

PL/SQL:

```
1 BEGIN
2   DBMS_OUTPUT.PUT_LINE('SQL IS EASY');
3 END;
```

```
BEGIN   DBMS_OUTPUT.PUT_LINE('SQL IS EASY'); END;
```

```
SQL IS EASY
```

```
Statement processed. 0.01 seconds
```

DECLARATIVE AND EXECUTABLE STATEMENTS:

```
1  ✓ DECLARE
2    V_DATE DATE:=SYSDATE ;
3    BEGIN
4    DBMS_OUTPUT.PUT_LINE(V_DATE);
5    END;
```

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

```
12-Aug-2024
```

```
Statement processed. 0.01 seconds
```

EXECUTABLE,DECLARATABLE AND EXCEPTION HANDLING:

```

1 DECLARE
2   V_FIRST_NAME VARCHAR(25);
3   V_LAST_NAME VARCHAR(25);
4   BEGIN
5     SELECT FIRST_NAME, LAST_NAME
6     INTO V_FIRST_NAME, V_LAST_NAME
7     FROM EMPLOYEES
8     WHERE LAST_NAME = 'NANDHA';
9     DBMS_OUTPUT.PUT_LINE('THE EMPLOYEE OF THE MONTH:' || V_FIRST_NAME || ' ' || V_LAST_NAME || '.');
10    EXCEPTION
11    WHEN TOO_MANY_ROWS THEN
12    DBMS_OUTPUT.PUT_LINE('YOUR SELECT STATEMENT RETRIEVED MULTIPLE ROWS. CONSIDER USING A CURSRT OR CHANGING THE SEARCH CRITERIA. ');
13    END;

```

View:
☒ Detail
☐ Summary
Show:
☒ Statement
☒ Results
☒ Feedback
Go

Create AppEdit Script

```

DECLARE V_FIRST_NAME VARCHAR(25); V_LAST_NAME VARCHAR(25); BEGIN SELECT FIRST_NAME, LAST_NAME INTO V_FIRST_NAME, V_LAST_NAME FROM EMPLOYEES WHERE LAST_NAME = 'NANDHA'; DBMS_OUTPUT.PUT_LINE('THE EMPLOYEE OF THE MONTH:' || V_FIRST_NAME || ' ' || V_LAST_NAME || '.'); END;

```

THE EMPLOYEE OF THE MONTH:GOUTHAM NANDHA.

Statement processed. 0.01 seconds

ADDING OF TWO NUMBERS:

[illegible]

GREATEST OF TWO NUMBERS:

```
1 DECLARE
2     a INTEGER := 10;
3     b INTEGER := 20;
4 BEGIN
5     IF a > b THEN
6         DBMS_OUTPUT.PUT_LINE('A IS GREATER');
7     ELSE
8         DBMS_OUTPUT.PUT_LINE('B IS GREATER');
9     END IF;
10 END;
11
```

```
DECLARE    a INTEGER := 10;    b INTEGER := 20; BEGIN    IF a > b THEN        DBMS_OUTPUT.PUT_LINE('A IS GREATER');    ELSE        DBMS_OUTPUT.PUT_LINE('B IS GREATER');    END IF;
END;
B IS GREATER
Statement processed. 0.01 seconds
```

CIRCLE AREA:

```
1 DECLARE
2     -- constant declaration
3     pi constant number := 3.141592654;
4     -- other declarations
5     radius number(5,2);
6     dia number(5,2);
7     circumference number(7, 2);
8     area number (10, 2);
9 BEGIN
10     -- processing
11     radius := 9.5;
12     dia := radius * 2;
13     circumference := 2.0 * pi * radius;
14     area := pi * radius * radius;
15     -- output
16     dbms_output.put_line('Radius: ' || radius);
17     dbms_output.put_line('Diameter: ' || dia);
18     dbms_output.put_line('Circumference: ' || circumference);
19     dbms_output.put_line('Area: ' || area);
20 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 0.00 seconds

