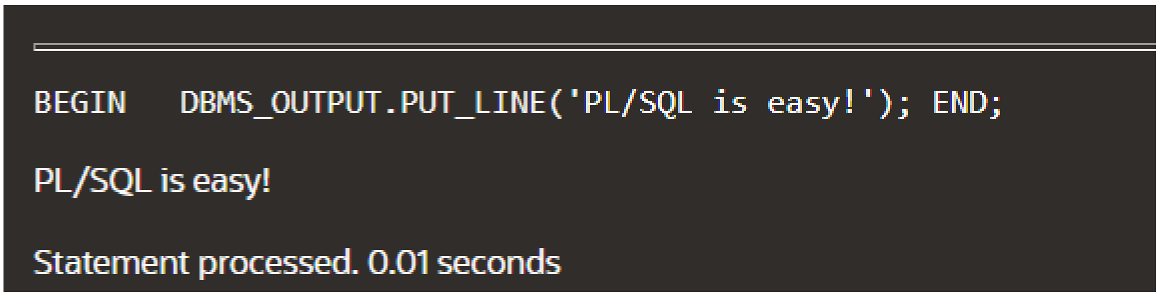
DATABASE MANAGEMENT SYSTEMS SQL PRACTICE -4

NAME: HARINI.R

REGISTER NO: 192324108

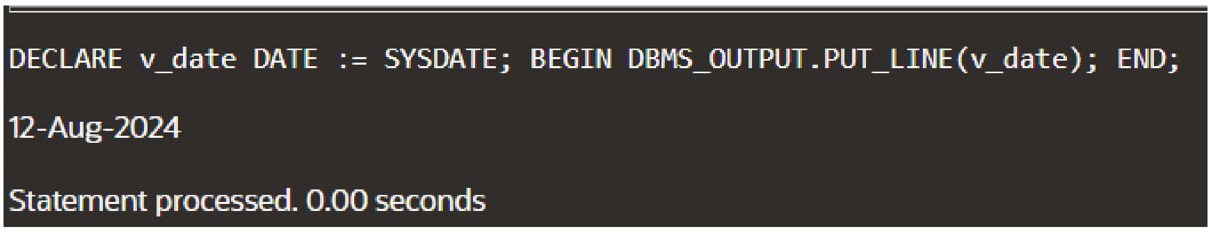
COURSE CODE: CSA0564

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



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

PL/SQL is easy!

Statement processed. 0.01 seconds
```

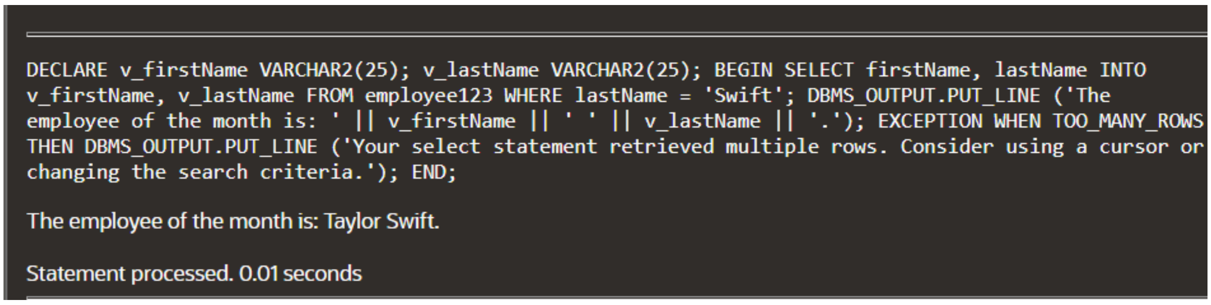
```
DECLARE
v_date DATE := SYSDATE;
BEGIN
DBMS_OUTPUT.PUT_LINE(v_date);
END;
```



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

12-Aug-2024

Statement processed. 0.00 seconds
```

