Aziz Marks: 92 Grade: S

Statement processed. 0.01 seconds
```

```
DECLARE
   v_number     NUMBER := 123456789;
   v_digit      NUMBER;
   v_num        NUMBER;
   v_even_count  NUMBER := 0;
   v_odd_count   NUMBER := 0;
BEGIN
   v_num := v_number;

   WHILE v_num > 0 LOOP
      v_digit := MOD(v_num, 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_num := FLOOR(v_num / 10);
   END LOOP;

   DBMS_OUTPUT.PUT_LINE('Even digits count: ' || v_even_count);
   DBMS_OUTPUT.PUT_LINE('Odd digits count: ' || v_odd_count);
END;
```

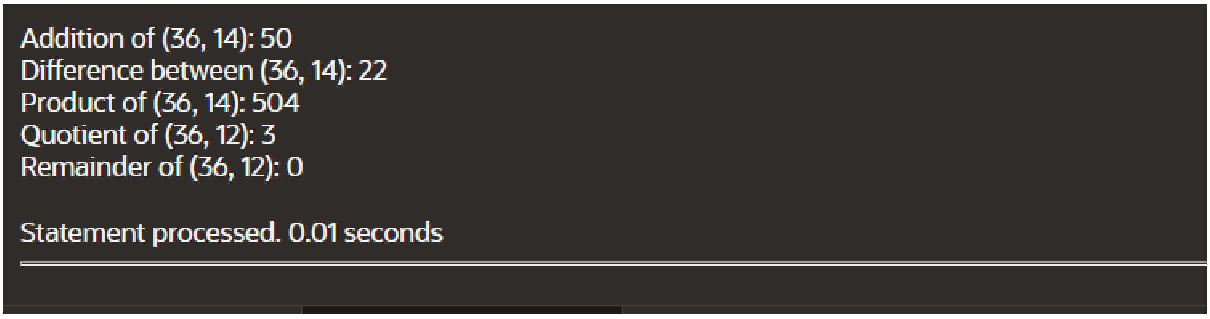


```
Even digits count: 4
Odd digits count: 5

Statement processed. 0.01 seconds
```

```
DECLARE
  a NUMBER;
  b NUMBER;
  c NUMBER;

  PROCEDURE Addtwo(x IN NUMBER, y IN NUMBER, z OUT NUMBER) IS
  BEGIN
```

```
    z := x + y;
  END;

  PROCEDURE Subtwo(x IN NUMBER, y IN NUMBER, z OUT NUMBER) IS
  BEGIN
    z := x - y;
  END;

  PROCEDURE Multwo(x IN NUMBER, y IN NUMBER, z OUT NUMBER) IS
  BEGIN
    z := x * y;
  END;

  PROCEDURE Divtwo(x IN NUMBER, y IN NUMBER, z OUT NUMBER) IS
  BEGIN
    IF y = 0 THEN
      DBMS_OUTPUT.PUT_LINE('Error: Division by zero.');
      RETURN;
    ELSE
      z := TRUNC(x / y);
    END IF;
  END;

  PROCEDURE Modtwp(x IN NUMBER, y IN NUMBER, z OUT NUMBER) IS
  BEGIN
    z := MOD(x, y);
  END;

BEGIN
  a := 36;
  b := 14;
  Addtwo(a, b, c);
  DBMS_OUTPUT.PUT_LINE('Addition of (36, 14): ' || c);

  Subtwo(a, b, c);
  DBMS_OUTPUT.PUT_LINE('Difference between (36, 14): ' || c);

  Multwo(a, b, c);
  DBMS_OUTPUT.PUT_LINE('Product of (36, 14): ' || c);

  b := 12;

  Divtwo(a, b, c);
  DBMS_OUTPUT.PUT_LINE('Quotient of (36, 12): ' || c);

