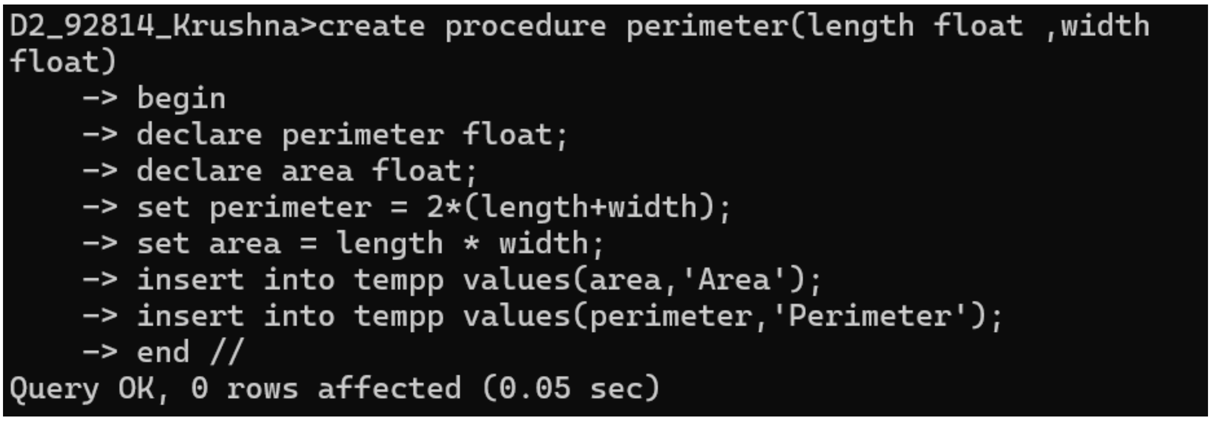
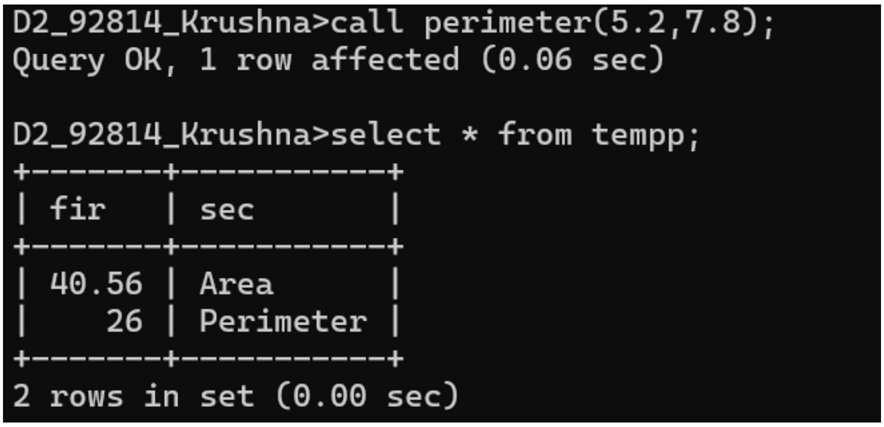
Exercise 1

1. Write a program that computes the perimeter and the area of a rectangle. Define your own values for the length and width. (Assuming that L and W are the length and width of the rectangle, Perimeter = 2*(L+W) and Area = L*W.