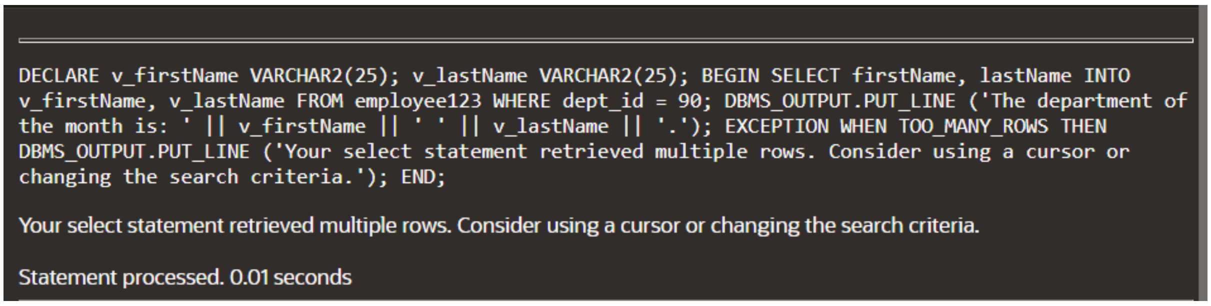
```
DECLARE
v_firstName VARCHAR2(25);
v_lastName VARCHAR2(25);
BEGIN
SELECT firstName, lastName
INTO v_firstName, v_lastName
FROM employee123
WHERE lastName = 'Swift';
```

```
DBMS_OUTPUT.PUT_LINE ('The employee of the month is: '
|| v_firstName || ' ' || v_lastName || '.');
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;
```



```
DECLARE v_firstName VARCHAR2(25); v_lastName VARCHAR2(25); BEGIN SELECT firstName, lastName INTO
v_firstName, v_lastName FROM employee123 WHERE lastName = 'Swift'; DBMS_OUTPUT.PUT_LINE ('The
employee of the month is: ' || v_firstName || ' ' || v_lastName || '.'); 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: Taylor Swift.

Statement processed. 0.01 seconds
```

```
DECLARE
v_firstName VARCHAR2(25);
v_lastName VARCHAR2(25);
BEGIN
SELECT firstName, lastName
INTO v_firstName, v_lastName
FROM employee123
WHERE dept_id = 90;
DBMS_OUTPUT.PUT_LINE ('The employee of the month is: '
|| v_firstName || ' ' || v_lastName || '.');
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;
```



```
DECLARE v_firstName VARCHAR2(25); v_lastName VARCHAR2(25); BEGIN SELECT firstName, lastName INTO
v_firstName, v_lastName FROM employee123 WHERE dept_id = 90; DBMS_OUTPUT.PUT_LINE ('The department of
the month is: ' || v_firstName || ' ' || v_lastName || '.'); 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;

Your select statement retrieved multiple rows. Consider using a cursor or changing the search criteria.

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;

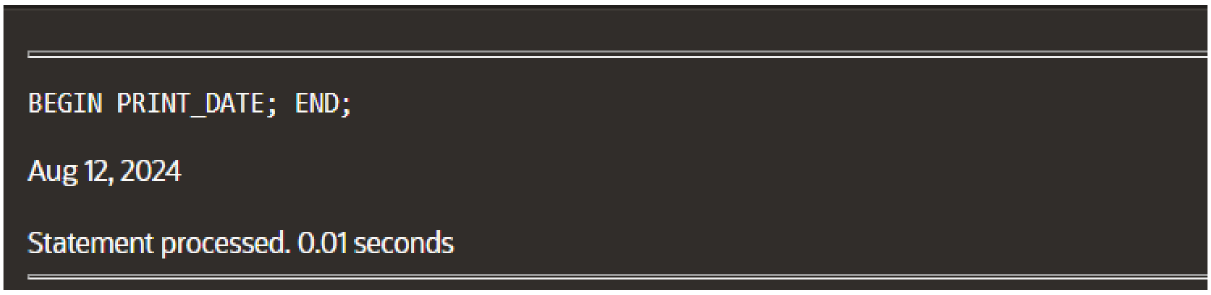
DBMS_OUTPUT.PUT_LINE(v_date);

END;

BEGIN

PRINT_DATE;

END;



```
BEGIN PRINT_DATE; END;

Aug 12, 2024

Statement processed. 0.01 seconds
```

CREATE OR REPLACE FUNCTION tomorrow (p_today IN DATE)

RETURN DATE IS

v_tomorrow DATE;

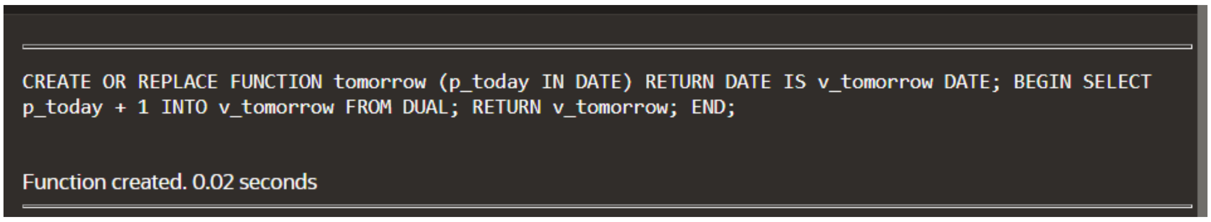
BEGIN

SELECT p_today + 1 INTO v_tomorrow

FROM DUAL;

RETURN v_tomorrow;

END;



```
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;

Function created. 0.02 seconds
```

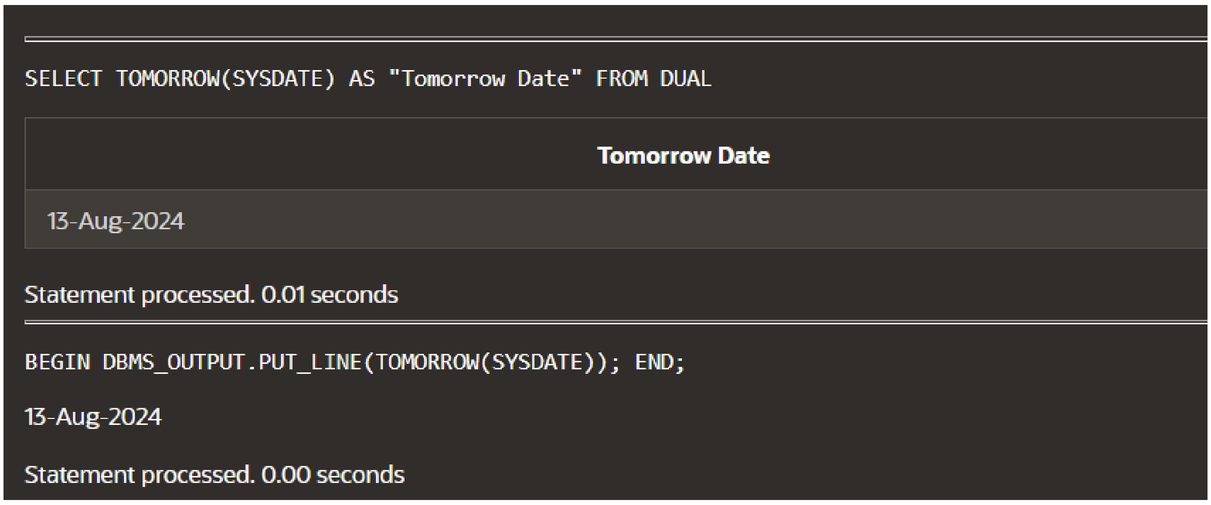
SELECT TOMORROW(SYSDATE) AS "Tomorrow Date"

FROM DUAL;

BEGIN

DBMS_OUTPUT.PUT_LINE(TOMORROW(SYSDATE));

END;



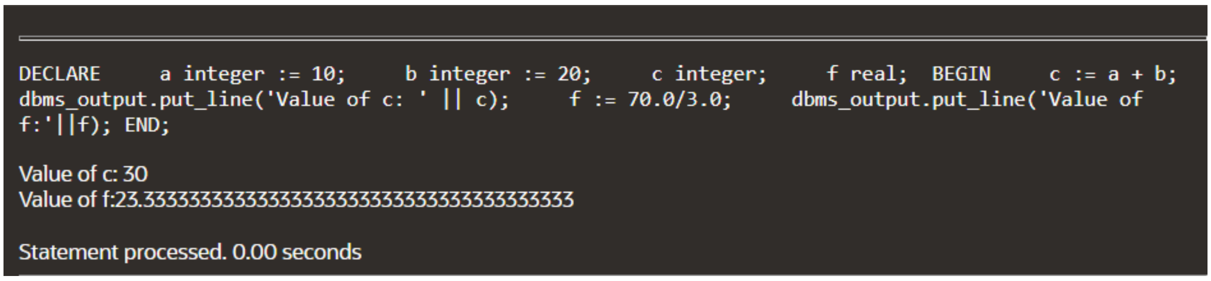
```
SELECT TOMORROW(SYSDATE) AS "Tomorrow Date" FROM DUAL
```

Tomorrow Date
13-Aug-2024

