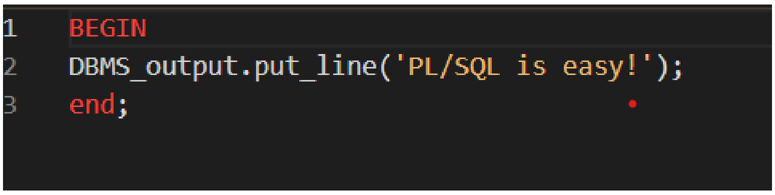
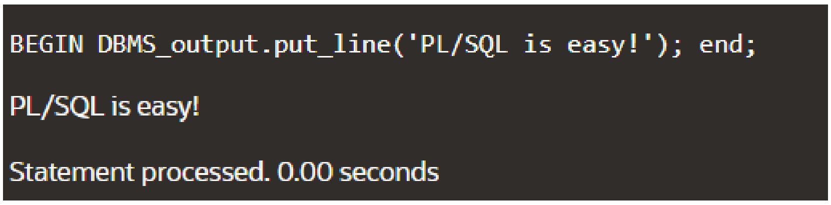
EXECUTABLE:

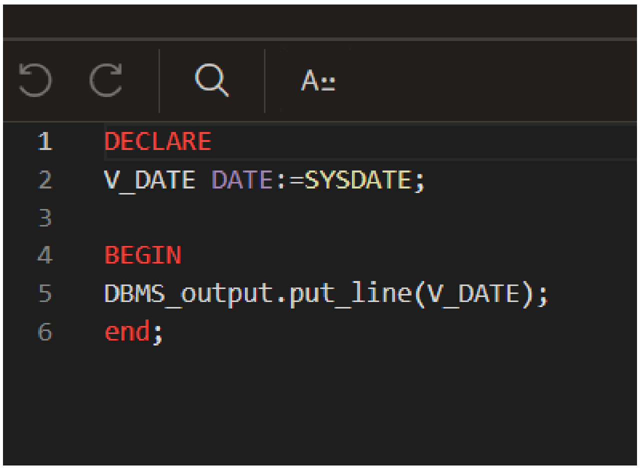


```
BEGIN
DBMS_output.put_line('PL/SQL is easy!');
end;
```

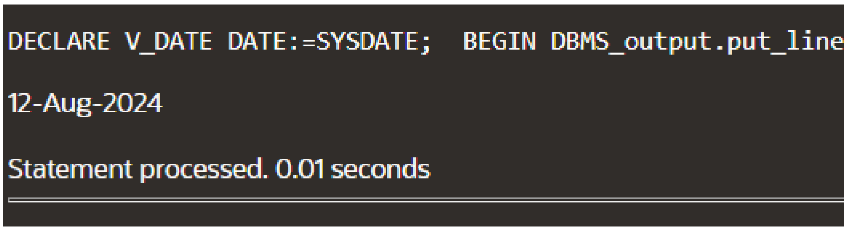


```
BEGIN DBMS_output.put_line('PL/SQL is easy!'); end;

PL/SQL is easy!

Statement processed. 0.00 seconds
```

EXECUTABLE AND DECLARATION:

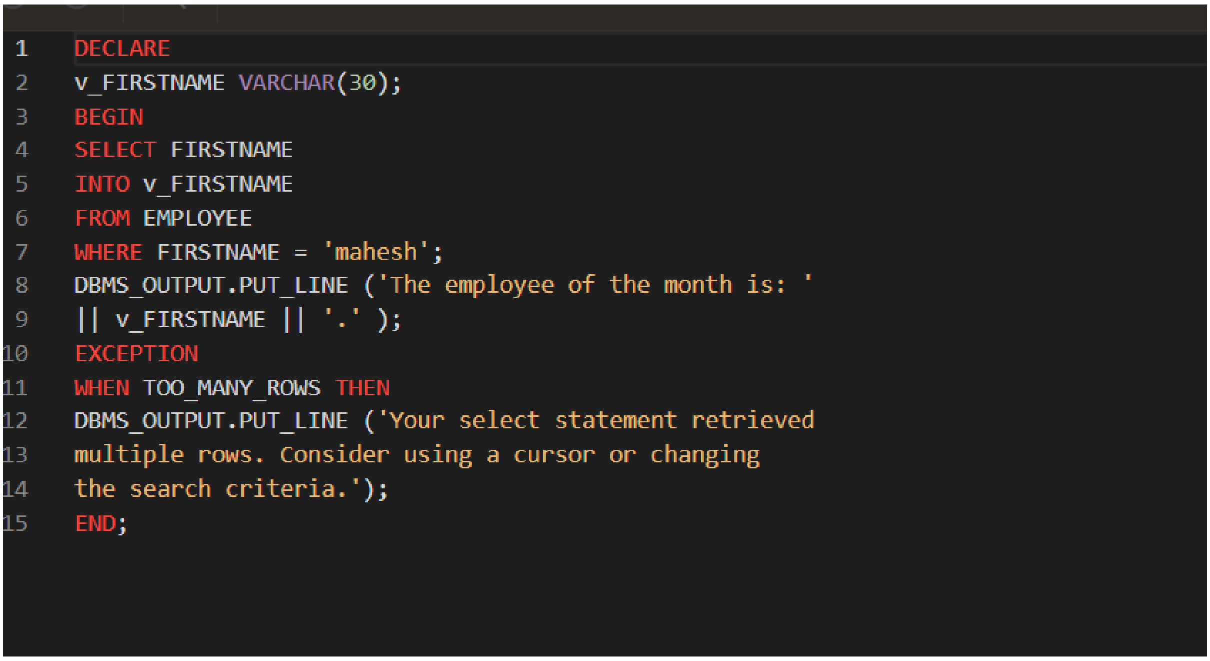


```
DECLARE
V_DATE DATE:=SYSDATE;

BEGIN
DBMS_output.put_line(V_DATE);
end;
```

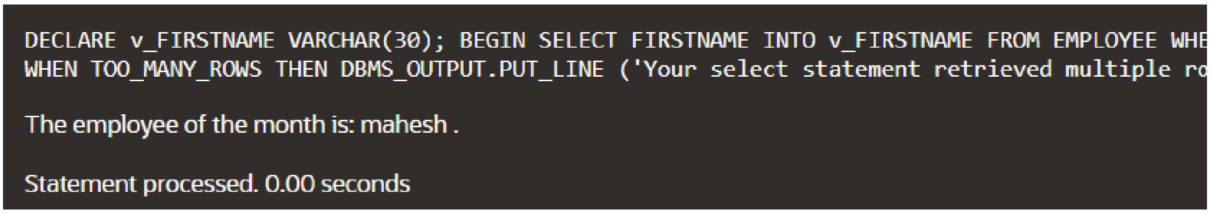


```
DECLARE V_DATE DATE:=SYSDATE;  BEGIN DBMS_output.put_line

12-Aug-2024

Statement processed. 0.01 seconds
```

EXCUTABLE ,DECLARATION AND EXCEPTION:

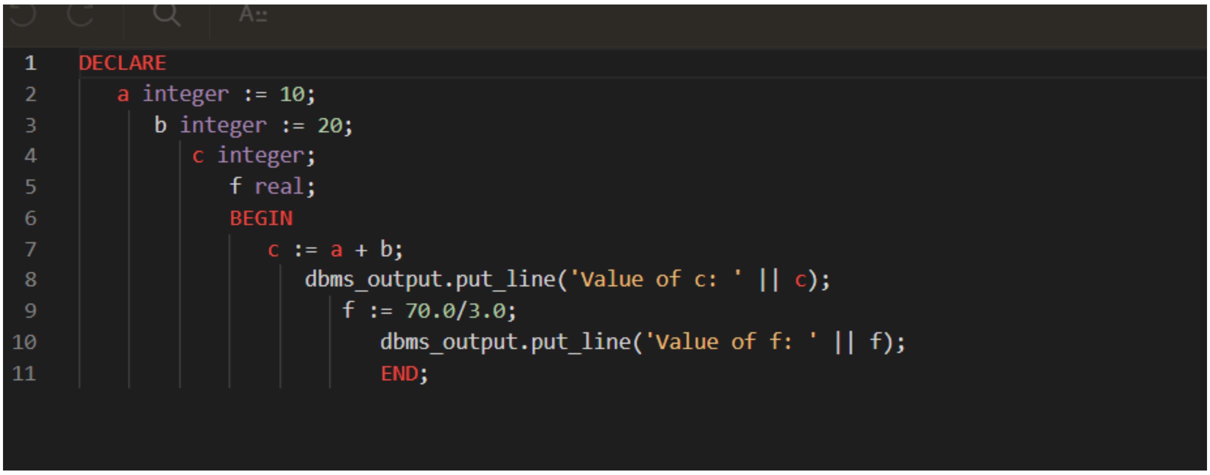


```
DECLARE
v_FIRSTNAME VARCHAR(30);
BEGIN
SELECT FIRSTNAME
INTO v_FIRSTNAME
FROM EMPLOYEE
WHERE FIRSTNAME = 'mahesh';
DBMS_OUTPUT.PUT_LINE ('The employee of the month is: '
|| v_FIRSTNAME || '.' );
EXCEPTION
WHEN TOO_MANY_ROWS THEN
DBMS_OUTPUT.PUT_LINE ('Your select statement retrieved
multiple rows. Consider using a cursor or changing
the search criteria.');
END;
```

