MySQL Exercise 5

1. Write a stored function to take three parameters, the sides of a triangle. The sides of the triangle should be accepted from the user. The function should return a Boolean value:- true if the triangle is valid, false otherwise. A triangle is valid if the length of each side is less than the sum of the lengths of the other two sides. Check if the dimensions entered can form a valid triangle.

```
D2_92814_Krushna>CREATE FUNCTION is_valid_triangle(a FLOAT, b FLOAT, c FLOAT)
    -> RETURNS BOOLEAN
    -> DETERMINISTIC
    -> BEGIN
    ->     IF (a + b > c) AND (b + c > a) AND (c + a > b) THEN
    ->         RETURN TRUE;
    ->     ELSE
    ->         RETURN FALSE;
    ->     END IF;
    -> END //
Query OK, 0 rows affected (0.04 sec)

D2_92814_Krushna>
D2_92814_Krushna>DELIMITER ;
D2_92814_Krushna>SELECT is_valid_triangle(3, 4, 5) AS result;   -- returns 1 (true)
+--------+
| result |
+--------+
|      1 |
+--------+
1 row in set (0.00 sec)

D2_92814_Krushna>SELECT is_valid_triangle(1, 2, 5) AS result;   -- returns 0 (false)
+--------+
| result |
+--------+
|      0 |
+--------+
1 row in set (0.00 sec)
```

2. Write a function that generates a random number between 1 and 10. Use any logic of your choice to achieve this.

```
D2_92814_Krushna>DELIMITER //
D2_92814_Krushna>
D2_92814_Krushna>CREATE FUNCTION random_1_to_10()
    -> RETURNS INT
    -> DETERMINISTIC
    -> BEGIN
    ->     RETURN FLOOR(1 + (RAND() * 10));
    -> END //
Query OK, 0 rows affected (0.05 sec)

D2_92814_Krushna>
D2_92814_Krushna>DELIMITER ;
D2_92814_Krushna>SELECT random_1_to_10() AS random_number;
+---------------+
| random_number |
+---------------+
|            10 |
+---------------+
1 row in set (0.00 sec)
```

3. Create a function that accepts a string of n characters and exchanges the first character with the last, the second with the next – to – last, and so forth until n exchanges have been made. What will the final string look like? Write the function to verify your conclusion.

```
D2_92814_Krushna>CREATE FUNCTION swap_string(s VARCHAR(255)
    -> RETURNS VARCHAR(255)
    -> DETERMINISTIC
    -> BEGIN
    ->     DECLARE i INT DEFAULT 1;
    ->     DECLARE j INT;
    ->     DECLARE temp CHAR(1);
    ->     DECLARE len INT;
    ->
    ->     SET len = CHAR_LENGTH(s);
    ->     SET j = len;
    ->
    ->     WHILE i < j DO
    ->         SET temp = SUBSTRING(s, i, 1);
    ->         SET s = INSERT(s, i, 1, SUBSTRING(s, j, 1));
    ->         SET s = INSERT(s, j, 1, temp);
    ->         SET i = i + 1;
    ->         SET j = j - 1;
    ->     END WHILE;
    ->
    ->     RETURN s;
    -> END //
Query OK, 0 rows affected (0.05 sec)

D2_92814_Krushna>
D2_92814_Krushna>DELIMITER ;
D2_92814_Krushna>SELECT swap_string('HELLO') AS swapped;
+---------+
| swapped |
+---------+
| OLLEH   |
+---------+
1 row in set (0.00 sec)

D2_92814_Krushna>SELECT swap_string('ABCDEFG') AS swapped;
+---------+
| swapped |
+---------+
| GFEDCBA |
```