```

NO OF CHARACTERS AND NO OF WORDS:

```

1  DECLARE
2      str VARCHAR2(40) := 'Tutorials Point';
3      nchars NUMBER(4) := 0;
4      nwords NUMBER(4) := 1;
5      s CHAR;
6  BEGIN
7      FOR i IN 1..Length(str) LOOP
8          s := Substr(str, i, 1);
9          nchars:= nchars+ 1;
10         IF s = ' ' THEN
11             nwords := nwords + 1;
12         END IF;
13     END LOOP;
14     dbms_output.Put_line('count of characters is:'
15         ||nchars);
16
17     dbms_output.Put_line('Count of words are: '
18         ||nwords);
19     END;

```

```

DECLARE      str VARCHAR2(40) := 'Tutorials Point';      nchars NUMBER(4) := 0;      nwords NUME
nchars:= nchars+ 1;      IF s = ' ' THEN      nwords := nwords + 1;      END IF; END LOOP; dbms_
||nwords); END;

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

Statement processed. 0.00 seconds

```

SUM OF N NUMBERS:

```
1 DECLARE
2     N NUMBER := 10;
3     N_SUM NUMBER := 0;
4 BEGIN
5     FOR i IN 1..N LOOP
6         N_SUM := N_SUM + i;
7     END LOOP;
8     dbms_output.Put_line('the sum of n numbers is:' || N_SUM);
9 END;
```

```
DECLARE      N NUMBER := 10;      N_SUM NUMBER := 0; BEGIN      FOR i IN 1..N LOOP      N_SUM := N_SUM + i;      END LOOP; dbms_output.Put_line('the sum of n numbers is:' || N_SUM); END;
the sum of n numbers is55
Statement processed. 0.00 seconds

Run By      IN_A816_PLSQL_S06
Parsing Schema      IN_A816_PLSQL_S06
```

Even numbers upto n:

```

1  DECLARE
2      num NUMBER := 20; -- Example value for n
3  BEGIN
4      FOR i IN 2..num LOOP
5          IF MOD(i, 2) = 0 THEN
6              DBMS_OUTPUT.PUT_LINE(i);
7          END IF;
8      END LOOP;
9  END;

```

```

DECLARE      num NUMBER := 20; -- Example v
2
4
6
8
10
12
14
16
18
20

Statement processed. 0.01 seconds

```

Odd numbers upto n:

```

1  DECLARE
2      num NUMBER := 20; -- Example value for n
3  BEGIN
4      FOR i IN 1..num LOOP
5          IF MOD(i, 2) != 0 THEN
6              DBMS_OUTPUT.PUT_LINE(i);
7          END IF;
8      END LOOP;
9  END;
10

```

```

DECLARE      num NUMBER := 20; -- Example va

1
3
5
7
9
11
13
15
17
19

Statement processed. 0.01 seconds

```

```

1  DECLARE
2      type namesarray IS VARRAY(5) OF VARCHAR2(10);
3      type grades IS VARRAY(5) OF INTEGER;
4      names namesarray;
5      marks grades;
6      total integer;
7  BEGIN
8      names := namesarray('Kavita', 'Pritam', 'Ayan', 'Rishav', 'Aziz');
9      marks:= grades(98, 97, 78, 87, 92);
10     total := names.count;
11     dbms_output.put_line('Total ' || total || ' Students');
12     FOR i in 1 .. total LOOP
13         dbms_output.put_line('Student: ' || names(i) || '
14         Marks: ' || marks(i));
15     END LOOP;
16 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. 0.00 seconds