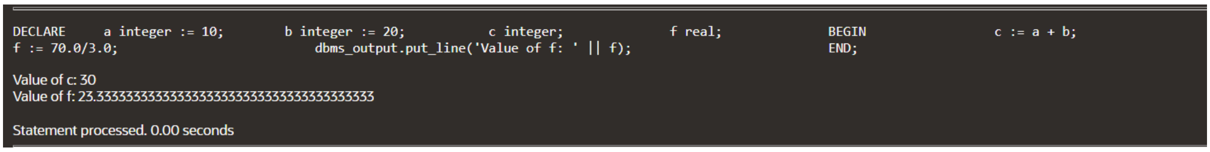


```
DECLARE v_FIRSTNAME VARCHAR(30); BEGIN SELECT FIRSTNAME INTO v_FIRSTNAME FROM EMPLOYEE WHE
WHEN TOO_MANY_ROWS THEN DBMS_OUTPUT.PUT_LINE ('Your select statement retrieved multiple ro

The employee of the month is: mahesh .

Statement processed. 0.00 seconds
```

ADDITION:

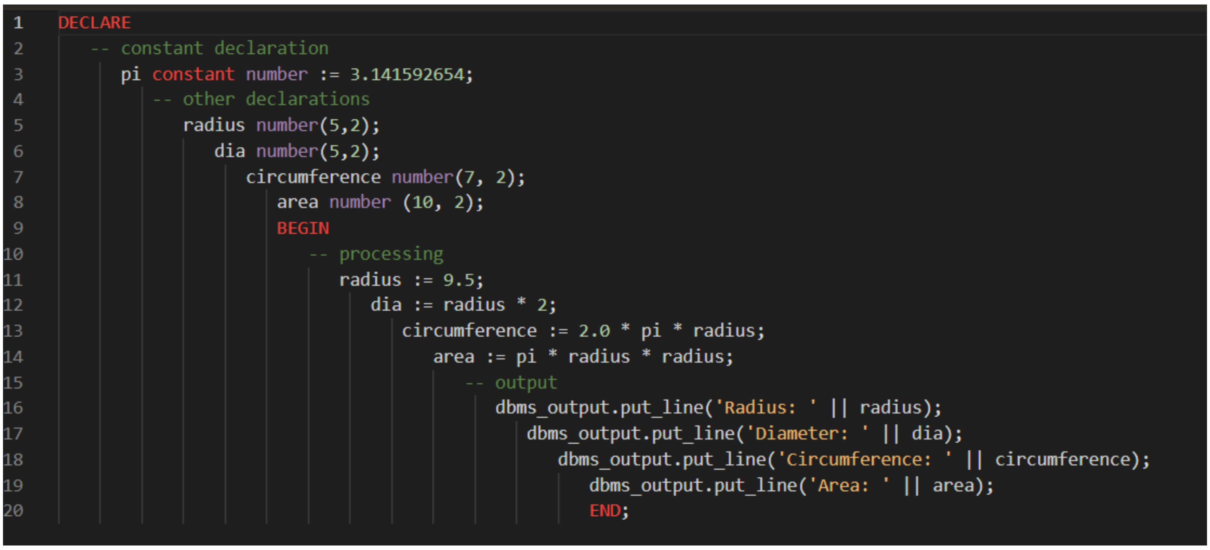


```
DECLARE
   a integer := 10;
      b integer := 20;
         c integer;
            f real;
            BEGIN
               c := a + b;
                  dbms_output.put_line('Value of c: ' || c);
                     f := 70.0/3.0;
                        dbms_output.put_line('Value of f: ' || f);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

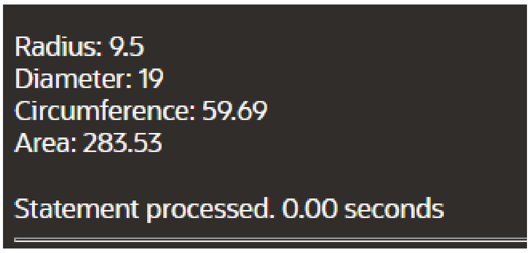


```
DECLARE      a integer := 10;         b integer := 20;            c integer;              f real;               BEGIN                c := a + b;
f := 70.0/3.0;                          dbms_output.put_line('Value of f: ' || f);             END;

Value of c: 30
Value of f: 23.33333333333333333333333333333333333333

Statement processed. 0.00 seconds
```

AREA:

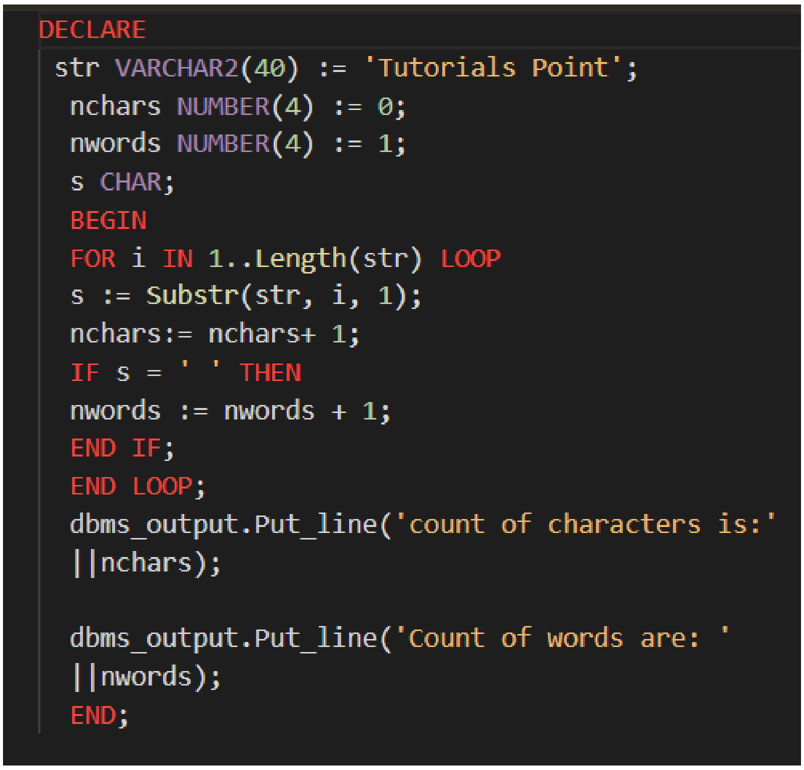


```
DECLARE
   -- constant declaration
      pi constant number := 3.141592654;
         -- other declarations
            radius number(5,2);
               dia number(5,2);
                  circumference number(7, 2);
                     area number (10, 2);
                     BEGIN
                        -- 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;
```

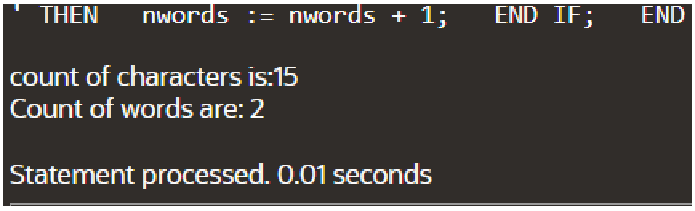