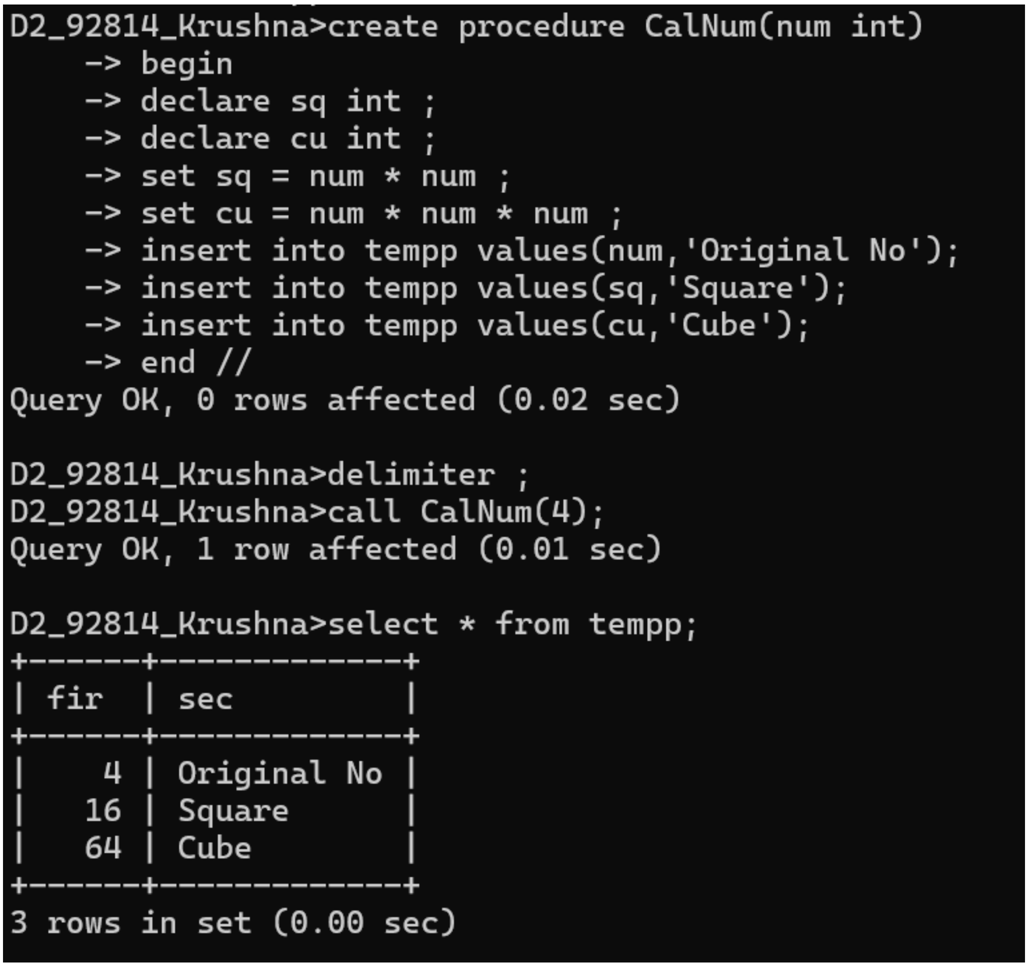


```
D2_92814_Krushna>create procedure perimeter(length float ,width
float)
    -> begin
    -> declare perimeter float;
    -> declare area float;
    -> set perimeter = 2*(length+width);
    -> set area = length * width;
    -> insert into tempp values(area,'Area');
    -> insert into tempp values(perimeter,'Perimeter');
    -> end //
Query OK, 0 rows affected (0.05 sec)
```