```

```

1  DECLARE
2      v_input_string  VARCHAR2(10) := '23146579';
3      v_odd_count     NUMBER := 0;
4      v_even_count    NUMBER := 0;
5  BEGIN
6      FOR i IN 1..LENGTH(v_input_string) LOOP
7          IF SUBSTR(v_input_string, i, 1) IN ('1', '3', '5', '7', '9') THEN
8              v_odd_count := v_odd_count + 1;
9          ELSIF SUBSTR(v_input_string, i, 1) IN ('0', '2', '4', '6', '8') THEN
10             v_even_count := v_even_count + 1;
11         END IF;
12     END LOOP;
13     DBMS_OUTPUT.PUT_LINE('Number of odd digits: ' || v_odd_count);
14     DBMS_OUTPUT.PUT_LINE('Number of even digits: ' || v_even_count);
15 END;

```

```

DECLARE    v_input_string  VARCHAR2(10) := '23146579';    v_odd_count    NU
1) IN ('1', '3', '5', '7', '9') THEN    v_odd_count := v_odd_count +
END IF;    END LOOP;    DBMS_OUTPUT.PUT_LINE('Number of odd digits: ' || v_c

Number of odd digits: 5
Number of even digits: 3

Statement processed. 0.00 seconds

```

CALCULATOR MODEL:


```

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;
DBMS_OUTPUT.PUT_LINE('THE MINIMUM VALUE: '||Z);
END;
PROCEDURE FINDSUM(X IN NUMBER, Y IN NUMBER, Z OUT NUMBER )IS
BEGIN
Z:=X+Y;
DBMS_OUTPUT.PUT_LINE('THE SUM VALUE: '||Z);
END;
PROCEDURE FINDSUB(X IN NUMBER, Y IN NUMBER, Z OUT NUMBER )IS
BEGIN
Z:=X-Y;
DBMS_OUTPUT.PUT_LINE('THE SUB VALUE: '||Z);
END;
PROCEDURE FINDMUL(X IN NUMBER, Y IN NUMBER, Z OUT NUMBER )IS
BEGIN
Z:=X*Y;
DBMS_OUTPUT.PUT_LINE('THE MULTIPLICATION VALUE: '||Z);
END;
PROCEDURE FINDDIV(X IN NUMBER, Y IN NUMBER, Z OUT NUMBER ) IS
BEGIN
Z:=X / Y;
DBMS_OUTPUT.PUT_LINE('THE DIV VALUE: '||Z);
END;
BEGIN
A:=4;
B:=2;
FINDMIN(A,B,C);
FINDSUM(A,B,C);
FINDSUB(A,B,C);
FINDMUL(A,B,C);
FINDDIV(A,B,C);
END;

```

FINDMUL(A,B,C); FINDDIV(A,B,C); END;

THE MINIMUM VALUE: 2

THE SUM VALUE: 6

THE SUB VALUE: 2

THE MULTIPLICATION VALUE: 8

THE DIV VALUE: 2

Statement processed. 0.00 seconds

Factorial using function:

```
1 DECLARE
2     num number;
3     factorial number;
4
5 FUNCTION fact(x number)
6 RETURN number
7 IS
8     f number;
9 BEGIN
10     IF x=0 THEN
11         f := 1;
12     ELSE
13         f := x * fact(x-1);
14     END IF;
15 RETURN f;
16 END;
17
18 BEGIN
19     num:= 6;
20     factorial := fact(num);
21     dbms_output.put_line(' Factorial ' || num || ' is ' || factorial);
22 END;
```