```
Statement processed. 0.01 seconds

BEGIN DBMS_OUTPUT.PUT_LINE(TOMORROW(SYSDATE)); END;

13-Aug-2024

Statement processed. 0.00 seconds
```

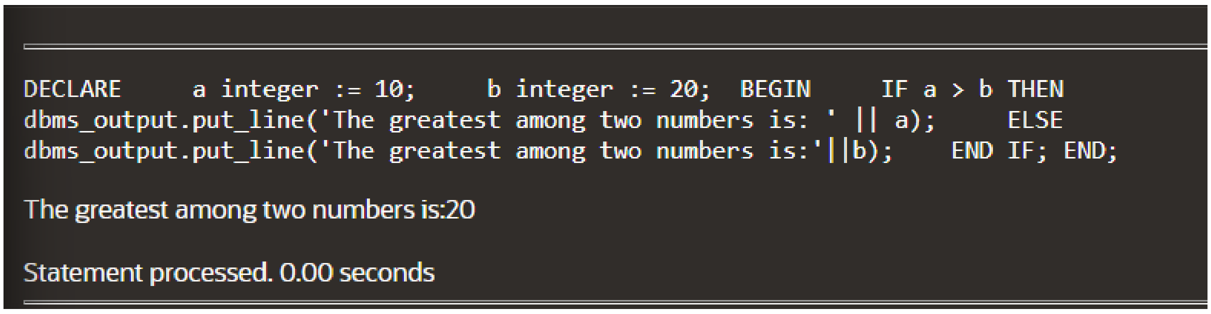
```
DECLARE
  a integer := 10;
  b integer := 20;
  c integer;
  f real;
BEGIN
  c := a + b;
  dbms_output.put_line('Value of c: ' || c);
  f := 70.0/3.0;
  dbms_output.put_line('Value of f:'||f);
```

END;



```
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. 0.00 seconds
```

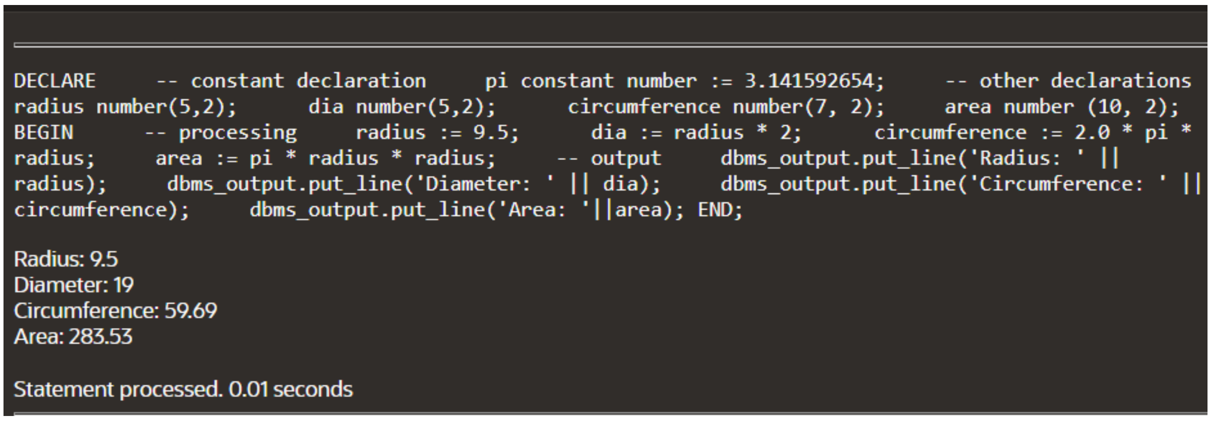
```
DECLARE
  a integer := 10;
  b integer := 20;
BEGIN
  IF a > b THEN
  dbms_output.put_line('The greatest among two numbers is: ' || a);
  ELSE
  dbms_output.put_line('The greatest among two numbers is:'||b);
  END IF;
END;
```



```
DECLARE     a integer := 10;     b integer := 20;  BEGIN     IF a > b THEN
dbms_output.put_line('The greatest among two numbers is: ' || a);     ELSE
dbms_output.put_line('The greatest among two numbers is:'||b);    END IF; END;

