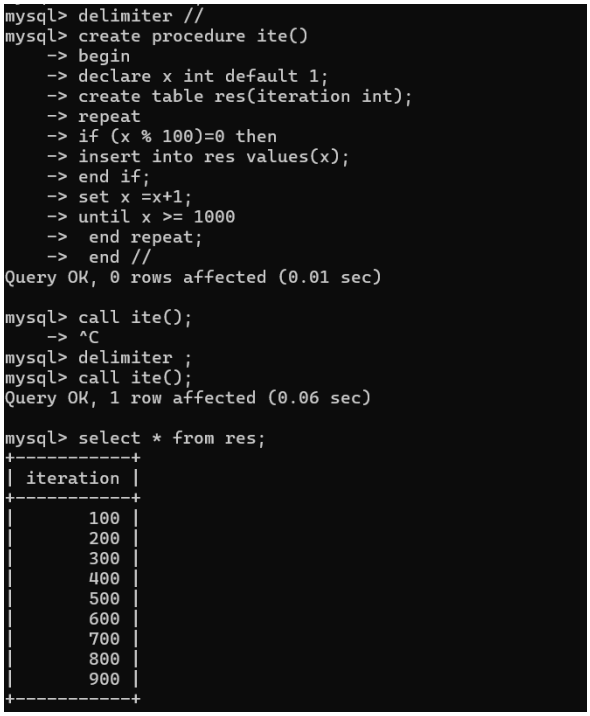
Exercise PL 2

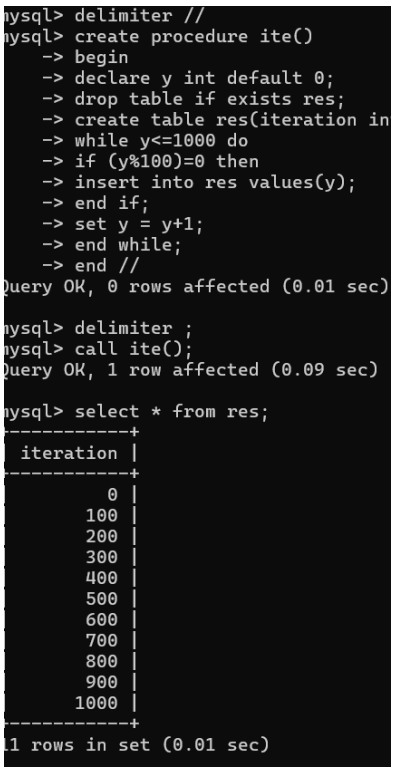
1. Write a program containing a loop that iterates from 1 to 1000 using a variable I, which is incremented each time around the loop. The program should output the value of I every hundred iterations (i.e., the output should be 100, 200, etc.).



```
mysql> delimiter //
mysql> create procedure ite()
    -> begin
    -> declare x int default 1;
    -> create table res(iteration int);
    -> repeat
    -> if (x % 100)=0 then
    -> insert into res values(x);
    -> end if;
    -> set x =x+1;
    -> until x >= 1000
    ->  end repeat;
    ->  end //
Query OK, 0 rows affected (0.01 sec)

mysql> call ite();
    -> ^C
mysql> delimiter ;
mysql> call ite();
Query OK, 1 row affected (0.06 sec)

mysql> select * from res;
+-----------+
| iteration |
+-----------+
|       100 |
|       200 |
|       300 |
|       400 |
|       500 |
|       600 |
|       700 |
|       800 |
|       900 |
+-----------+
```



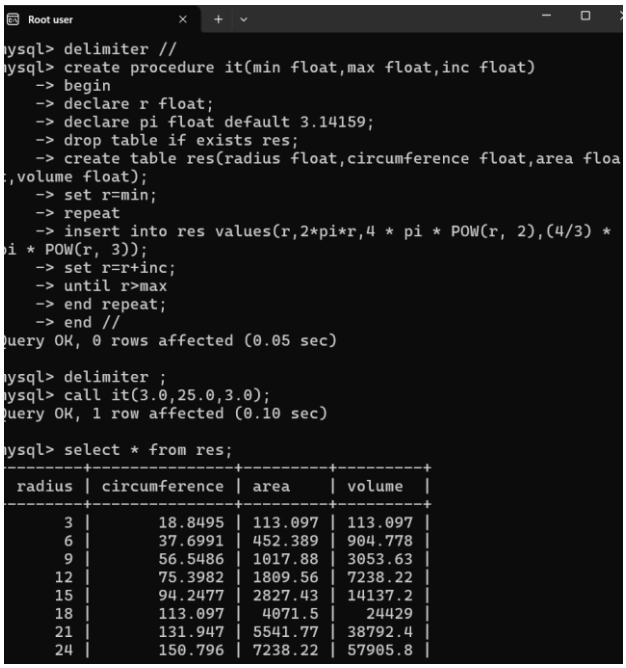
```
ysql> delimiter //
ysql> create procedure ite()
    -> begin
    -> declare y int default 0;
    -> drop table if exists res;
    -> create table res(iteration in
    -> while y<=1000 do
    -> if (y%100)=0 then
    -> insert into res values(y);
    -> end if;
    -> set y = y+1;
    -> end while;
    -> end //
Query OK, 0 rows affected (0.01 sec)

ysql> delimiter ;
ysql> call ite();
Query OK, 1 row affected (0.09 sec)

ysql> select * from res;
+-----------+
| iteration |
+-----------+
|         0 |
|       100 |
|       200 |
|       300 |
|       400 |
|       500 |
|       600 |
|       700 |
|       800 |
|       900 |
|      1000 |
+-----------+
11 rows in set (0.01 sec)
```

2. Write a program that examines all the numbers from 1 to 999, displaying all those for which the sum of the cubes of the digits equal the number itself.