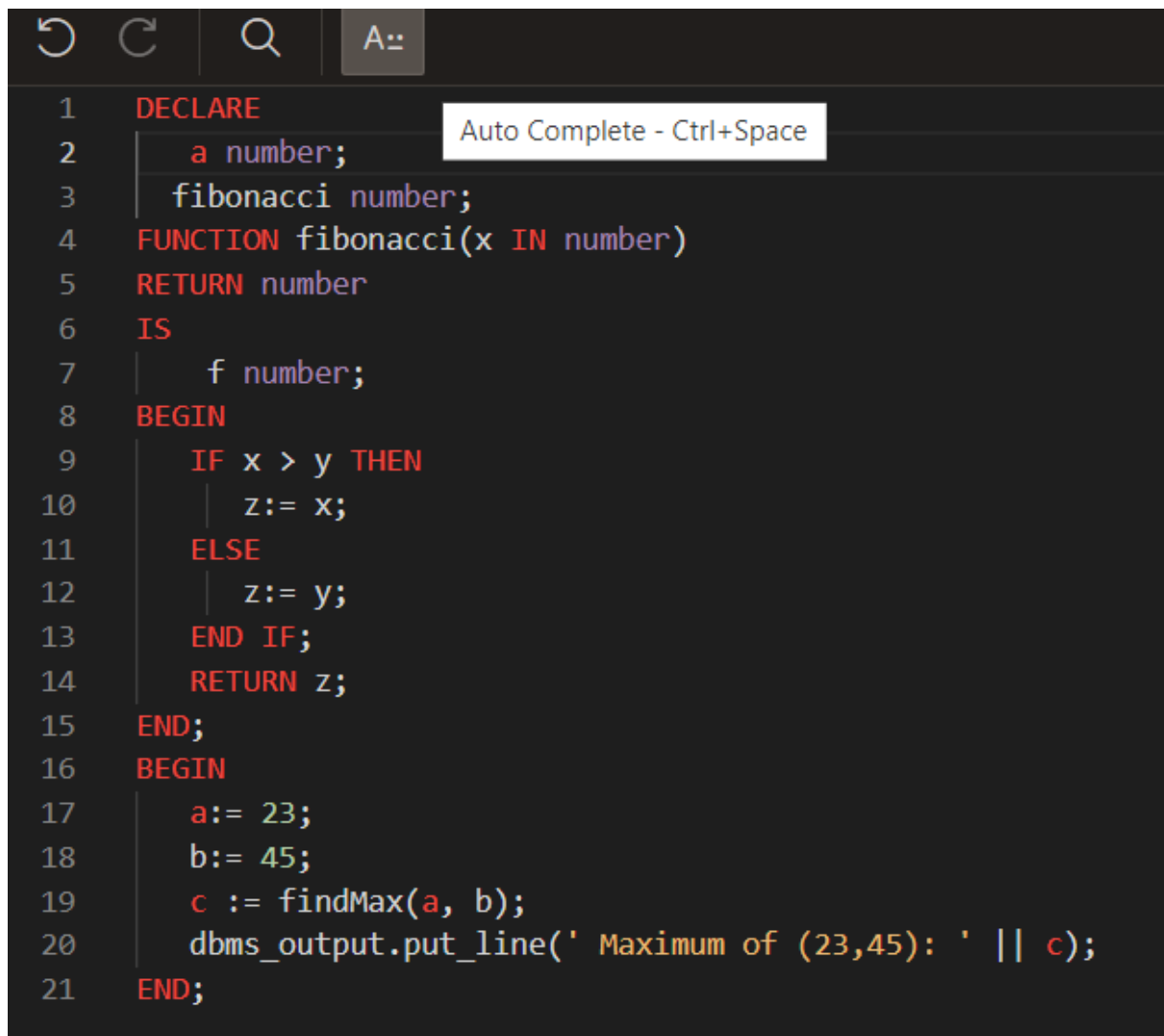
Maximum of two numbers using using function:

```
1 DECLARE
2     a number;
3     b number;
4     c number;
5 FUNCTION findMax(x IN number, y IN number)
6 RETURN number
7 IS
8     z number;
9 BEGIN
10    IF x > y THEN
11        z:= x;
12    ELSE
13        z:= y;
14    END IF;
15    RETURN z;
16 END;
17 BEGIN
18     a:= 23;
19     b:= 45;
20     c := findMax(a, b);
21     dbms_output.put_line(' Maximum of (23,45): ' || c);
22 END;
```

```
DECLARE      a number;      b number;      c number;  FUNCTION
RETURN z; END; BEGIN      a:= 23;      b:= 45;      c := f
```