The greatest among two numbers is:20

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;
```



```
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;
```

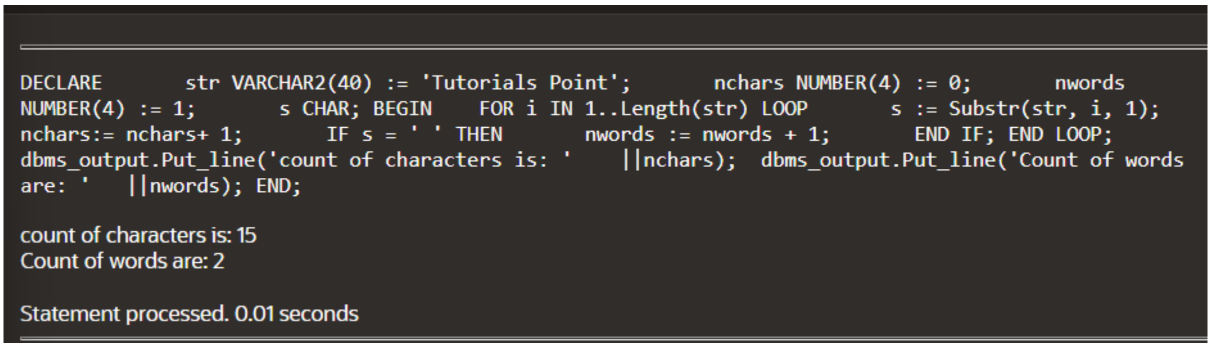
dbms_output.Put_line('count of characters is: '

  ||nchars);

dbms_output.Put_line('Count of words are: '

  ||nwords);

END;



```
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

  n NUMBER := 10;

  nsum NUMBER := 0;

BEGIN

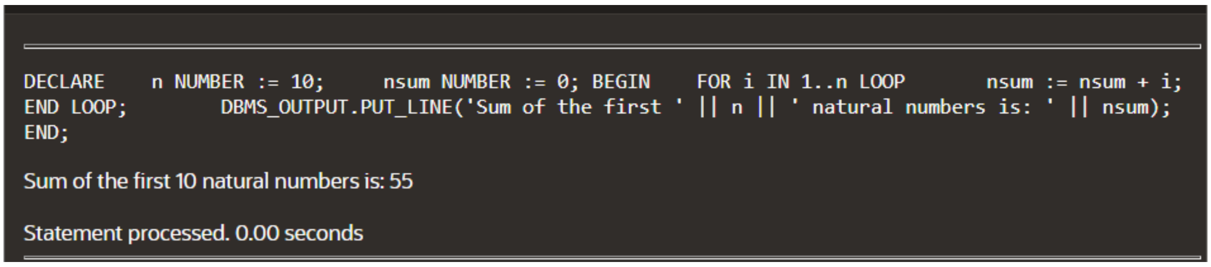
  FOR i IN 1..n LOOP

    nsum := nsum + i;

  END LOOP;

  DBMS_OUTPUT.PUT_LINE('Sum of the first ' || n || ' natural numbers is: ' || nsum);

END;



```
DECLARE    n NUMBER := 10;     nsum NUMBER := 0; BEGIN    FOR i IN 1..n LOOP       nsum := nsum + i;
END LOOP;        DBMS_OUTPUT.PUT_LINE('Sum of the first ' || n || ' natural numbers is: ' || nsum);
END;

Sum of the first 10 natural numbers is: 55

Statement processed. 0.00 seconds
```

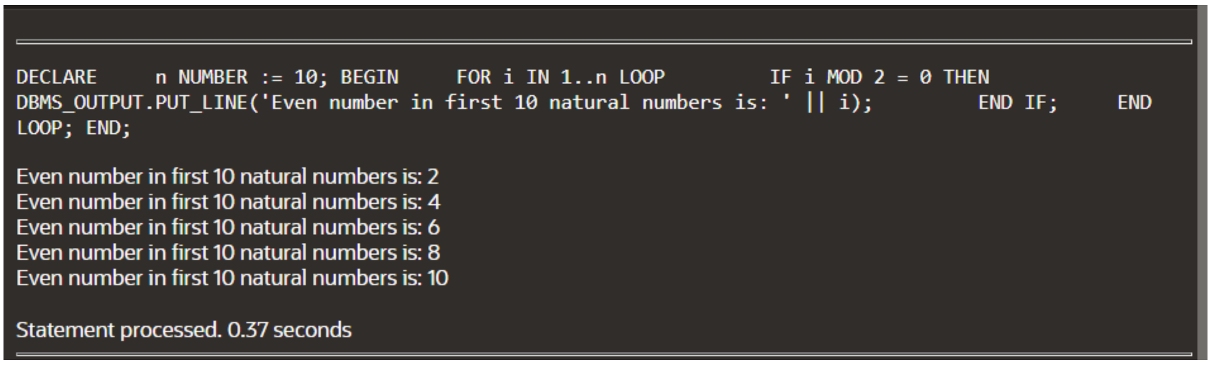
DECLARE

   n NUMBER := 10;

BEGIN

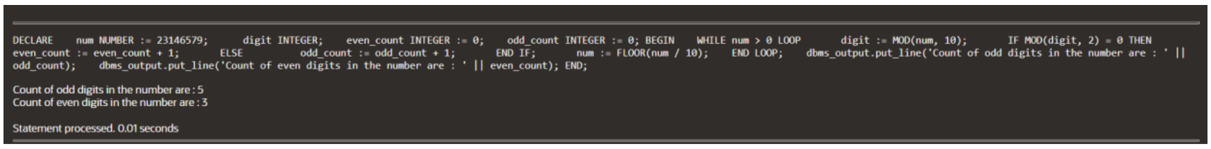
   FOR i IN 1..n LOOP

```
    IF i MOD 2 = 0 THEN
      DBMS_OUTPUT.PUT_LINE('Even number in first 10 natural numbers is: ' || i);
    END IF;
  END LOOP;
END;
```