```
Radius: 9.5
Diameter: 19
Circumference: 59.69
Area: 283.53

Statement processed. 0.00 seconds
```

COUNT OF WORDS:

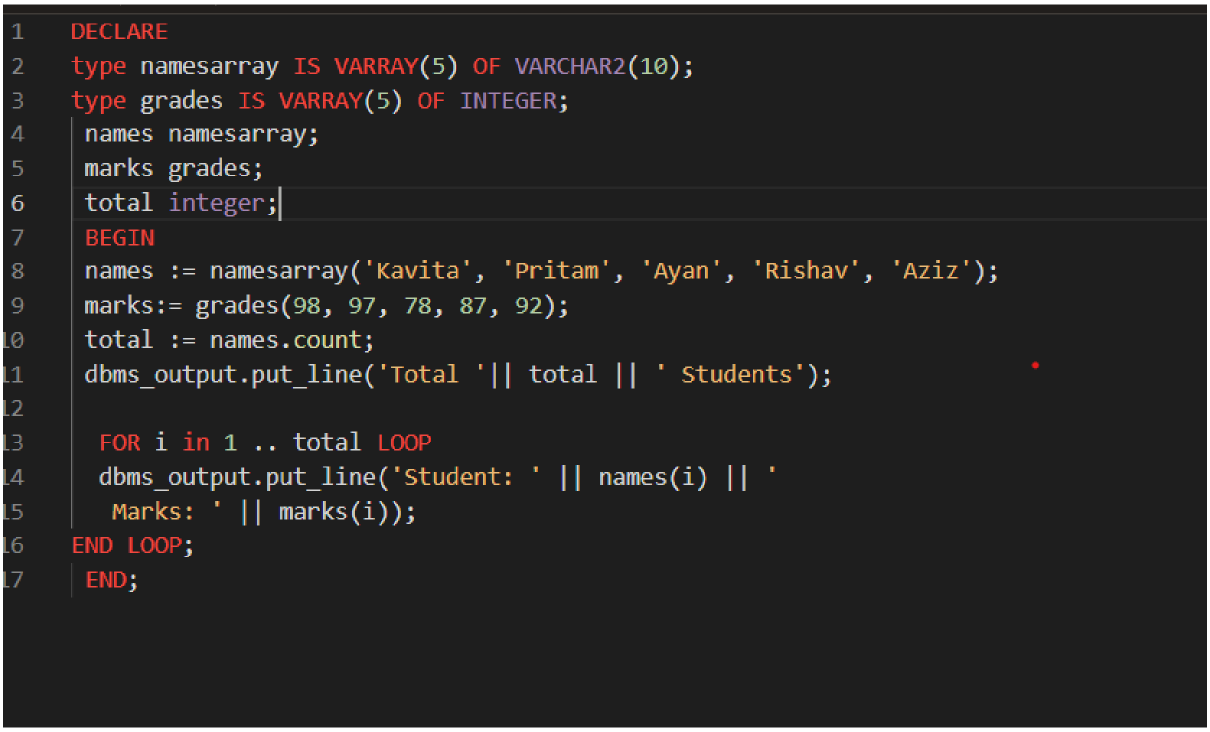


```
DECLARE
 str VARCHAR2(40) := 'Tutorials Point';
  nchars NUMBER(4) := 0;
  nwords NUMBER(4) := 1;
  s CHAR;
  BEGIN
  FOR i IN 1..Length(str) LOOP
  s := Substr(str, i, 1);
  nchars:= nchars+ 1;
  IF s = ' ' THEN
  nwords := nwords + 1;
  END IF;
  END LOOP;
  dbms_output.Put_line('count of characters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

  dbms_output.Put_line('Count of words are: '
  ||nwords);
  END;
```

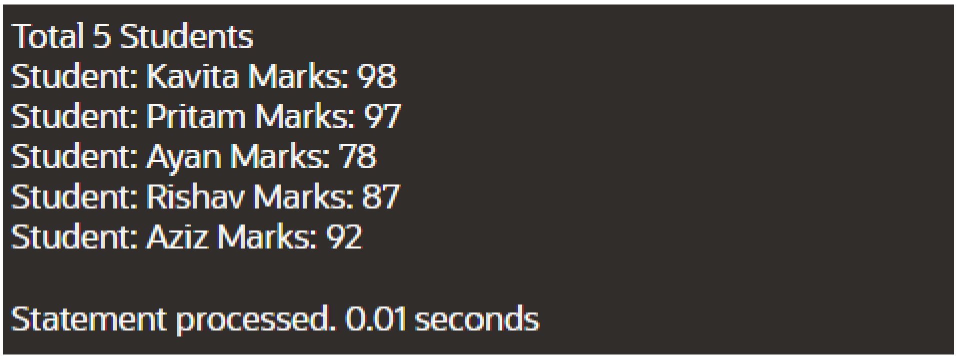


```
' THEN    nwords := nwords + 1;    END IF;    END

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

Statement processed. 0.01 seconds
```

STUDENT MARKS:

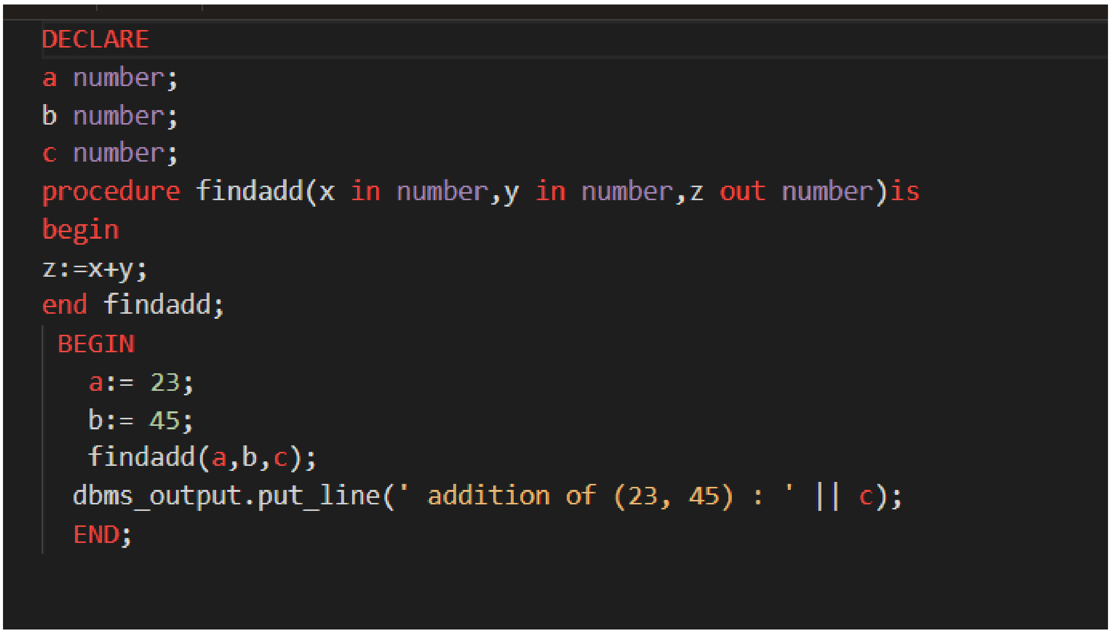


```
DECLARE
type namesarray IS VARRAY(5) OF VARCHAR2(10);
type grades IS VARRAY(5) OF INTEGER;
 names namesarray;
 marks grades;
 total integer;
 BEGIN
 names := namesarray('Kavita', 'Pritam', 'Ayan', 'Rishav', 'Aziz');
 marks:= grades(98, 97, 78, 87, 92);
 total := names.count;
 dbms_output.put_line('Total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

  FOR i in 1 .. total LOOP
  dbms_output.put_line('Student: ' || names(i) || '
   Marks: ' || marks(i));
END LOOP;
 END;
```

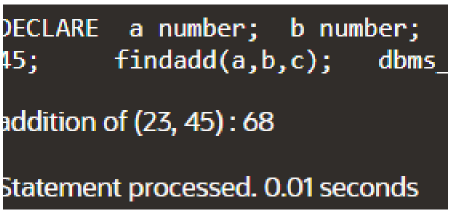