Maximum of (23,45): 45

Statement processed. 0.00 seconds



The image shows a code editor with a dark theme. At the top, there is a toolbar with icons for undo, redo, search, and a dropdown menu currently showing 'A:'. Below the toolbar, the code is written in a light-colored font. The code defines a function named 'fibonacci' and then calls it within a larger block. An auto-complete tooltip is visible above the 'a' variable on line 2, displaying the text 'Auto Complete - Ctrl+Space'. The code is as follows:

```
1 DECLARE
2     a number;
3     fibonacci number;
4 FUNCTION fibonacci(x IN number)
5 RETURN number
6 IS
7     f number;
8 BEGIN
9     IF x > y THEN
10         z:= x;
11     ELSE
12         z:= y;
13     END IF;
14     RETURN z;
15 END;
16 BEGIN
17     a:= 23;
18     b:= 45;
19     c := findMax(a, b);
20     dbms_output.put_line(' Maximum of (23,45): ' || c);
21 END;
```

Fibonacci series using function:

```
DECLARE
num NUMBER;
fibonacci_number NUMBER;
FUNCTION fib(x NUMBER)
RETURN NUMBER
IS
    f NUMBER;
BEGIN
    IF x = 0 THEN
        f := 0;
    ELSIF x = 1 THEN
        f := 1;
    ELSE
        f := fib(x-1) + fib(x-2);
    END IF;
    RETURN f;
END;
BEGIN
    num := 6;
    fibonacci_number := fib(num);
    dbms_output.put_line('Fibonacci number ' || num || ' is ' || fibonacci_number);
END;
```


```
DECLARE      num NUMBER;      fibonacci_number NUMBER;
f := 1;      ELSE      f := fib(x-1) + fib(x-2);
|| ' is ' || fibonacci_number); END;
```

Fibonacci number 6 is 8

Statement processed. 0.01 seconds

INSERT THE DATA EXPLICITLY:

```
1  INSERT INTO EMPLOYEEESI (EMP_ID ,FIRST_NAME ,LAST_NAME ,EMAIL)
2  VALUES(434532, 'KORNANA', 'GOUTHAM', 'GOUTHAM@EMAIL.COM');
3
```


Query Count Rows Insert Row Load Data				
EDIT	EMP_ID	FIRST_NAME	LAST_NAME	EMAIL
	434532	KORNANA	GOUTHAM	GOUTHAM@EMAIL.COM

INSERT THE DATA IMPLICITLY:

```

1  INSERT INTO EMPLOYEEESI
2  VALUES(434532, 'MILKY', 'MANASA', 'MANASA@EMAIL.COM');
3

```

EDIT	EMP_ID	FIRST_NAME	LAST_NAME	EMAIL
	434532	MILKY	MANASA	MANASA@EMAIL.COM

UPDATE:

```

1  UPDATE EMPLOYEEESI
2  SET EMAIL='SAVEETHA@MAIL.COM'
3  WHERE FIRST_NAME='KORNANA';



```

Query

Count Rows

Insert Row

Load Data

EDIT	EMP_ID	FIRST_NAME	LAST_NAME	EMAIL
	434532	MILKY	MANASA	MANASA@EMAIL.COM
	434532	KORNANA	GOUTHAM	SAVEETHA@MAIL.COM

DELETING THE DATA SYNTAX;

```

DELETE FROM EMPLOYEEESI
WHERE FIRST_NAME='KORNANA';

```

EDIT	EMP_ID	FIRST_NAME	LAST_NAME	EMAIL
	434532	MILKY	MANASA	MANASA@EMAIL.COM

MERGE :

```

1  MERGE INTO EMPLO E
2  USING DEP D
3  ON (E.DEPARTMENT_ID=D.DEPARTMENT_ID)
4  WHEN MATCHED THEN
5  UPDATE SET E.FIRST_NAME=D.DEPARTMENT_ID*05;






```

Query

Count Rows

Insert Row

Load Data

EDIT	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
	1	50	KORNANA	10
	2	100	EDUKONDALU	20
	3	100	MAHESH	20
	5	150	KONIDELA	30
	4	150	RAJ	30