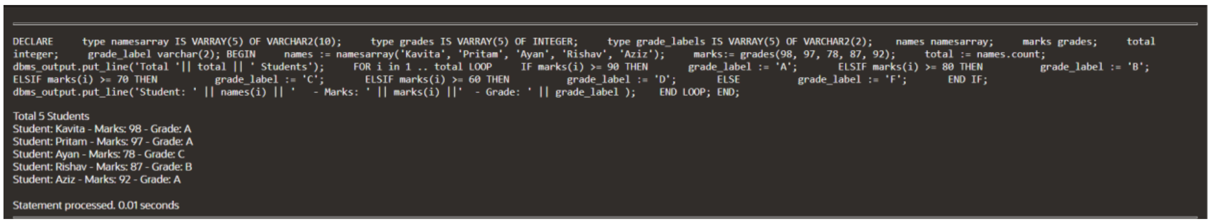
```
DECLARE      n NUMBER := 10; BEGIN      FOR i IN 1..n LOOP          IF i MOD 2 = 0 THEN
DBMS_OUTPUT.PUT_LINE('Even number in first 10 natural numbers is: ' || i);         END IF;     END
LOOP; END;

Even number in first 10 natural numbers is: 2
Even number in first 10 natural numbers is: 4
Even number in first 10 natural numbers is: 6
Even number in first 10 natural numbers is: 8
Even number in first 10 natural numbers is: 10

Statement processed. 0.37 seconds
```

```
DECLARE
  num NUMBER := 23146579;
  digit INTEGER;
  even_count INTEGER := 0;
  odd_count INTEGER := 0;
BEGIN
  WHILE num > 0 LOOP
    digit := MOD(num, 10);
    IF MOD(digit, 2) = 0 THEN
      even_count := even_count + 1;
    ELSE
      odd_count := odd_count + 1;
    END IF;
    num := FLOOR(num / 10);
  END LOOP;
  dbms_output.put_line('Count of odd digits in the number are : ' || odd_count);
  dbms_output.put_line('Count of even digits in the number are : ' || even_count);
END;
```



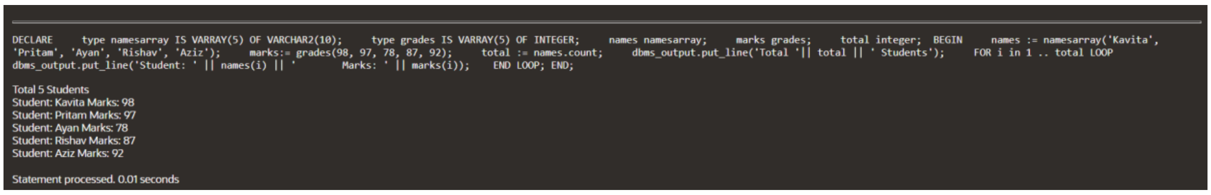
```
DECLARE    num NUMBER := 23146579;      digit INTEGER;    even_count INTEGER := 0;    odd_count INTEGER := 0; BEGIN    WHILE num > 0 LOOP       digit := MOD(num, 10);       IF MOD(digit, 2) = 0 THEN
even_count := even_count + 1;       ELSE          odd_count := odd_count + 1;       END IF;       num := FLOOR(num / 10);    END LOOP;    dbms_output.put_line('Count of odd digits in the number are : ' ||
odd_count);    dbms_output.put_line('Count of even digits in the number are : ' || even_count); END;

Count of odd digits in the number are : 5
Count of even digits in the number are : 3

Statement processed. 0.01 seconds
```

```
DECLARE
  type namesarray IS VARRAY(5) OF VARCHAR2(10);
  type grades IS VARRAY(5) OF INTEGER;
  type grade_labels IS VARRAY(5) OF VARCHAR2(2);
  names namesarray;
  marks grades;
  total integer;
  grade_label varchar(2);
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_label := 'A';
    ELSIF marks(i) >= 80 THEN
      grade_label := 'B';
    ELSIF marks(i) >= 70 THEN
      grade_label := 'C';
    ELSIF marks(i) >= 60 THEN
      grade_label := 'D';
    ELSE
      grade_label := 'F';
    END IF;
    dbms_output.put_line('Student: ' || names(i) || '   - Marks: ' || marks(i) ||'  - Grade: ' ||
grade_label );
 END LOOP;
END;
```



```
DECLARE     type namesarray IS VARRAY(5) OF VARCHAR2(10);     type grades IS VARRAY(5) OF INTEGER;     type grade_labels IS VARRAY(5) OF VARCHAR2(2);    names namesarray;     marks grades;     total
integer;     grade_label varchar(2); 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_label := 'A';       ELSIF marks(i) >= 80 THEN          grade_label := 'B';
ELSIF marks(i) >= 70 THEN          grade_label := 'C';       ELSIF marks(i) >= 60 THEN          grade_label := 'D';       ELSE          grade_label := 'F';       END IF;
dbms_output.put_line('Student: ' || names(i) || '   - Marks: ' || marks(i) ||'  - Grade: ' || grade_label );    END LOOP; END;

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

Statement processed. 0.01 seconds
```