```
Total 5 Students
Student: Kavita Marks: 98
Student: Pritam Marks: 97
Student: Ayan Marks: 78
Student: Rishav Marks: 87
Student: Aziz Marks: 92

Statement processed. 0.01 seconds
```

Function addition:

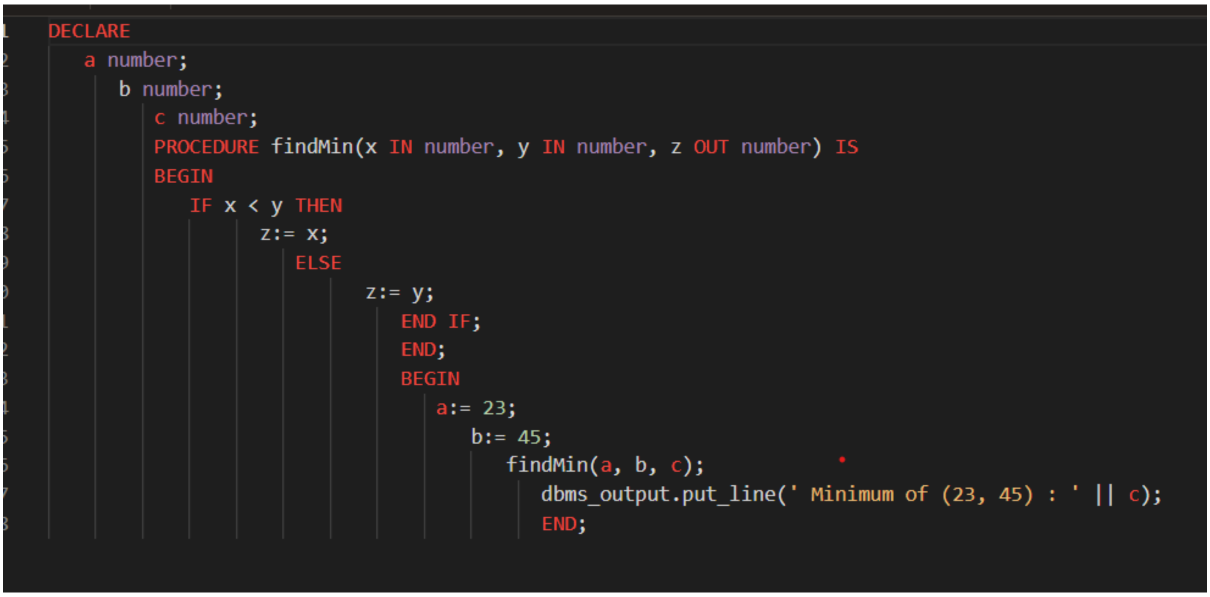


```
DECLARE
a number;
b number;
c number;
procedure findadd(x in number,y in number,z out number)is
begin
z:=x+y;
end findadd;
 BEGIN
   a:= 23;
   b:= 45;
   findadd(a,b,c);
  dbms_output.put_line(' addition of (23, 45) : ' || c);
  END;
```

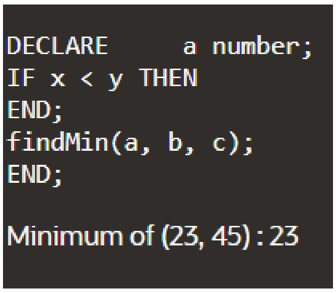


```
DECLARE    a number;    b number;
45;     findadd(a,b,c);    dbms_

addition of (23, 45) : 68

Statement processed. 0.01 seconds
```

Minimum:

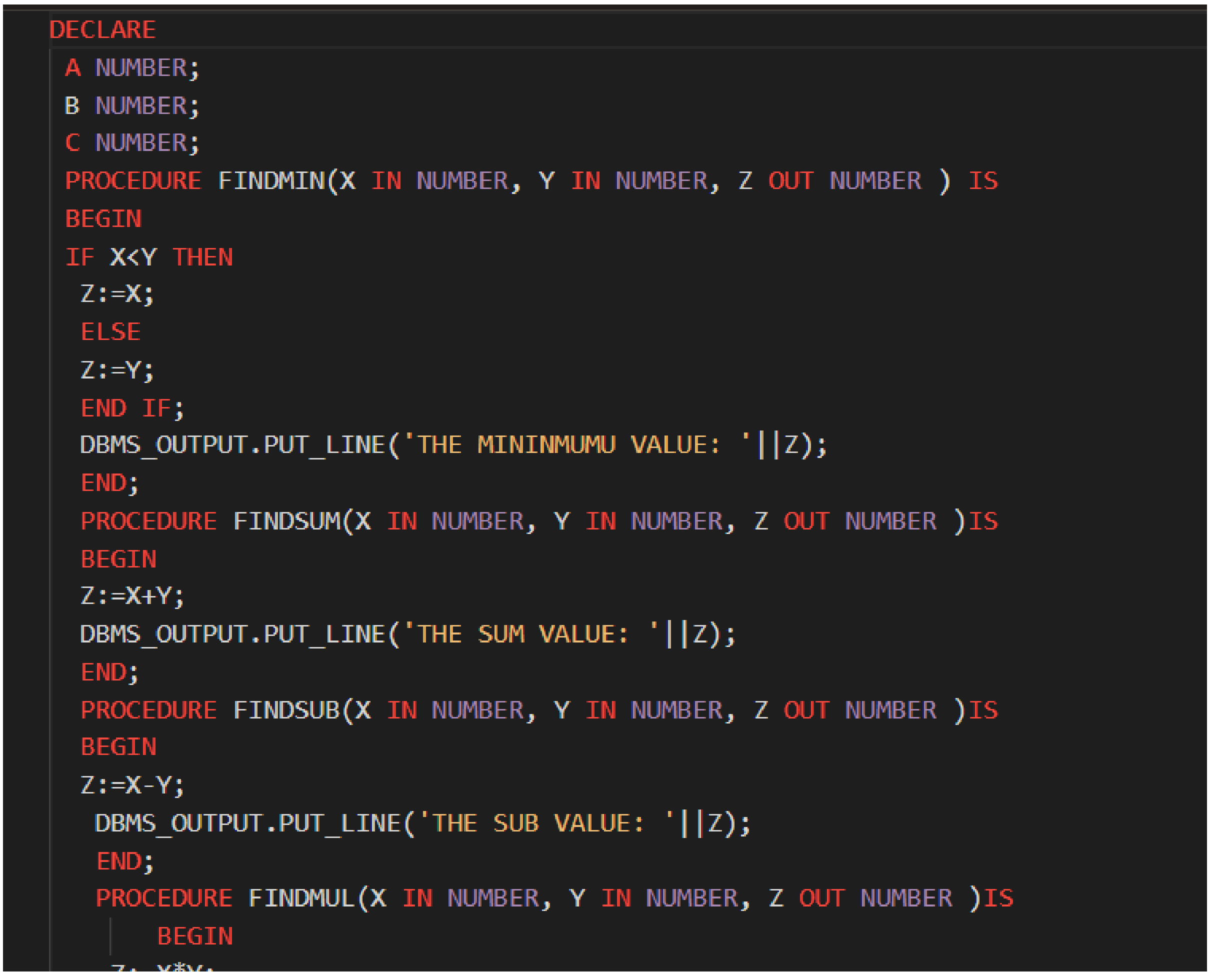


```
DECLARE
   a number;
      b number;
         c number;
         PROCEDURE findMin(x IN number, y IN number, z OUT number) IS
         BEGIN
            IF x < y THEN
                  z:= x;
                     ELSE
                           z:= y;
                              END IF;
      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