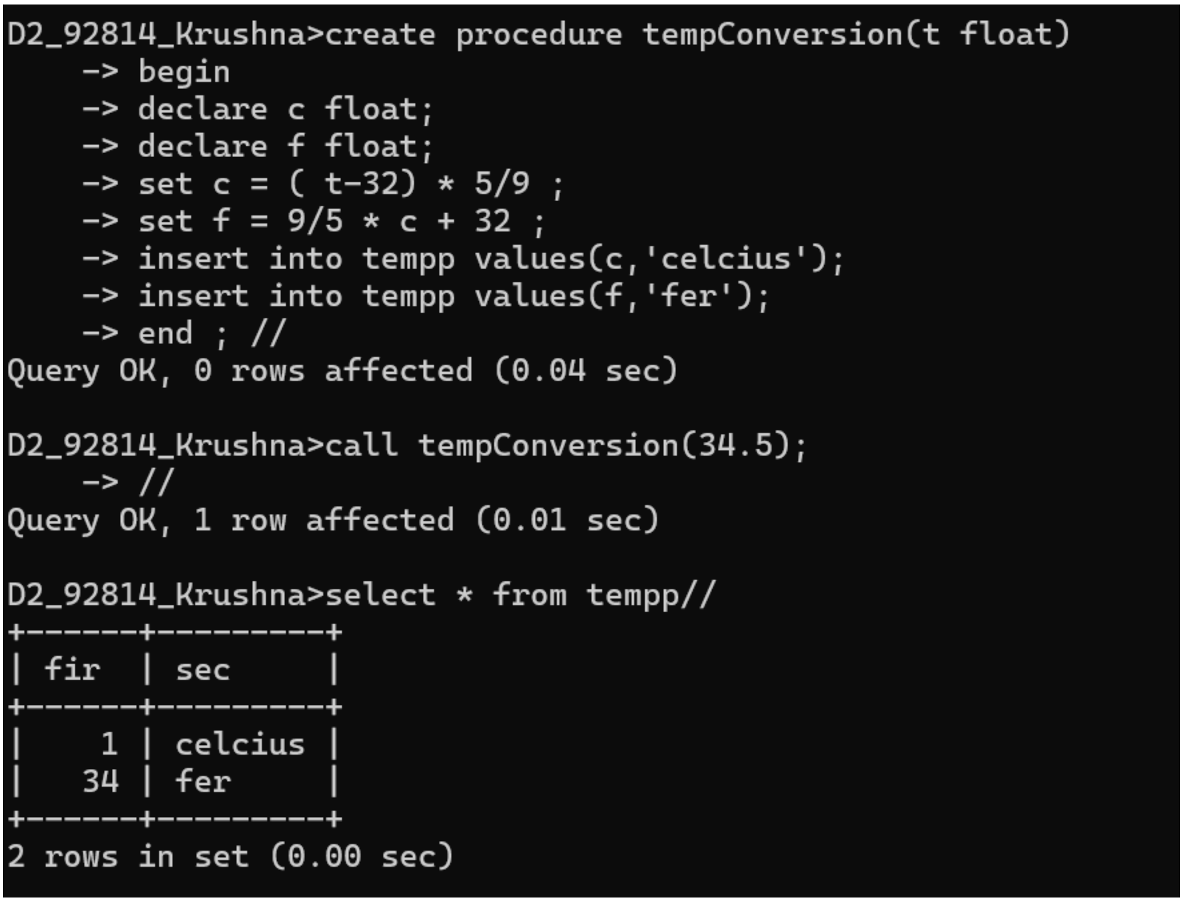
```
D2_92814_Krushna>call perimeter(5.2,7.8);
Query OK, 1 row affected (0.06 sec)

D2_92814_Krushna>select * from tempp;
+-------+-----------+
| fir   | sec       |
+-------+-----------+
| 40.56 | Area      |
|    26 | Perimeter |
+-------+-----------+
2 rows in set (0.00 sec)
```

2. Write a program that declares an integer variable called num, assigns a value to it, and computes and inserts into the tempp table the value of the variable itself, its square, and its cube.



