PL/SQL, Functions,Procedures

1. Write a PL/SQL code to check given number is Armstrong or not.

```
DECLARE
  num NUMBER := &num;
  temp NUMBER;
  sum NUMBER := 0;
  r NUMBER;
BEGIN
  temp := num;
  WHILE temp > 0 LOOP
    r := MOD(temp, 10);
    sum := sum + r*r*r;
    temp := temp / 10;
  END LOOP;
  IF sum = num THEN
    DBMS_OUTPUT.PUT_LINE(num || ' is an Armstrong number');
  ELSE
    DBMS_OUTPUT.PUT_LINE(num || ' is not an Armstrong number');
  END IF;
END;
```

2. Write a PL/SQL code to implement calculator program

```
DECLARE
  a NUMBER := &a;
  b NUMBER := &b;
  result NUMBER;
BEGIN
  result := a + b;
  DBMS_OUTPUT.PUT_LINE('Addition: ' || result);
  result := a - b;
  DBMS_OUTPUT.PUT_LINE('Subtraction: ' || result);
  result := a * b;
  DBMS_OUTPUT.PUT_LINE('Multiplication: ' || result);
  result := a / b;
  DBMS_OUTPUT.PUT_LINE('Division: ' || result);
END;
```

3. Write query PL/SQL procedure to find factorial of a number

```
CREATE OR REPLACE PROCEDURE factorial(num IN NUMBER, fact OUT NUMBER)
AS
BEGIN
  fact := 1;
  FOR i IN 1..num LOOP
    fact := fact * i;
  END LOOP;
END;
```

4. Write a function to find cube of a number passed as an argument

```
CREATE OR REPLACE FUNCTION cube_number(num IN NUMBER) RETURN
NUMBER IS
  result NUMBER;
BEGIN
  result := num * num * num;
  RETURN result;
END;
```

5. Write a function to find perfect number

```
CREATE OR REPLACE FUNCTION is_perfect(num IN NUMBER) RETURN
VARCHAR2 IS
  sum NUMBER := 0;
BEGIN
  FOR i IN 1..num-1 LOOP
    IF MOD(num, i) = 0 THEN
      sum := sum + i;
    END IF;
  END LOOP;
  IF sum = num THEN
    RETURN 'Perfect';
  ELSE
    RETURN 'Not Perfect';
  END IF;
END;
```

6. Using procedures find he sum of digits of a number

```
CREATE OR REPLACE PROCEDURE sum_of_digits(num IN NUMBER, sum OUT
NUMBER) AS
  digit NUMBER;
BEGIN
  sum := 0;
  WHILE num > 0 LOOP
    digit := MOD(num, 10);
    sum := sum + digit;
    num := num / 10;
  END LOOP;
END;
```

7. Using functions find gcd of 2 numbers

```
CREATE OR REPLACE FUNCTION gcd(a IN NUMBER, b IN NUMBER) RETURN
NUMBER IS
BEGIN
  IF b = 0 THEN
    RETURN a;
  ELSE
    RETURN gcd(b, MOD(a, b));
  END IF;
```

```
END;
```

8. Write a PL/SQL code to check given number is even or not.

```
CREATE OR REPLACE PROCEDURE check_even(num IN NUMBER) AS
BEGIN
  IF MOD(num, 2) = 0 THEN
    DBMS_OUTPUT.PUT_LINE(num || ' is even');
  ELSE
    DBMS_OUTPUT.PUT_LINE(num || ' is odd');
  END IF;
END;
```

9. Write a procedure to find reverse of a number

```
CREATE OR REPLACE PROCEDURE reverse_number(num IN NUMBER, reversed OUT
NUMBER) AS
  temp NUMBER := num;
  remainder NUMBER;
BEGIN
  reversed := 0;
  WHILE temp > 0 LOOP
    remainder := MOD(temp, 10);
    reversed := (reversed * 10) + remainder;
    temp := temp / 10;
  END LOOP;
END;
```

10. Write a function to check a number is prime or not

```
CREATE OR REPLACE FUNCTION is_prime(num IN NUMBER) RETURN
VARCHAR2 IS
  i NUMBER;
BEGIN
  IF num <= 1 THEN
    RETURN 'Not Prime';
  END IF;
  FOR i IN 2..SQRT(num) LOOP
    IF MOD(num, i) = 0 THEN
      RETURN 'Not Prime';
    END IF;
  END LOOP;
  RETURN 'Prime';
END;
```