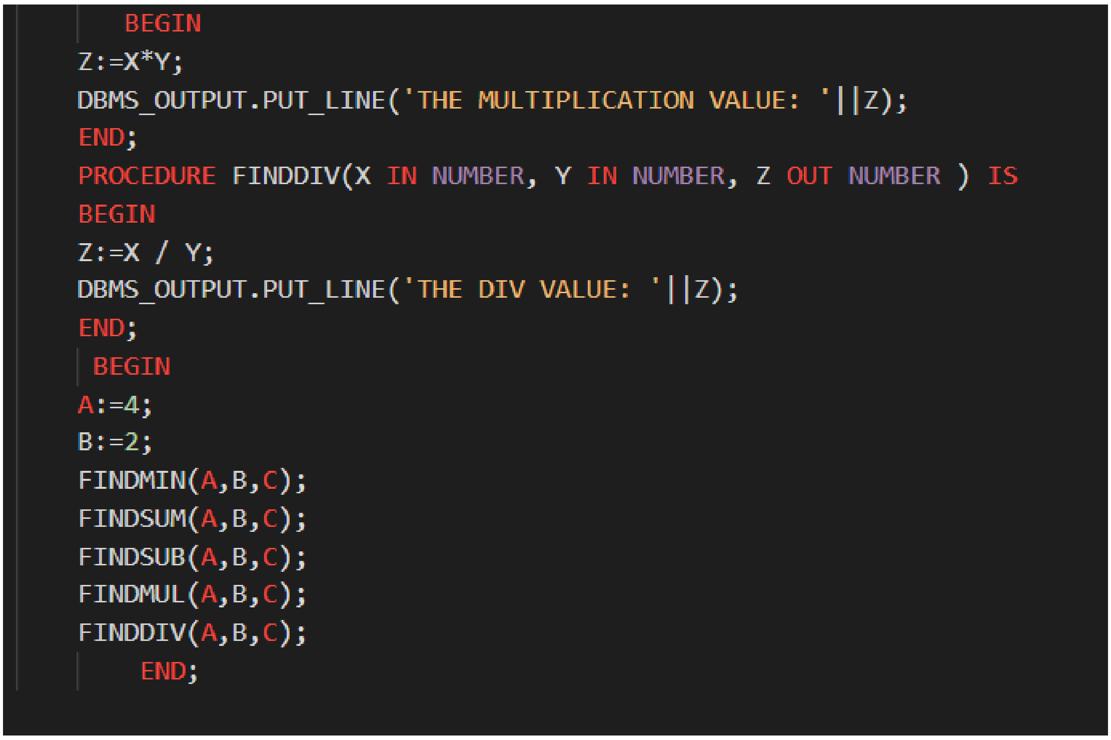


```
DECLARE      a number;
IF x < y THEN
END;
findMin(a, b, c);
END;

Minimum of (23, 45) : 23
```

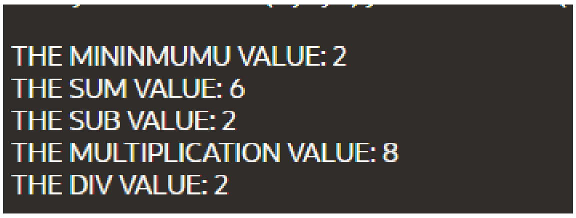
Calculator:



```
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 MININMUMU 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;
```

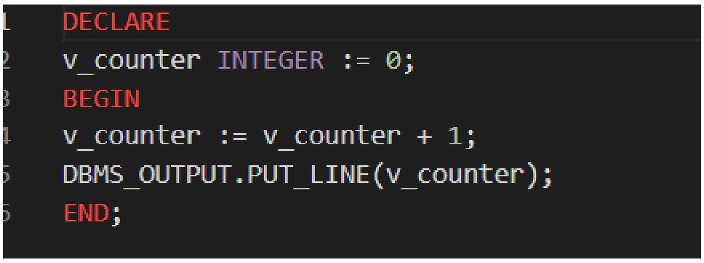


```
    BEGIN
Z:=X*Y;
DBMS_OUTPUT.PUT_LINE('THE MULTIPLICATION VALUE: '||Z);
END;
PROCEDURE FINDDIV(X IN NUMBER, Y IN NUMBER, Z OUT NUMBER ) IS
BEGIN
Z:=X / Y;
DBMS_OUTPUT.PUT_LINE('THE DIV VALUE: '||Z);
END;
 BEGIN
A:=4;
B:=2;
FINDMIN(A,B,C);
FINDSUM(A,B,C);
FINDSUB(A,B,C);
FINDMUL(A,B,C);
FINDDIV(A,B,C);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

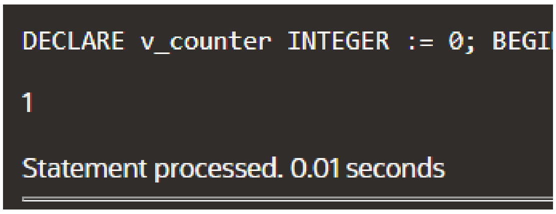


```
THE MININMUMU VALUE: 2
THE SUM VALUE: 6
THE SUB VALUE: 2
THE MULTIPLICATION VALUE: 8
THE DIV VALUE: 2
```

Initializing veriables:

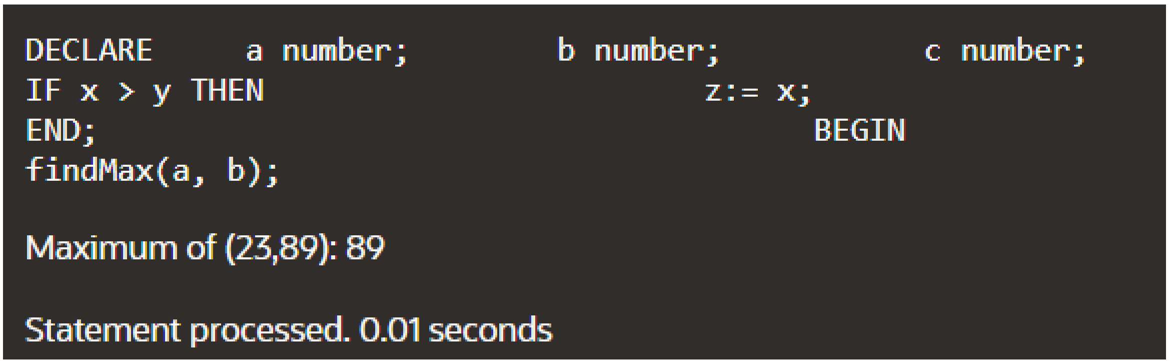


```
DECLARE
v_counter INTEGER := 0;
BEGIN
v_counter := v_counter + 1;
DBMS_OUTPUT.PUT_LINE(v_counter);
END;
```

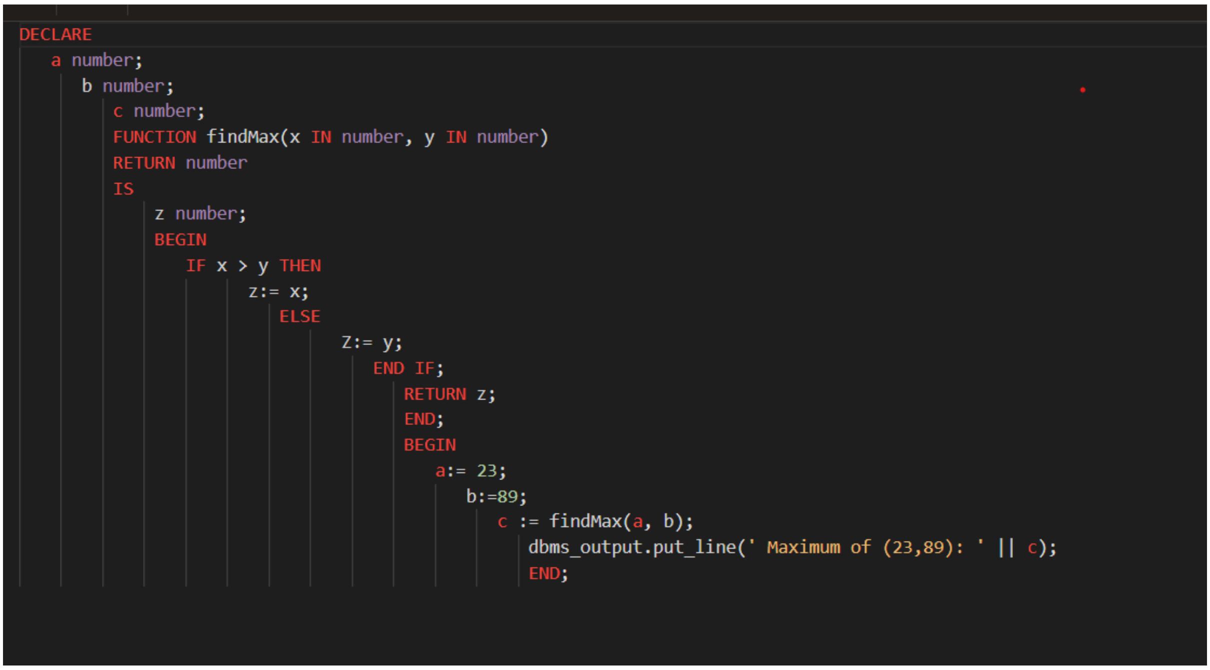