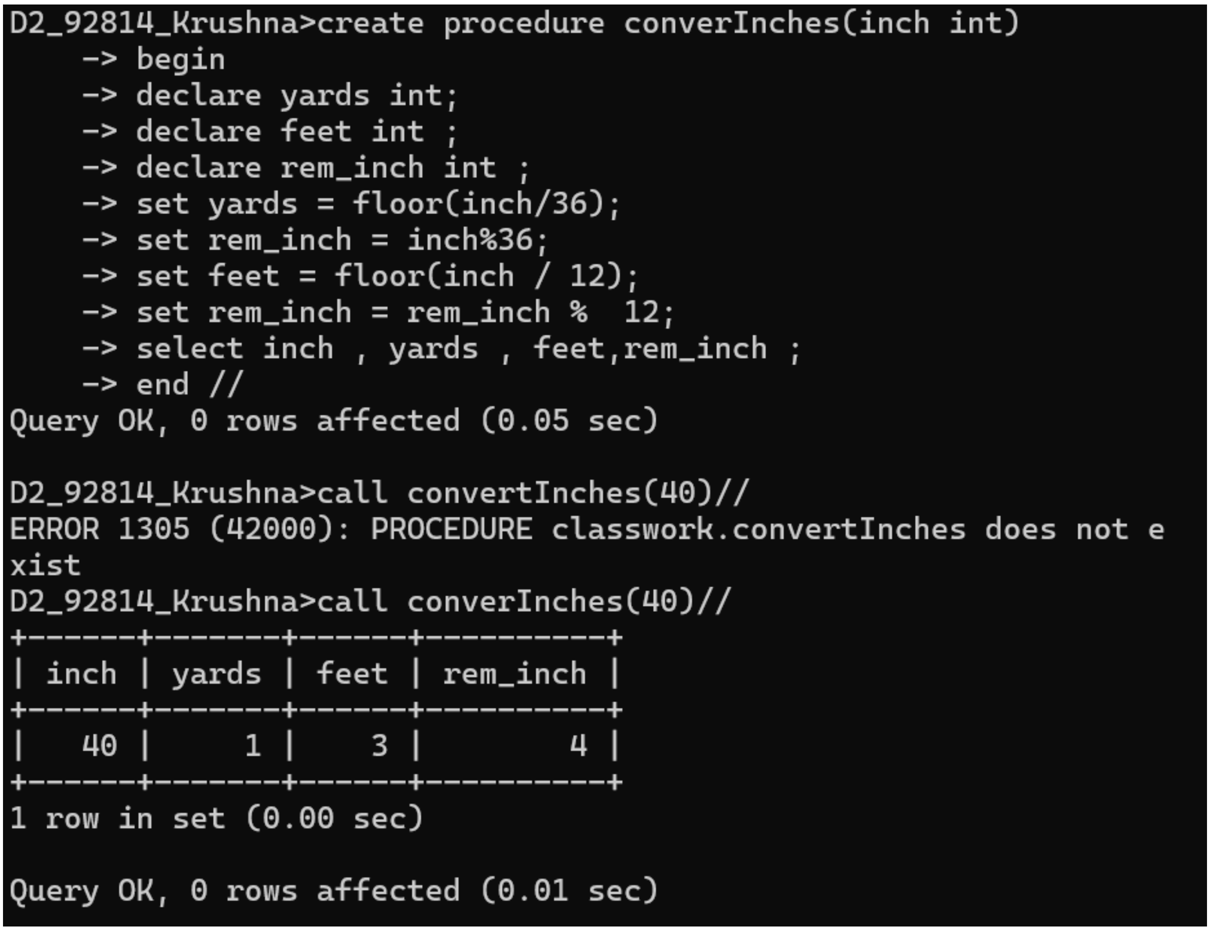
```
D2_92814_Krushna>create procedure CalNum(num int)
    -> begin
    -> declare sq int ;
    -> declare cu int ;
    -> set sq = num * num ;
    -> set cu = num * num * num ;
    -> insert into tempp values(num,'Original No');
    -> insert into tempp values(sq,'Square');
    -> insert into tempp values(cu,'Cube');
    -> end //
Query OK, 0 rows affected (0.02 sec)

D2_92814_Krushna>delimiter ;
D2_92814_Krushna>call CalNum(4);
Query OK, 1 row affected (0.01 sec)

D2_92814_Krushna>select * from tempp;
+------+-------------+
| fir  | sec         |
+------+-------------+
|    4 | Original No |
|   16 | Square      |
|   64 | Cube        |
+------+-------------+
3 rows in set (0.00 sec)
```

3. Convert a temperature in Fahrenheit (F) to its equivalent in Celsius (C) and vice versa. The required formulae are:- C= (F-32)*5/9 F= 9/5*C + 32