3. Write a program that Selects from any table a minimum and maximum value for a radius, along with an increment factor, and generates a series of radii by repeatedly adding the increment to the minimum until the maximum is reached. For each value of the radius, compute and display the circumference, area, and volume of the sphere. (Be sure to include both the maximum and the minimum values.).



```
Root user
ysql> delimiter //
ysql> create procedure it(min float,max float,inc float)
    -> begin
    -> declare r float;
    -> declare pi float default 3.14159;
    -> drop table if exists res;
    -> create table res(radius float,circumference float,area floa
,volume float);
    -> set r=min;
    -> repeat
    -> insert into res values(r,2*pi*r,4 * pi * POW(r, 2),(4/3) *
i * POW(r, 3));
    -> set r=r+inc;
    -> until r>max
    -> end repeat;
    -> end //
uery OK, 0 rows affected (0.05 sec)

ysql> delimiter ;
ysql> call it(3.0,25.0,3.0);
uery OK, 1 row affected (0.10 sec)

ysql> select * from res;
+--------+---------------+---------+---------+
| radius | circumference | area    | volume  |
+--------+---------------+---------+---------+
|      3 |       18.8495 | 113.097 | 113.097 |
|      6 |       37.6991 | 452.389 | 904.778 |
|      9 |       56.5486 | 1017.88 | 3053.63 |
|     12 |       75.3982 | 1809.56 | 7238.22 |
|     15 |       94.2477 | 2827.43 | 14137.2 |
|     18 |       113.097 |  4071.5 |   24429 |
|     21 |       131.947 | 5541.77 | 38792.4 |
|     24 |       150.796 | 7238.22 | 57905.8 |
```

4. A palindrome is a word that is spelled the same forward and backward, such as level, radar, etc. Write a program to Selects from any table a five letter word and determine whether it is a palindrome.

```
mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE check_palindromes()
    -> BEGIN
    ->     DECLARE done INT DEFAULT 0;
    ->     DECLARE w VARCHAR(10);
    ->     DECLARE reversed VARCHAR(10);
    ->     DECLARE cur CURSOR FOR SELECT word FROM words WHERE CHAR_LENGTH(word) = 5;
    ->     DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
    ->
    ->     CREATE TEMPORARY TABLE IF NOT EXISTS palindromes (
    ->         original VARCHAR(10),
    ->         status VARCHAR(20)
    ->     );
    ->
    ->     OPEN cur;
    ->
    ->     read_loop: LOOP
    ->         FETCH cur INTO w;
    ->         IF done THEN
    ->             LEAVE read_loop;
    ->         END IF;
    ->
    ->         SET reversed = REVERSE(w);
    ->
    ->         IF w = reversed THEN
    ->             INSERT INTO palindromes VALUES (w, 'Palindrome');
    ->         ELSE
    ->             INSERT INTO palindromes VALUES (w, 'Not Palindrome');
    ->         END IF;
    ->     END LOOP;
    ->
    ->     CLOSE cur;
    ->
    ->     SELECT * FROM palindromes;
    -> END //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL check_palindromes();
ERROR 1146 (42S02): Table 'assignment1.words' doesn't exist
mysql> CREATE TABLE words (
    ->     word VARCHAR(10)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> INSERT INTO words VALUES ('level'), ('radar'), ('hello'), ('world'), ('refer');
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql>  CALL check_palindromes();
+----------+----------------+
| original | status         |
+----------+----------------+
| level    | Palindrome     |
| radar    | Palindrome     |
| hello    | Not Palindrome |
| world    | Not Palindrome |
| refer    | Palindrome     |
+----------+----------------+
```

5. Modify the above program to Select from any table a variable length word. This requires determining how many characters are read in.

```
mysql> CREATE PROCEDURE check_variable_palindromes()
    -> BEGIN
    ->     DECLARE done INT DEFAULT 0;
    ->     DECLARE w VARCHAR(100);
    ->     DECLARE reversed VARCHAR(100);
    ->
    ->     DECLARE cur CURSOR FOR SELECT word FROM words;
    ->     DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
    ->
    ->     DROP TEMPORARY TABLE IF EXISTS palindromes;
    ->
    ->     CREATE TEMPORARY TABLE palindromes (
    ->         original VARCHAR(100),
    ->         length_of_word INT,
    ->         status VARCHAR(20)
    ->     );
    ->
    ->     OPEN cur;
    ->
    ->     read_loop: LOOP
    ->         FETCH cur INTO w;
    ->         IF done THEN
    ->             LEAVE read_loop;
    ->         END IF;
    ->
    ->         SET reversed = REVERSE(w);
    ->
    ->         IF w = reversed THEN
    ->             INSERT INTO palindromes VALUES (w, CHAR_LENGTH(w), 'Palindrome');
    ->         ELSE
    ->             INSERT INTO palindromes VALUES (w, CHAR_LENGTH(w), 'Not Palindrome');
    ->         END IF;
    ->     END LOOP;
    ->
    ->     CLOSE cur;
    ->
    ->     SELECT * FROM palindromes;
    -> END //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql>  call check_variable_palindromes();
+----------+----------------+----------------+
| original | length_of_word | status         |
+----------+----------------+----------------+
| level    |              5 | Palindrome     |
| radar    |              5 | Palindrome     |
| hello    |              5 | Not Palindrome |
| world    |              5 | Not Palindrome |
| refer    |              5 | Palindrome     |
| madam    |              5 | Palindrome     |
| cat      |              3 | Not Palindrome |
| deed     |              4 | Palindrome     |
| civic    |              5 | Palindrome     |
```