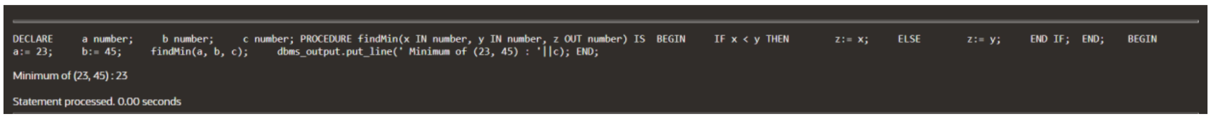
```
DECLARE
  type namesarray IS VARRAY(5) OF VARCHAR2(10);
  type grades IS VARRAY(5) OF INTEGER;
  names namesarray;
  marks grades;
  total integer;
BEGIN
  names := namesarray('Kavita', 'Pritam', 'Ayan', 'Rishav', 'Aziz');
  marks:= grades(98, 97, 78, 87, 92);
  total := names.count;
  dbms_output.put_line('Total '|| total || ' Students');
  FOR i in 1 .. total LOOP
    dbms_output.put_line('Student: ' || names(i) || '
    Marks: ' || marks(i));
  END LOOP;
END;
```



```
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. 0.01 seconds
```

```
DECLARE
  a number;
  b number;
  c number;
PROCEDURE findMin(x IN number, y IN number, z OUT number) IS
```

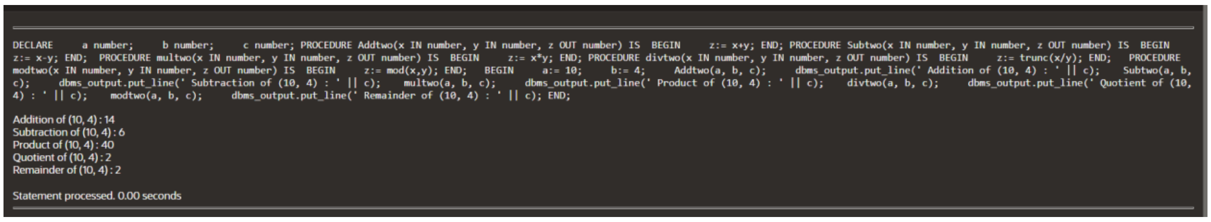
```
BEGIN
  IF x < y THEN
    z:= x;
  ELSE
    z:= y;
  END IF;
END;
BEGIN
  a:= 23;
  b:= 45;
  findMin(a, b, c);
  dbms_output.put_line(' Minimum of (23, 45) : '||c);
END;
```



```
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;

Minimum of (23, 45) : 23

Statement processed. 0.00 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;
```

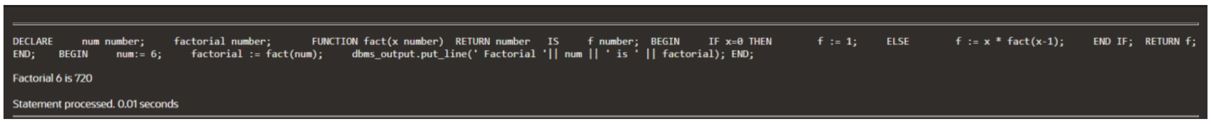
```
PROCEDURE divtwo(x IN number, y IN number, z OUT number) IS
BEGIN
  z:= trunc(x/y);
END;
PROCEDURE modtwo(x IN number, y IN number, z OUT number) IS
BEGIN
  z:= mod(x,y);
END;
BEGIN
  a:= 10;
  b:= 4;
  Addtwo(a, b, c);
  dbms_output.put_line(' Addition of (10, 4) : ' || c);
  Subtwo(a, b, c);
  dbms_output.put_line(' Subtraction of (10, 4) : ' || c);
  multwo(a, b, c);
  dbms_output.put_line(' Product of (10, 4) : ' || c);
  divtwo(a, b, c);
  dbms_output.put_line(' Quotient of (10, 4) : ' || c);
  modtwo(a, b, c);
  dbms_output.put_line(' Remainder of (10, 4) : ' || c);
END;
```



```
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     z:= trunc(x/y); END;   PROCEDURE
modtwo(x IN number, y IN number, z OUT number) IS  BEGIN     z:= mod(x,y); END;   BEGIN     a:= 10;     b:= 4;     Addtwo(a, b, c);     dbms_output.put_line(' Addition of (10, 4) : ' || c);    Subtwo(a, b,
c);     dbms_output.put_line(' Subtraction of (10, 4) : ' || c);    multwo(a, b, c);     dbms_output.put_line(' Product of (10, 4) : ' || c);    divtwo(a, b, c);     dbms_output.put_line(' Quotient of (10,
4) : ' || c);    modtwo(a, b, c);     dbms_output.put_line(' Remainder of (10, 4) : ' || c); END;

Addition of (10, 4) : 14
Subtraction of (10, 4) : 6
Product of (10, 4) : 40
Quotient of (10, 4) : 2
Remainder of (10, 4) : 2

Statement processed. 0.00 seconds
```

```
DECLARE
  num number;
  factorial number;


FUNCTION fact(x number)
```

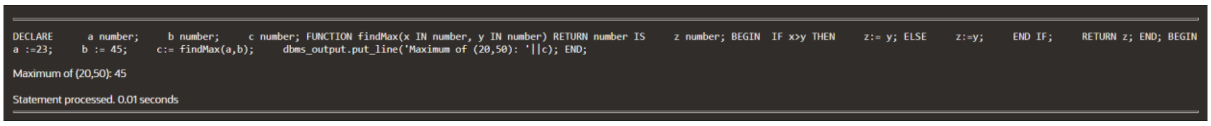
```
RETURN number
IS
  f number;
BEGIN
  IF x=0 THEN
    f := 1;
  ELSE
    f := x * fact(x-1);
  END IF;
RETURN f;
END;


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



```
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. 0.01 seconds
```

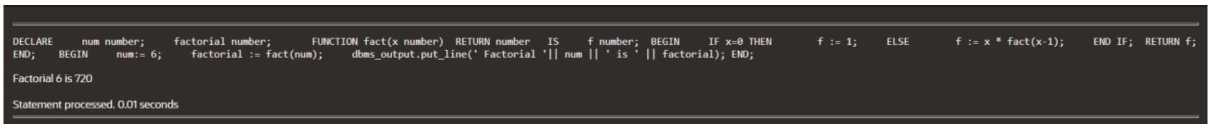
```
DECLARE
   a number;
   b number;
   c number;
FUNCTION findMax(x IN number, y IN number)
RETURN number
IS
   z number;
BEGIN
IF x>y THEN
   z:= y;
ELSE
```