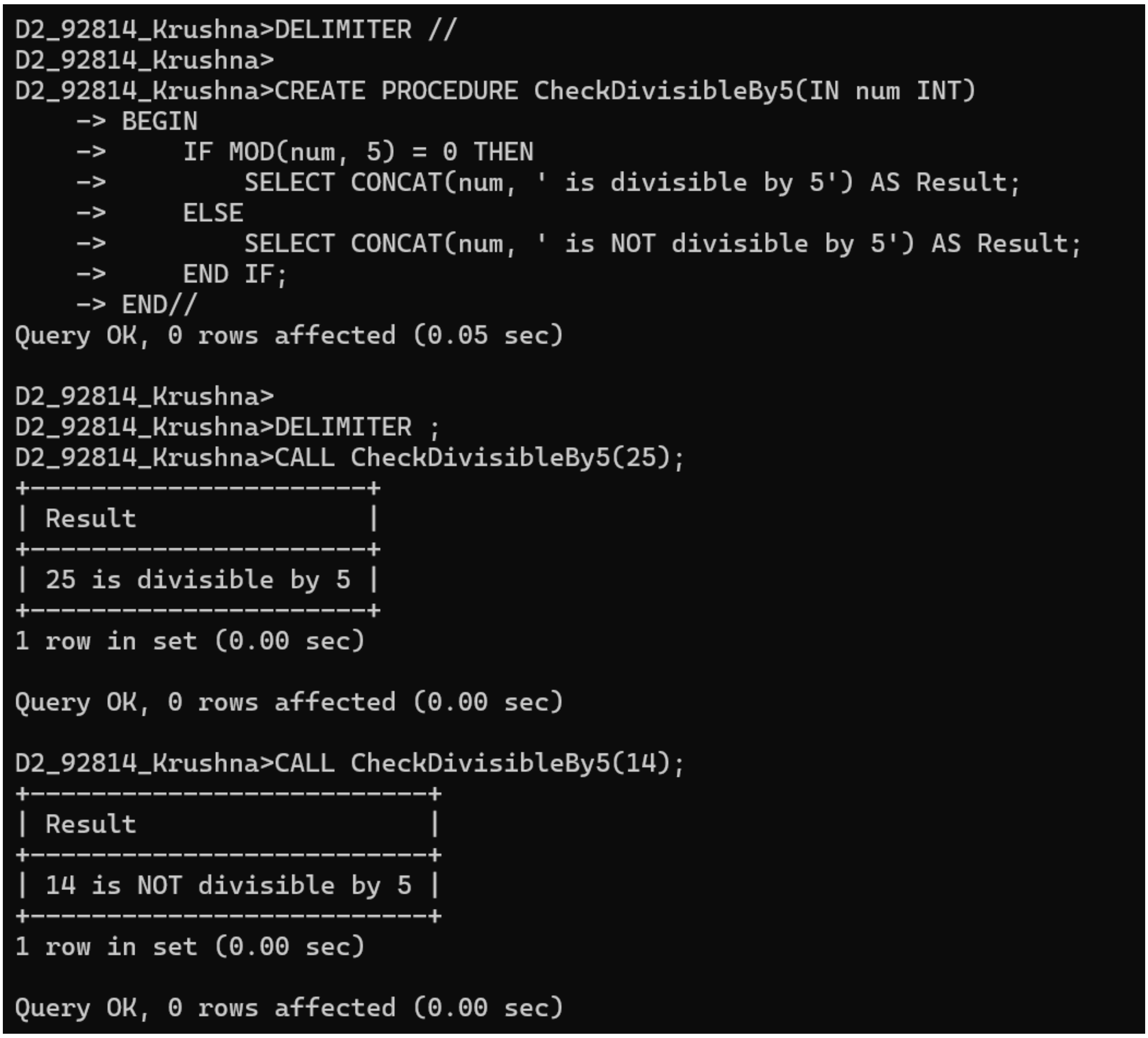
```
D2_92814_Krushna>create procedure tempConversion(t float)
    -> begin
    -> declare c float;
    -> declare f float;
    -> set c = ( t-32) * 5/9 ;
    -> set f = 9/5 * c + 32 ;
    -> insert into tempp values(c,'celcius');
    -> insert into tempp values(f,'fer');
    -> end ; //
Query OK, 0 rows affected (0.04 sec)

D2_92814_Krushna>call tempConversion(34.5);
    -> //
Query OK, 1 row affected (0.01 sec)

D2_92814_Krushna>select * from tempp//
+------+---------+
| fir  | sec     |
+------+---------+
|    1 | celcius |
|   34 | fer     |
+------+---------+
2 rows in set (0.00 sec)
```

4. Convert a number of inches into yards, feet, and inches. For example, 124 inches equals 3 yards, 1 foot, and 4 inches.