  Modtwp(a, b, c);
  DBMS_OUTPUT.PUT_LINE('Remainder of (36, 12): ' || c);
END;
/
```

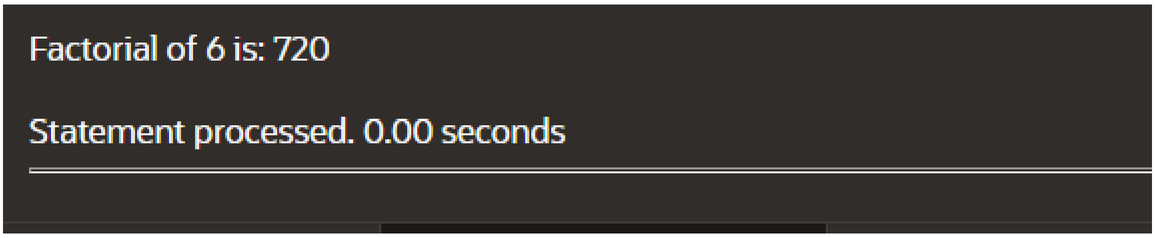
O/P:



```
Addition of (36, 14): 50
Difference between (36, 14): 22
Product of (36, 14): 504
Quotient of (36, 12): 3
Remainder of (36, 12): 0

Statement processed. 0.01 seconds
```

```
DECLARE
  num number;
  factorial number;
```

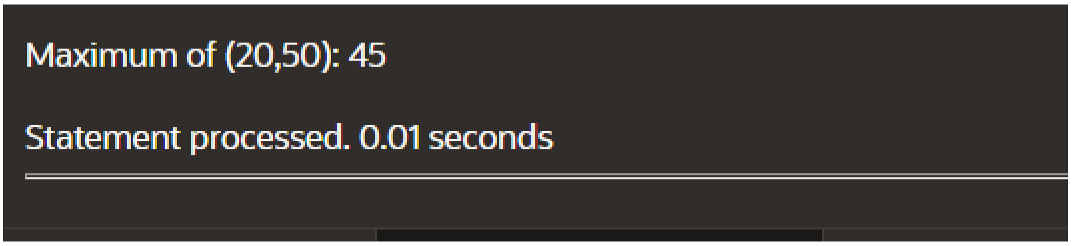
```
FUNCTION fact(x number)
RETURN number
IS
  f number;
BEGIN
  IF x=0 THEN
    f := 1;
  ELSE
    f := x * fact(x-1);
  END IF;
RETURN f;
END;

BEGIN
  num:= 6;
  factorial := fact(num);
  dbms_output.put_line(' Factorial '|| num || ' is ' || factorial);
END;
```



```
Factorial of 6 is: 720

Statement processed. 0.00 seconds
```

```
DECLARE
   a number;
   b number;
   c number;
FUNCTION findMax(x IN number, y IN number)
RETURN number
IS
   z number;
```

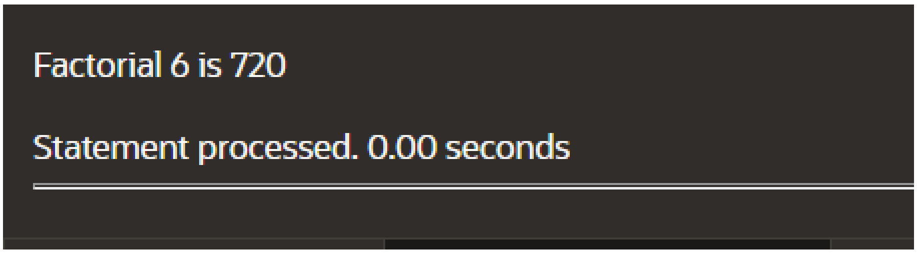
```
BEGIN
IF x>y THEN
   z:= y;
ELSE
   z:=y;
   END IF;
   RETURN z;
END;
BEGIN
   a :=23;
   b := 45;
   c:= findMax(a,b);
   dbms_output.put_line('Maximum of (20,50): ' || c);
   END;
```



```
Maximum of (20,50): 45

Statement processed. 0.01 seconds
```

```
DECLARE
  num number;
  factorial number;