```
DECLARE v_counter INTEGER := 0; BEGI

1

Statement processed. 0.01 seconds
```

Find max using functions:

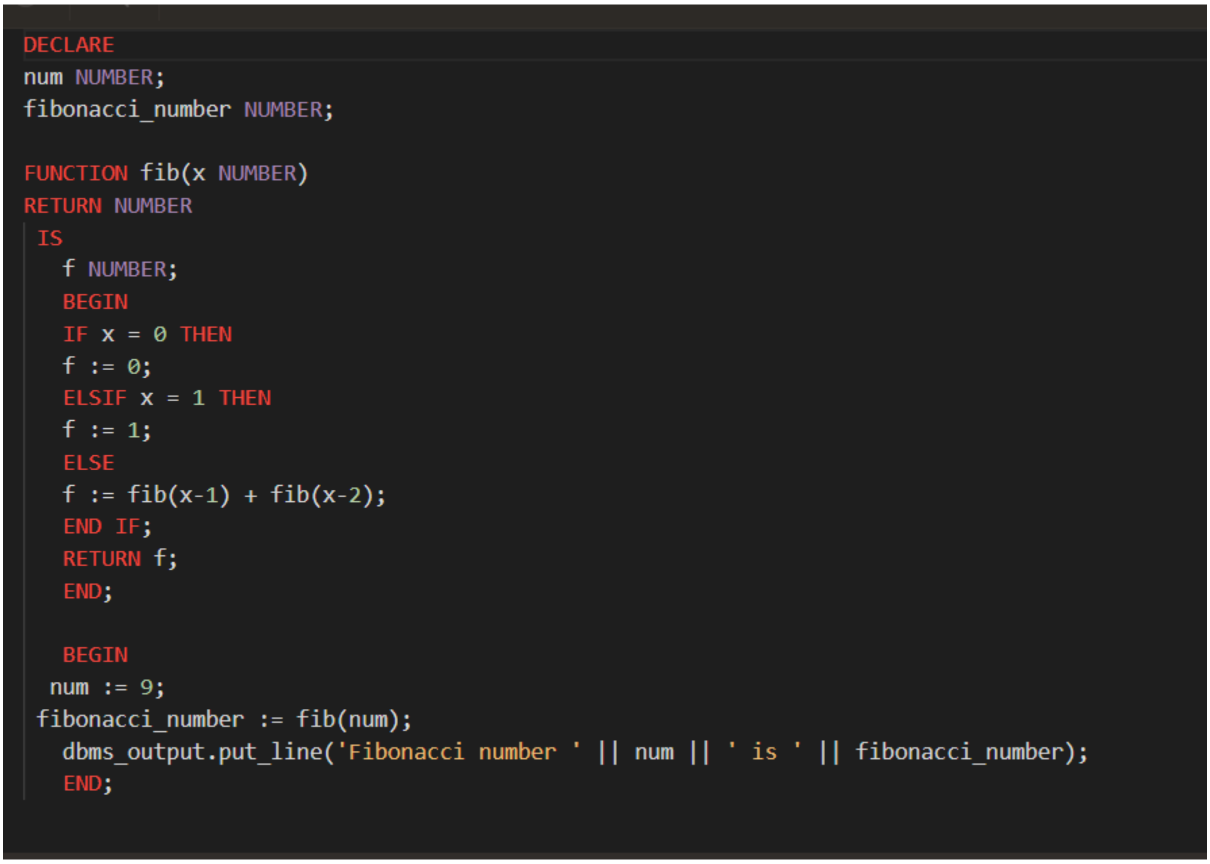


```
DECLARE      a number;         b number;          c number;
IF x > y THEN                        z:= x;
END;                                        BEGIN
findMax(a, b);

Maximum of (23,89): 89

Statement processed. 0.01 seconds
```

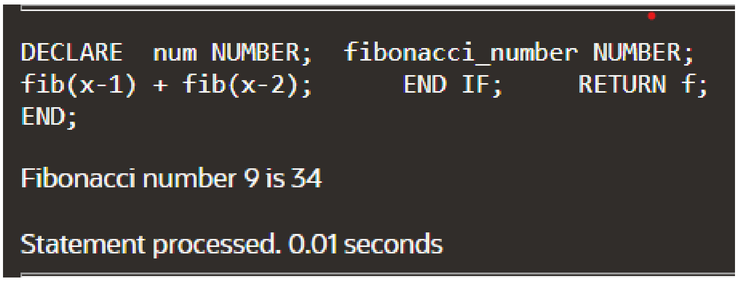


```
DECLARE
   a number;
      b number;
         c number;
         FUNCTION findMax(x IN number, y IN number)
         RETURN number
         IS
             z number;
             BEGIN
                IF x > y THEN
                      z:= x;
                         ELSE
                               Z:= y;
                                  END IF;
                                     RETURN z;
                                     END;
                                     BEGIN
                                        a:= 23;
                                           b:=89;
                                              c := findMax(a, b);
                                                 dbms_output.put_line(' Maximum of (23,89): ' || c);
                                                 END;
```

Fibonacci:

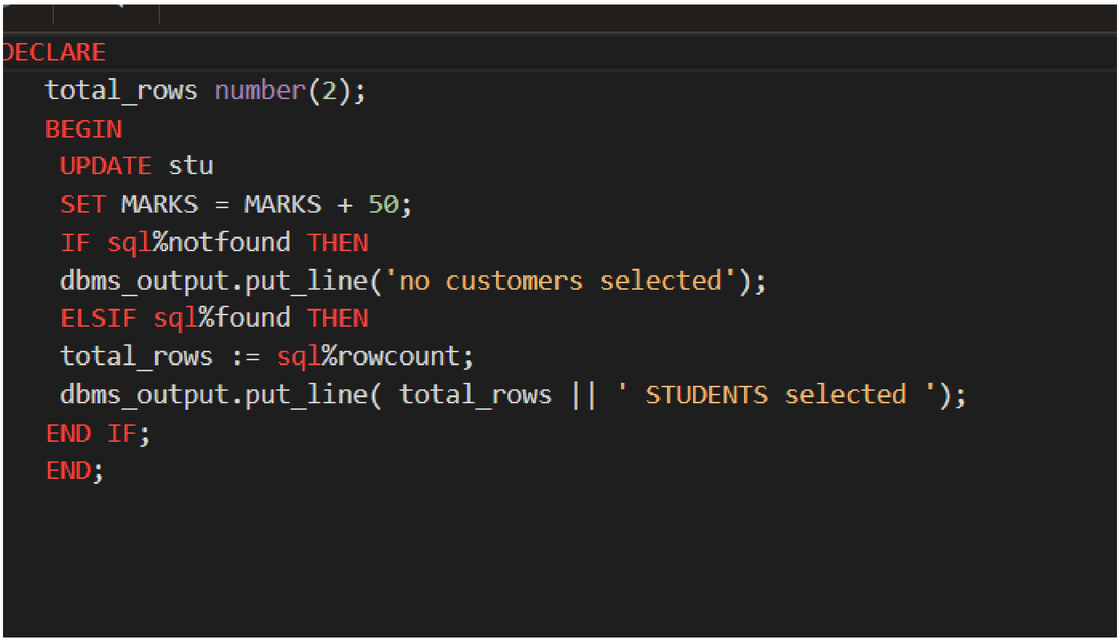


```
DECLARE
num NUMBER;
fibonacci_number NUMBER;