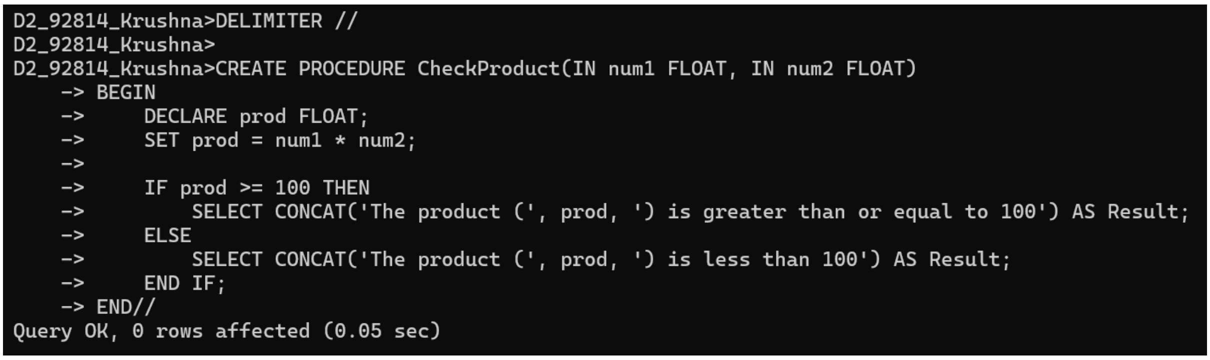
```
D2_92814_Krushna>create procedure converInches(inch int)
    -> begin
    -> declare yards int;
    -> declare feet int ;
    -> declare rem_inch int ;
    -> set yards = floor(inch/36);
    -> set rem_inch = inch%36;
    -> set feet = floor(inch / 12);
    -> set rem_inch = rem_inch %  12;
    -> select inch , yards , feet,rem_inch ;
    -> end //
Query OK, 0 rows affected (0.05 sec)

D2_92814_Krushna>call convertInches(40)//
ERROR 1305 (42000): PROCEDURE classwork.convertInches does not e
xist
D2_92814_Krushna>call converInches(40)//
+------+-------+------+----------+
| inch | yards | feet | rem_inch |
+------+-------+------+----------+
|   40 |     1 |    3 |        4 |
+------+-------+------+----------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

5. Write a program that enables a user to input an integer. The program should then state whether the integer is evenly divisible by 5.

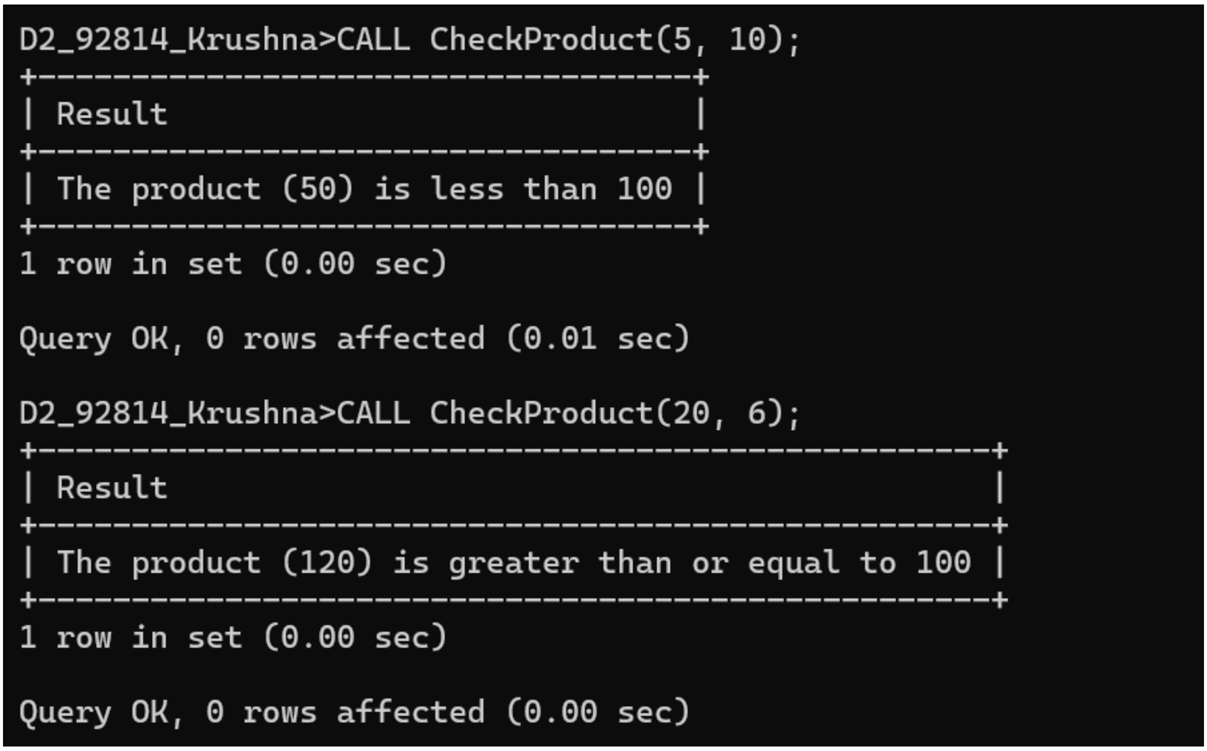