FUNCTION fact(x number)
RETURN number
IS
  f number;
BEGIN
  IF x=0 THEN
    f := 1;
  ELSE
```

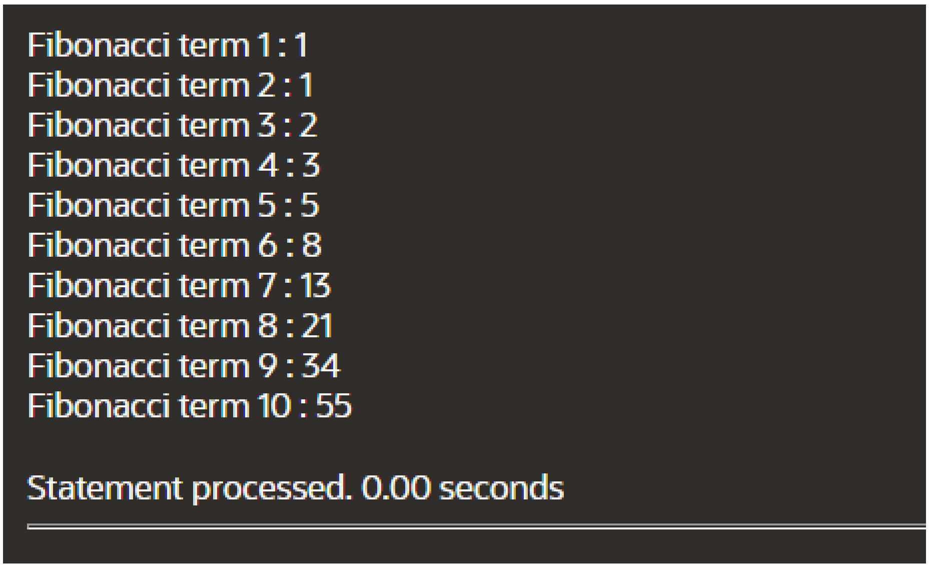
```
    f := x * fact(x-1);
  END IF;
RETURN f;
END;

BEGIN
  num:= 6;
  factorial := fact(num);
  dbms_output.put_line(' Factorial '|| num || ' is ' || factorial);
END;
```



```
Factorial 6 is 720

Statement processed. 0.00 seconds
```

```
CREATE OR REPLACE FUNCTION fibonacci(n IN NUMBER) RETURN NUMBER IS
  result NUMBER;
BEGIN
  IF n <= 0 THEN
    result := 0;
  ELSIF n = 1 THEN
    result: = 1;
  ELSE
    result := fibonacci(n - 1) + fibonacci(n - 2);
  END IF;
  RETURN result;
END;
/

DECLARE
```

```
  num_terms NUMBER := 10;
  i NUMBER;
  fib_num NUMBER;
BEGIN
  FOR i IN 1..num_terms LOOP
    fib_num := fibonacci(i);
    DBMS_OUTPUT.PUT_LINE('Fibonacci term ' || i || ' : ' || fib_num);
  END LOOP;
END;
/
```

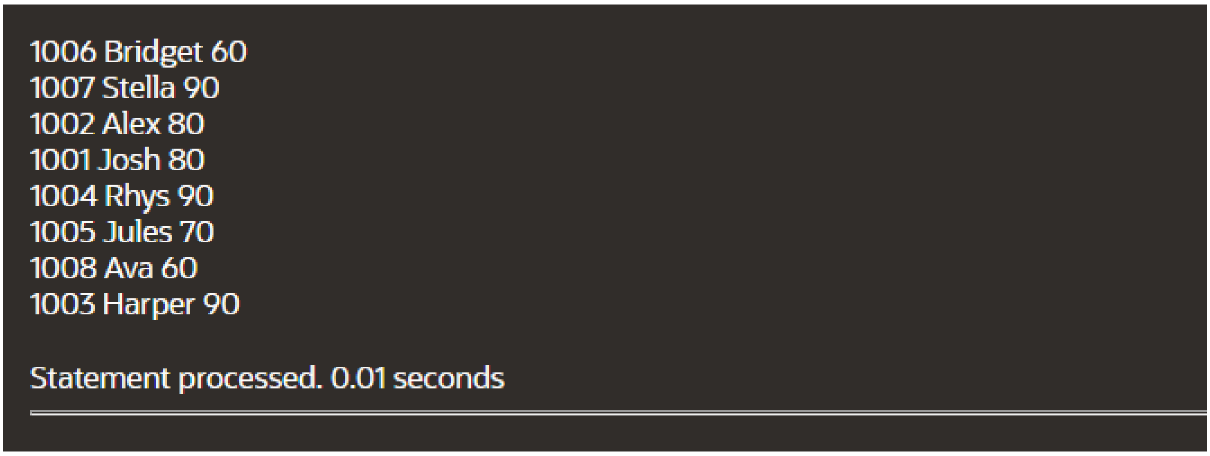