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 := 9;
 fibonacci_number := fib(num);
   dbms_output.put_line('Fibonacci number ' || num || ' is ' || fibonacci_number);
   END;
```

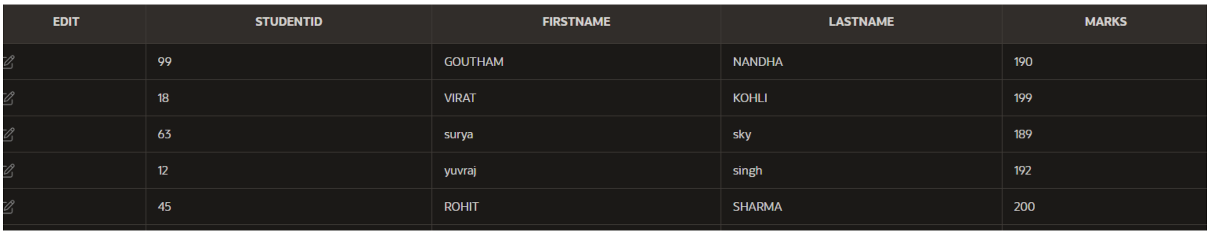


```
DECLARE   num NUMBER;   fibonacci_number NUMBER;
fib(x-1) + fib(x-2);       END IF;      RETURN f;
END;

Fibonacci number 9 is 34

Statement processed. 0.01 seconds
```

NO.OF ROWS USING FUNCTIONS:

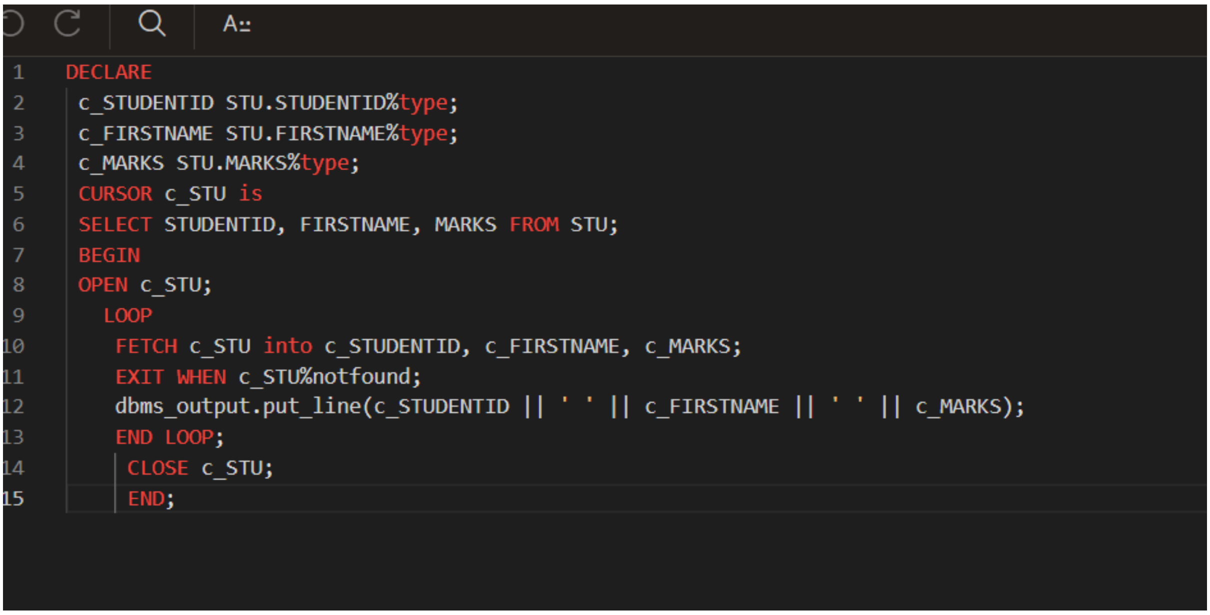


```
DECLARE
   total_rows number(2);
   BEGIN
    UPDATE stu
    SET MARKS = MARKS + 50;
    IF sql%notfound THEN
    dbms_output.put_line('no customers selected');
    ELSIF sql%found THEN
    total_rows := sql%rowcount;
    dbms_output.put_line( total_rows || ' STUDENTS selected ');
   END IF;
   END;
```

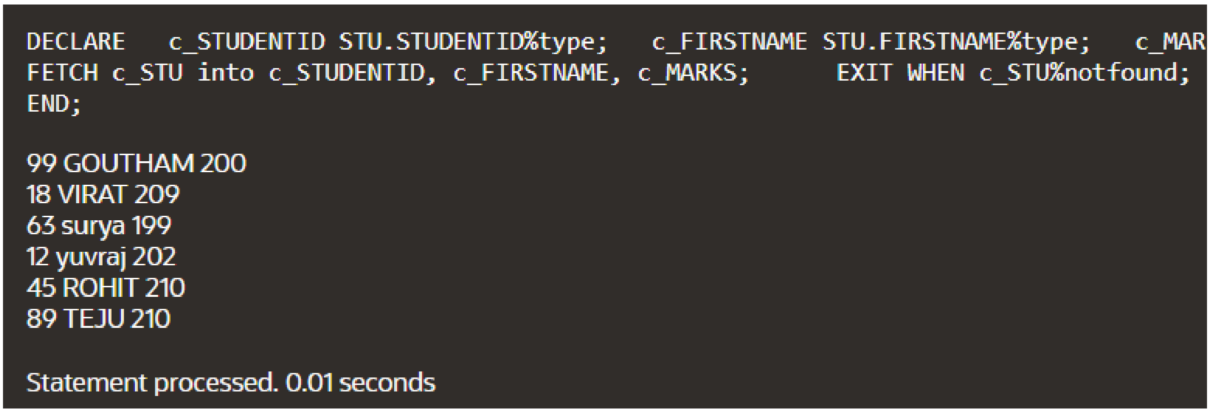


EDIT	STUDENTID	FIRSTNAME	LASTNAME	MARKS
✎	99	GOUTHAM	NANDHA	190
✎	18	VIRAT	KOHLI	199
✎	63	surya	sky	189
✎	12	yuvraj	singh	192
✎	45	ROHIT	SHARMA	200

SELECT ROWS FROM TABLE:

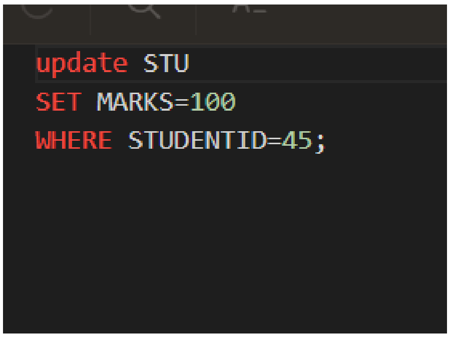


```
DECLARE
 c_STUDENTID STU.STUDENTID%type;
 c_FIRSTNAME STU.FIRSTNAME%type;
 c_MARKS STU.MARKS%type;
 CURSOR c_STU is
 SELECT STUDENTID, FIRSTNAME, MARKS FROM STU;
 BEGIN
 OPEN c_STU;
   LOOP
    FETCH c_STU into c_STUDENTID, c_FIRSTNAME, c_MARKS;
    EXIT WHEN c_STU%notfound;
    dbms_output.put_line(c_STUDENTID || ' ' || c_FIRSTNAME || ' ' || c_MARKS);
    END LOOP;
     CLOSE c_STU;
     END;
```

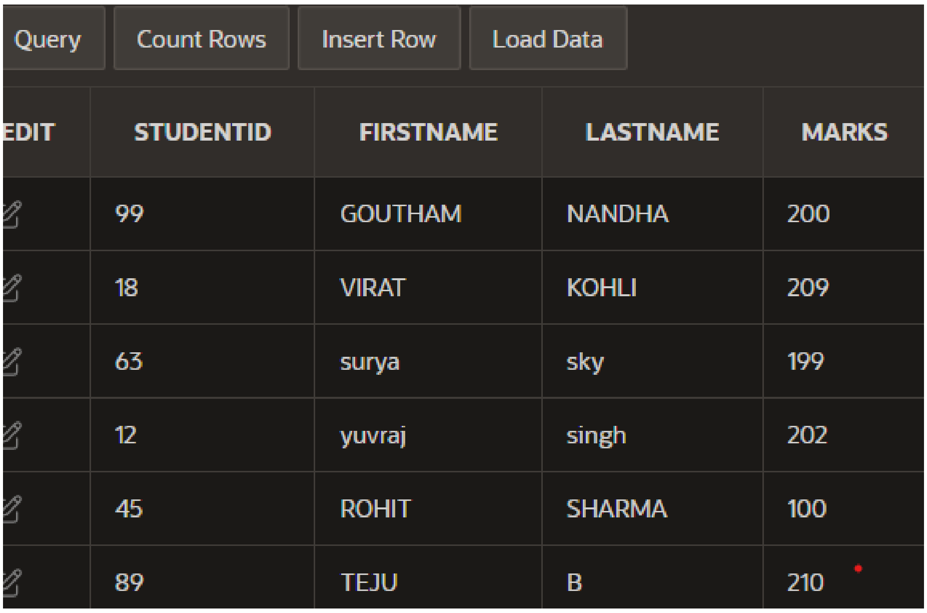