```
  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;
```



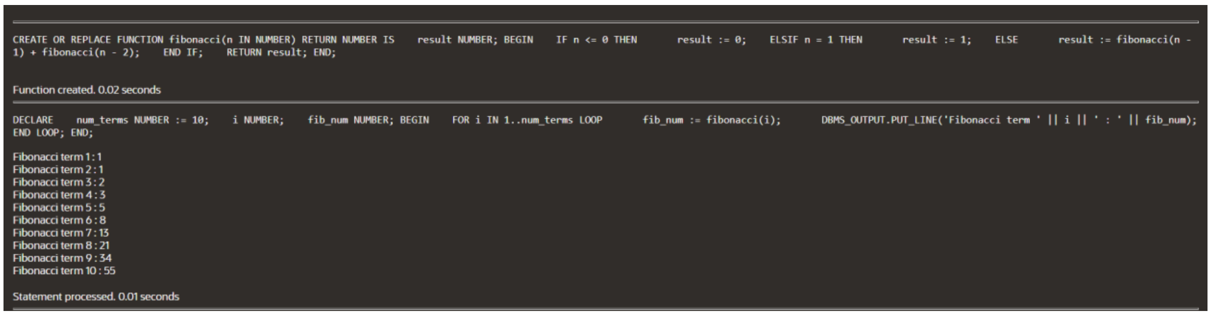
```
DECLARE      a number;     b number;     c number; FUNCTION findMax(x IN number, y IN number) RETURN number IS     z number; BEGIN  IF x>y THEN     z:= y; ELSE     z:=y;     END IF;     RETURN z; END; BEGIN
a :=23;     b := 45;     c:= findMax(a,b);     dbms_output.put_line('Maximum of (20,50): '||c);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



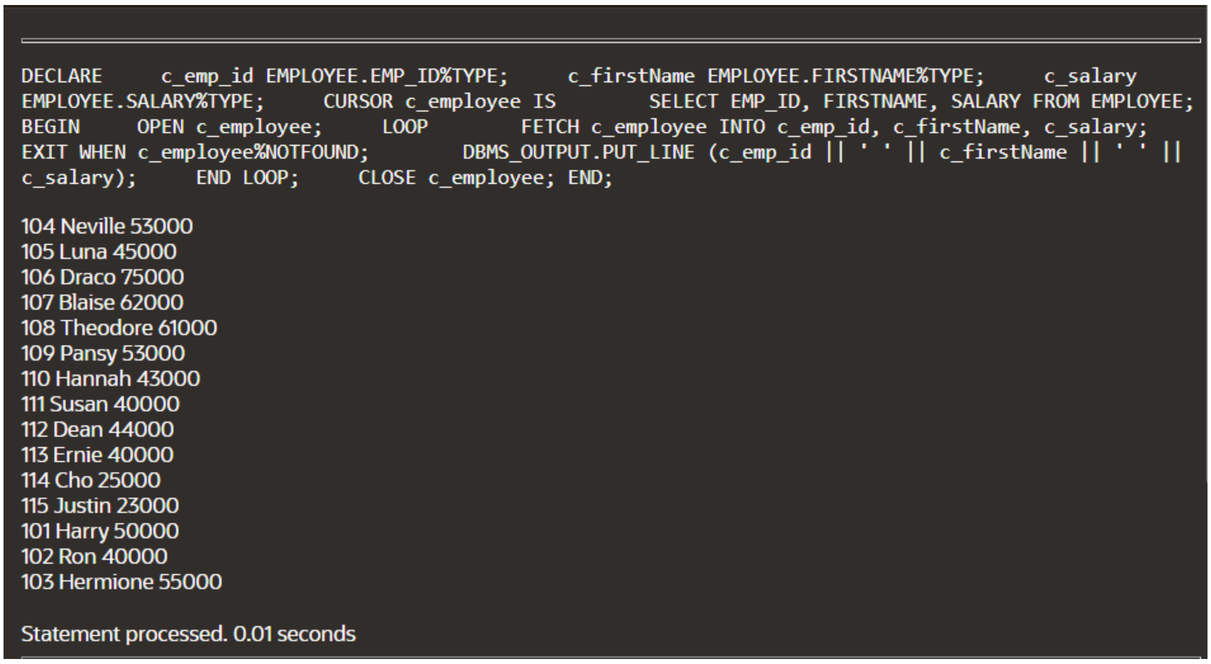
```
DECLARE     num number;      factorial number;       FUNCTION fact(x number)  RETURN number   IS     f number;  BEGIN     IF x=0 THEN        f := 1;      ELSE        f := x * fact(x-1);      END IF;  RETURN f;
END;    BEGIN     num:= 6;      factorial := fact(num);     dbms_output.put_line(' Factorial '|| num || ' is ' || factorial); END;

Factorial 6 is 720

Statement processed. 0.01 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;
```