```
Fibonacci term 1 : 1
Fibonacci term 2 : 1
Fibonacci term 3 : 2
Fibonacci term 4 : 3
Fibonacci term 5 : 5
Fibonacci term 6 : 8
Fibonacci term 7 : 13
Fibonacci term 8 : 21
Fibonacci term 9 : 34
Fibonacci term 10 : 55

Statement processed. 0.00 seconds
```

--(IMPLICIT)

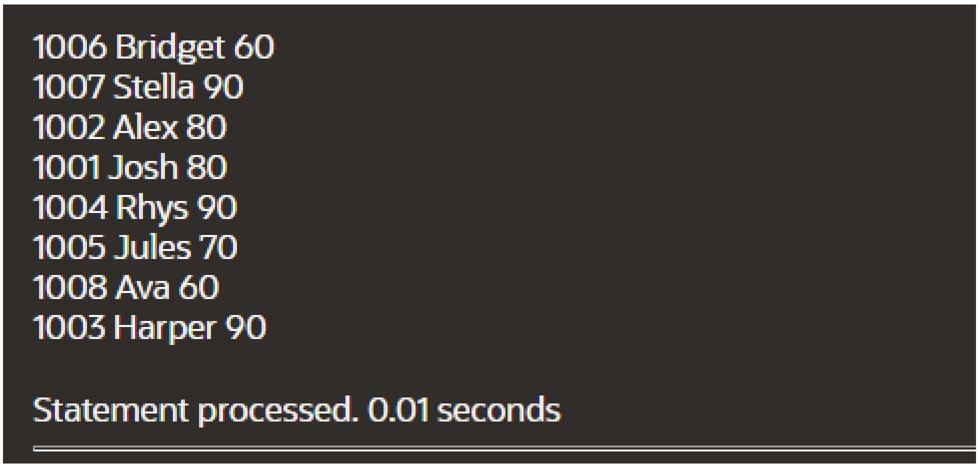
```
DECLARE
  c_emp_id   EMPLOYEE.EMP_ID%TYPE;
  c_emp_name EMPLOYEE.EMP_NAME%TYPE;
  c_dept_id  EMPLOYEE.DEPT_ID%TYPE;
  CURSOR c_employee IS
    SELECT EMP_ID, EMP_NAME, DEPT_ID FROM EMPLOYEE;
BEGIN
  OPEN c_employee;
  LOOP
```

```
    FETCH c_employee INTO c_emp_id, c_emp_name, c_dept_id;
    EXIT WHEN c_employee%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE (c_emp_id || ' ' || c_emp_name || ' ' || c_dept_id);
  END LOOP;
  CLOSE c_employee;
END;
```



```
1006 Bridget 60
1007 Stella 90
1002 Alex 80
1001 Josh 80
1004 Rhys 90
1005 Jules 70
1008 Ava 60
1003 Harper 90

Statement processed. 0.01 seconds
```

```
--(EXPLICIT)
DECLARE
  c_emp_id   employee.emp_id%TYPE;
  c_emp_name employee.emp_name%TYPE;
  c_dept_id employee.dept_id%TYPE;

  CURSOR c_employee IS
    SELECT emp_id, emp_name, dept_id
    FROM employee;

BEGIN
  OPEN c_employee;
  LOOP
    FETCH c_employee INTO c_emp_id, c_emp_name, c_dept_id;
    EXIT WHEN c_employee%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE (c_emp_id || ' ' || c_emp_name || ' ' || c_dept_id);
  END LOOP;

  CLOSE c_employee;

END;
```



```
1006 Bridget 60
1007 Stella 90
1002 Alex 80
1001 Josh 80
1004 Rhys 90
1005 Jules 70
1008 Ava 60
1003 Harper 90

Statement processed. 0.01 seconds
```