```
D2_92814_Krushna>DELIMITER //
D2_92814_Krushna>
D2_92814_Krushna>CREATE PROCEDURE CheckDivisibleBy5(IN num INT)
    -> BEGIN
    ->     IF MOD(num, 5) = 0 THEN
    ->         SELECT CONCAT(num, ' is divisible by 5') AS Result;
    ->     ELSE
    ->         SELECT CONCAT(num, ' is NOT divisible by 5') AS Result;
    ->     END IF;
    -> END//
Query OK, 0 rows affected (0.05 sec)

D2_92814_Krushna>
D2_92814_Krushna>DELIMITER ;
D2_92814_Krushna>CALL CheckDivisibleBy5(25);
+-----------------------+
| Result                |
+-----------------------+
| 25 is divisible by 5 |
+-----------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

D2_92814_Krushna>CALL CheckDivisibleBy5(14);
+---------------------------+
| Result                    |
+---------------------------+
| 14 is NOT divisible by 5 |
+---------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

6. Your block should read in two real numbers and tell whether the product of the two numbers is equal to or greater than 100.



```
D2_92814_Krushna>DELIMITER //
D2_92814_Krushna>
D2_92814_Krushna>CREATE PROCEDURE CheckProduct(IN num1 FLOAT, IN num2 FLOAT)
    -> BEGIN
    ->     DECLARE prod FLOAT;
    ->     SET prod = num1 * num2;
    ->
    ->     IF prod >= 100 THEN
    ->         SELECT CONCAT('The product (', prod, ') is greater than or equal to 100') AS Result;
    ->     ELSE
    ->         SELECT CONCAT('The product (', prod, ') is less than 100') AS Result;
    ->     END IF;
    -> END//
Query OK, 0 rows affected (0.05 sec)
```



```
D2_92814_Krushna>CALL CheckProduct(5, 10);
+-----------------------------------+
| Result                            |
+-----------------------------------+
| The product (50) is less than 100 |
+-----------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

D2_92814_Krushna>CALL CheckProduct(20, 6);
+---------------------------------------------------+
| Result                                            |
+---------------------------------------------------+
| The product (120) is greater than or equal to 100 |
+---------------------------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```