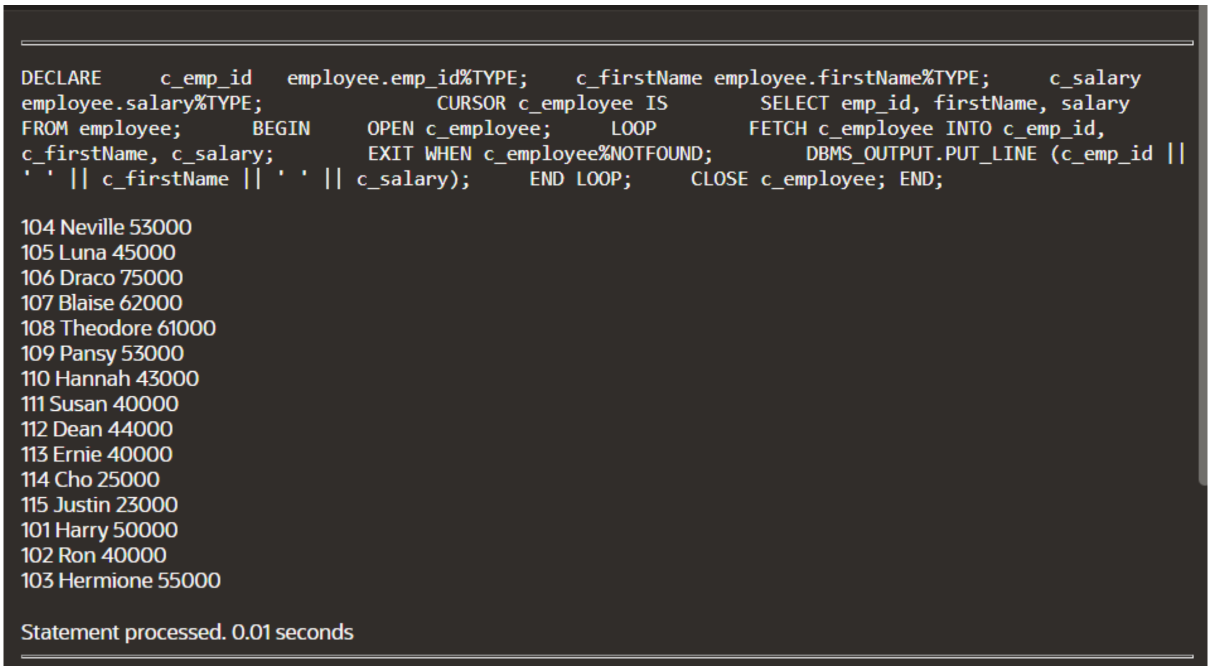
```
CREATE OR REPLACE FUNCTION fibonacci(n IN NUMBER) RETURN NUMBER IS    result NUMBER; BEGIN    IF n <= 0 THEN       result := 0;    ELSIF n = 1 THEN       result := 1;    ELSE       result := fibonacci(n -
1) + fibonacci(n - 2);    END IF;    RETURN result; END;

Function created. 0.02 seconds

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. 0.01 seconds
```

```
--(IMPLICIT)

DECLARE

  c_emp_id EMPLOYEE.EMP_ID%TYPE;

  c_firstName EMPLOYEE.FIRSTNAME%TYPE;

  c_salary EMPLOYEE.SALARY%TYPE;

  CURSOR c_employee IS

    SELECT EMP_ID, FIRSTNAME, SALARY FROM EMPLOYEE;

BEGIN

  OPEN c_employee;

  LOOP

    FETCH c_employee INTO c_emp_id, c_firstName, c_salary;

    EXIT WHEN c_employee%NOTFOUND;

    DBMS_OUTPUT.PUT_LINE (c_emp_id || ' ' || c_firstName || ' ' || c_salary);

  END LOOP;

  CLOSE c_employee;

END;
```



```
DECLARE     c_emp_id EMPLOYEE.EMP_ID%TYPE;     c_firstName EMPLOYEE.FIRSTNAME%TYPE;     c_salary
EMPLOYEE.SALARY%TYPE;     CURSOR c_employee IS        SELECT EMP_ID, FIRSTNAME, SALARY FROM EMPLOYEE;
BEGIN     OPEN c_employee;     LOOP        FETCH c_employee INTO c_emp_id, c_firstName, c_salary;
EXIT WHEN c_employee%NOTFOUND;        DBMS_OUTPUT.PUT_LINE (c_emp_id || ' ' || c_firstName || ' ' ||
c_salary);     END LOOP;     CLOSE c_employee; END;

104 Neville 53000
105 Luna 45000
106 Draco 75000
107 Blaise 62000
108 Theodore 61000
109 Pansy 53000
110 Hannah 43000
111 Susan 40000
112 Dean 44000
113 Ernie 40000
114 Cho 25000
115 Justin 23000
101 Harry 50000
102 Ron 40000
103 Hermione 55000

Statement processed. 0.01 seconds
```

```
--(EXPLICIT)
DECLARE
  c_emp_id   employee.emp_id%TYPE;
  c_firstName employee.firstName%TYPE;
  c_salary employee.salary%TYPE;

  CURSOR c_employee IS
    SELECT emp_id, firstName, salary
    FROM employee;

BEGIN
  OPEN c_employee;
  LOOP
    FETCH c_employee INTO c_emp_id, c_firstName, c_salary;
    EXIT WHEN c_employee%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE (c_emp_id || ' ' || c_firstName || ' ' || c_salary);
  END LOOP;
  CLOSE c_employee;
END;
```



```
DECLARE     c_emp_id    employee.emp_id%TYPE;     c_firstName employee.firstName%TYPE;      c_salary
employee.salary%TYPE;               CURSOR c_employee IS        SELECT emp_id, firstName, salary
FROM employee;      BEGIN     OPEN c_employee;     LOOP        FETCH c_employee INTO c_emp_id,
c_firstName, c_salary;        EXIT WHEN c_employee%NOTFOUND;        DBMS_OUTPUT.PUT_LINE (c_emp_id ||
' ' || c_firstName || ' ' || c_salary);     END LOOP;     CLOSE c_employee; END;

104 Neville 53000
105 Luna 45000
106 Draco 75000
107 Blaise 62000
108 Theodore 61000
109 Pansy 53000
110 Hannah 43000
111 Susan 40000
112 Dean 44000
113 Ernie 40000
114 Cho 25000
115 Justin 23000
101 Harry 50000
102 Ron 40000
103 Hermione 55000

Statement processed. 0.01 seconds
```