Exercise 3

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.).

Ans:

```
DELIMITER //
CREATE PROCEDURE loop_1000()
BEGIN
   DECLARE i INT DEFAULT 1;
   WHILE i <= 1000 DO
      IF i MOD 100 = 0 THEN
         SELECT i AS output;
      END IF;
      SET i = i + 1;
   END WHILE;
END//
DELIMITER ;
```

```
+--------+
| output |
+--------+
|    100 |
+--------+
```

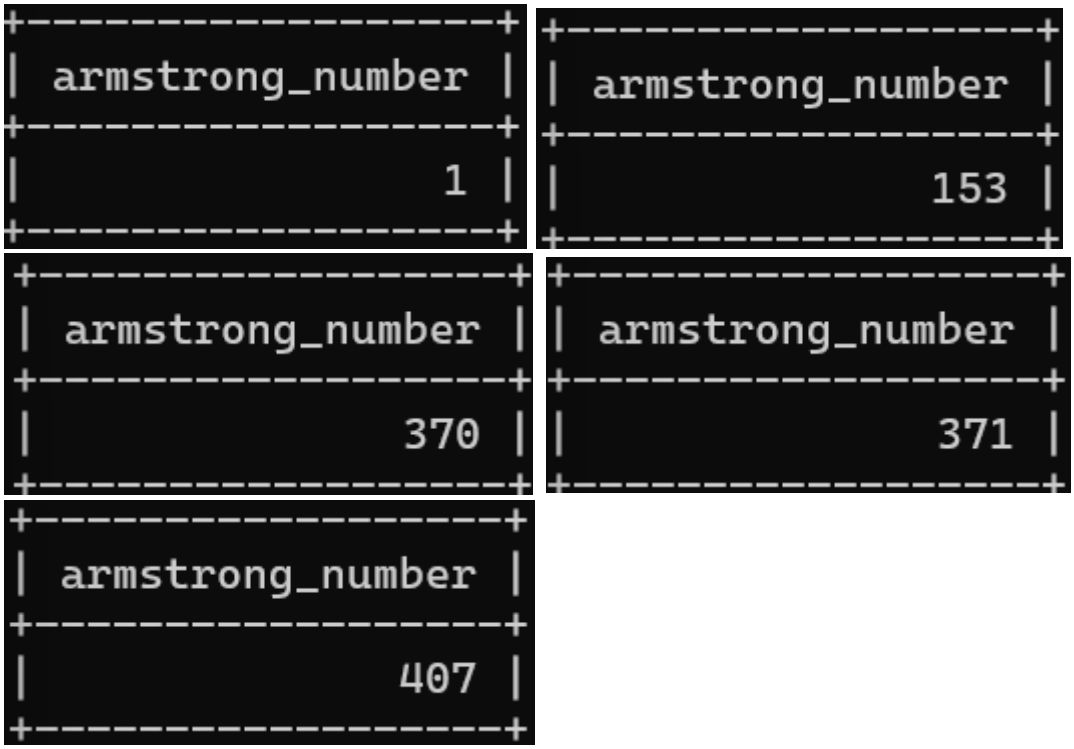
```
+--------+
| output |
+--------+
|   1000 |
+--------+
```

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.

Ans:

```
DELIMITER //
```

```
CREATE PROCEDURE find_armstrong()
BEGIN
   DECLARE i INT DEFAULT 1;
   DECLARE a INT;
   DECLARE b INT;
   DECLARE c INT;
   DECLARE sum INT;
   WHILE i <= 999 DO
      SET a = FLOOR(i / 100);
      SET b = FLOOR((i % 100) / 10);
      SET c = i % 10;
      SET sum = POW(a, 3) + POW(b, 3) + POW(c, 3);
      IF sum = i THEN
         SELECT i AS armstrong_number;
      END IF;
      SET i = i + 1;
   END WHILE;
END//
DELIMITER ;
```



armstrong_number
1

armstrong_number
153

armstrong_number
370

armstrong_number
371

armstrong_number
407

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.).

Ans:

```
DELIMITER //

CREATE PROCEDURE sphere_calc()

BEGIN

    DECLARE r FLOAT;

    DECLARE min_r FLOAT;

    DECLARE max_r FLOAT;

    DECLARE inc FLOAT;

    SELECT min_radius, max_radius, increment INTO min_r, max_r, inc FROM radius_table
LIMIT 1;

    SET r = min_r;

    WHILE r <= max_r DO

        SELECT
```

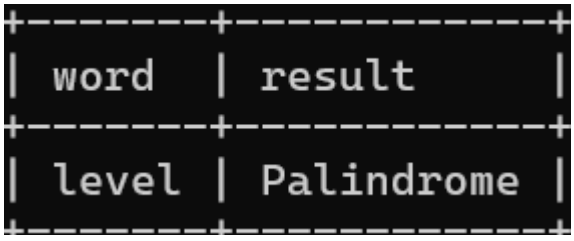
```
        r AS radius,
        2 * PI() * r AS circumference,
        PI() * POW(r, 2) AS area,
        (4/3) * PI() * POW(r, 3) AS volume;
    SET r = r + inc;
  END WHILE;
END//
DELIMITER ;
```

```
+--------+--------------------+--------------------+--------------------+
| radius | circumference      | area               | volume             |
+--------+--------------------+--------------------+--------------------+
|      1 | 6.283185307179586 | 3.141592653589793 | 4.188790203739193 |
+--------+--------------------+--------------------+--------------------+
```

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.

Ans:

```
DELIMITER //
CREATE PROCEDURE check_palindrome()
BEGIN
  DECLARE w VARCHAR(10);
  SELECT word INTO w FROM word_table LIMIT 1;
  IF w = REVERSE(w) THEN
    SELECT w AS word, 'Palindrome' AS result;
  ELSE
    SELECT w AS word, 'Not a Palindrome' AS result;
  END IF;
END//
DELIMITER ;
```

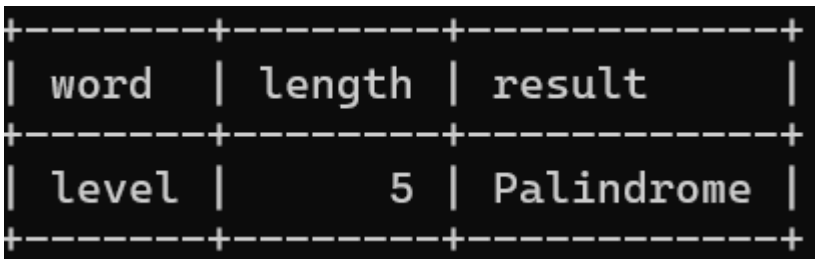


word	result
level	Palindrome

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

Ans: DELIMITER //

```
CREATE PROCEDURE check_var_length_palindrome()
BEGIN
   DECLARE w VARCHAR(50);
   SELECT word INTO w FROM word_table LIMIT 1;
   IF w = REVERSE(w) THEN
      SELECT w AS word, LENGTH(w) AS length, 'Palindrome' AS result;
   ELSE
      SELECT w AS word, LENGTH(w) AS length, 'Not a Palindrome' AS result;
   END IF;
END//
DELIMITER ;
```



word	length	result
level	5	Palindrome