```
DECLARE    c_STUDENTID STU.STUDENTID%type;    c_FIRSTNAME STU.FIRSTNAME%type;    c_MAR
FETCH c_STU into c_STUDENTID, c_FIRSTNAME, c_MARKS;      EXIT WHEN c_STU%notfound;
END;

99 GOUTHAM 200
18 VIRAT 209
63 surya 199
12 yuvraj 202
45 ROHIT 210
89 TEJU 210

Statement processed. 0.01 seconds
```

UPDATE:

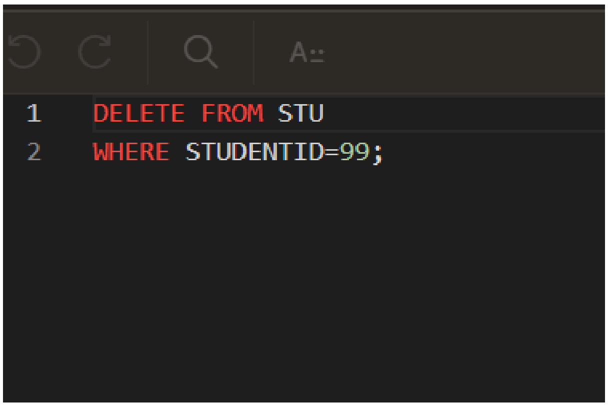


```
update STU
SET MARKS=100
WHERE STUDENTID=45;
```

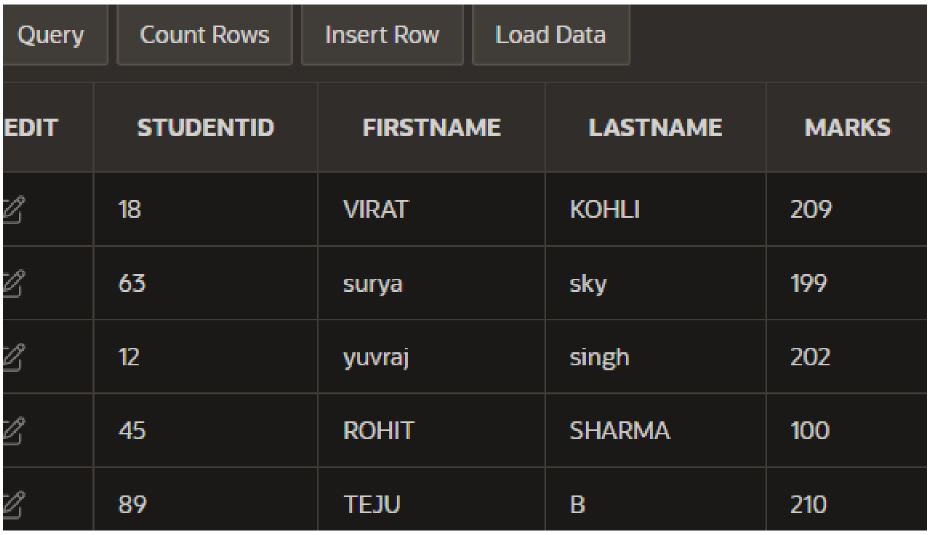


EDIT	STUDENTID	FIRSTNAME	LASTNAME	MARKS
✎	99	GOUTHAM	NANDHA	200
✎	18	VIRAT	KOHLI	209
✎	63	surya	sky	199
✎	12	yuvraj	singh	202
✎	45	ROHIT	SHARMA	100
✎	89	TEJU	B	210

DELETE:

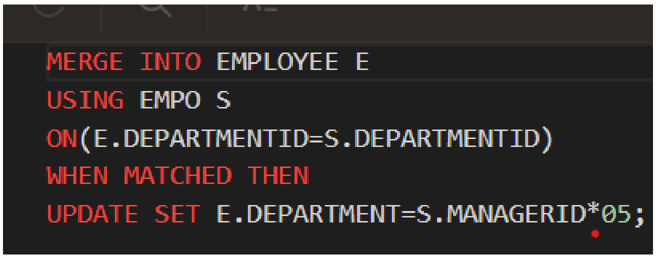


```
DELETE FROM STU
WHERE STUDENTID=99;
```

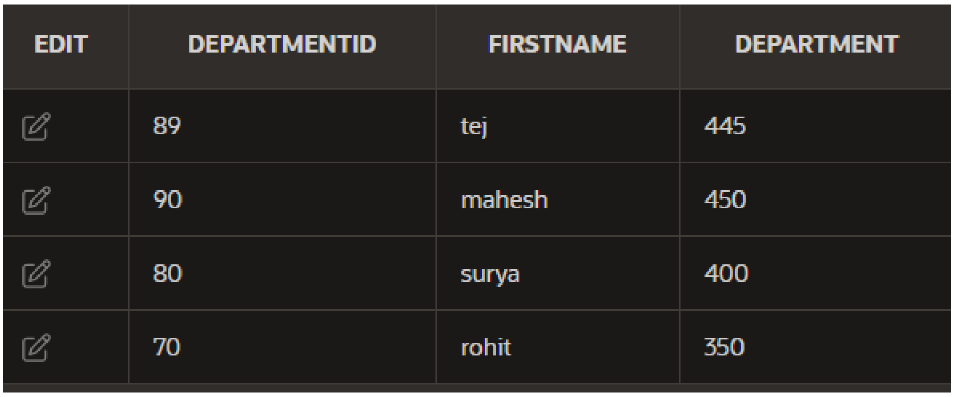


EDIT	STUDENTID	FIRSTNAME	LASTNAME	MARKS
✎	18	VIRAT	KOHLI	209
✎	63	surya	sky	199
✎	12	yuvraj	singh	202
✎	45	ROHIT	SHARMA	100
✎	89	TEJU	B	210

MERGE:

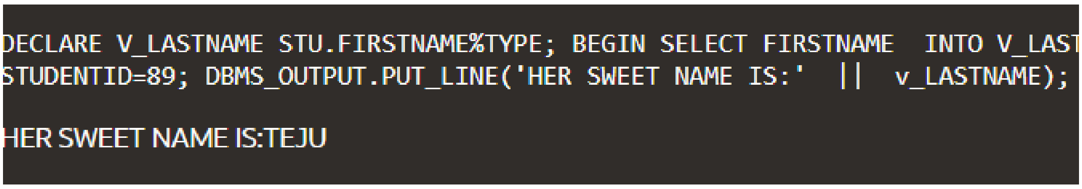


```
MERGE INTO EMPLOYEE E
USING EMPO S
ON(E.DEPARTMENTID=S.DEPARTMENTID)
WHEN MATCHED THEN
UPDATE SET E.DEPARTMENT=S.MANAGERID*05;
```



EDIT	DEPARTMENTID	FIRSTNAME	DEPARTMENT
✎	89	tej	445
✎	90	mahesh	450
✎	80	surya	400
✎	70	rohit	350

USING INTO CLAUSE:



```
DECLARE V_LASTNAME STU.FIRSTNAME%TYPE; BEGIN SELECT FIRSTNAME  INTO V_LAST
STUDENTID=89; DBMS_OUTPUT.PUT_LINE('HER SWEET NAME IS:'  ||  v_LASTNAME);

HER SWEET NAME IS:TEJU
```