DBMS Lab

Name: **Soham Das**

Section: **A1**

Roll No: **002311001004**

Assignment – 4A

IT-UG2

1. Write a PL/SQL code to print Today is fall on weekend or weekdays using if else statement.

```
Declare
todays_date DATE;
current_day varchar(9);
Begin
todays_date :=sysdate;
current_day:=to_char(todays_date,'day');
current_day:=initcap(current_day);
current_day:=rtrim(current_day);
if current_day='Sunday' or current_day='Saturday' then
dbms_output.put_line('Today is fall on weekend');
Else
dbms_output.put_line('Today is fall on weekdays');
end if;
end;
```

```
Statement processed.
Today is fall on weekend
```

2. Write a PL/SQL code to check that an inputted a single character is vowel or not .If vowel then display which vowel it is.

```
DECLARE
  input_char CHAR(1);
  is_vowel BOOLEAN;

BEGIN
```

```
  input_char := 'S';
  is_vowel := UPPER(input_char) IN ('A', 'E', 'I', 'O', 'U');

  IF is_vowel THEN
    DBMS_OUTPUT.PUT_LINE(input_char || ' is a vowel.');
  ELSE
    DBMS_OUTPUT.PUT_LINE(input_char || ' is not a vowel.');
  END IF;
END;
```

```
Statement processed.
S is not a vowel.
```

3. Write a PL/SQL code block to find out the sum of first twenty natural numbers (1+2+3+4+5+6+7+8+9+10+-----+20 this series).

```
DECLARE
  v_sum NUMBER := 0;
BEGIN
  FOR i IN 1..20 LOOP
    v_sum := v_sum + i;
  END LOOP;

  DBMS_OUTPUT.PUT_LINE('Sum : ' || v_sum);
END;
```

```
Statement processed.
Sum : 210
```

4. Write a PL/SQL block that will ask for two numbers and one operand (+, -, *, /). Then it will calculate and display the result.

```
DECLARE
  v_num1 NUMBER := 10;  -- Change this value to & for SQL+
  v_num2 NUMBER := 5;   -- Change this value to & for SQL+
  v_operator CHAR(1) := '+';
  v_result NUMBER;
BEGIN
  IF v_operator = '+' THEN
    v_result := v_num1 + v_num2;
  ELSIF v_operator = '-' THEN
    v_result := v_num1 - v_num2;
  ELSIF v_operator = '*' THEN
    v_result := v_num1 * v_num2;
  ELSIF v_operator = '/' THEN
    IF v_num2 != 0 THEN
      v_result := v_num1 / v_num2;
    ELSE
      DBMS_OUTPUT.PUT_LINE('Error: Division by zero.');
      RETURN;
    END IF;
  ELSE
    DBMS_OUTPUT.PUT_LINE('Error: Invalid operator.');
    RETURN;
  END IF;

  DBMS_OUTPUT.PUT_LINE('The result of ' || v_num1 || ' ' || v_operator || ' ' || v_num2 || '
is: ' || v_result);
END;
```

```
Statement processed.
The result of 10 + 5 is: 15
```

5. Write a PL/SQL code block to display a number in a reverse way.

```
DECLARE
  original_number NUMBER := 12345;
  reversed_number NUMBER := 0;
BEGIN
  WHILE original_number > 0 LOOP
    reversed_number := reversed_number * 10 + MOD(original_number, 10);
    original_number := FLOOR(original_number / 10);
  END LOOP;

  DBMS_OUTPUT.PUT_LINE('Reversed Number: ' || reversed_number);
END;
```

```
Statement processed.
Reversed Number: 54321
```

6. Write a PL/SQL block to display the dates of this month which are Tuesday.

```
DECLARE
        v_date DATE := TRUNC(SYSDATE, 'MM');  -- Start from the first day of the current
month
BEGIN
        DBMS_OUTPUT.PUT_LINE('Tuesdays of this month:');
        WHILE v_date < LAST_DAY(SYSDATE) LOOP
        IF TO_CHAR(v_date, 'DY') = 'TUE' THEN
        DBMS_OUTPUT.PUT_LINE(TO_CHAR(v_date, 'DD-MM-YYYY'));
```

```
        END IF;
        v_date := v_date + 1;  -- Move to the next day
        END LOOP;
END;
```

```
Statement processed.
Tuesdays of this month:
05-11-2024
12-11-2024
19-11-2024
26-11-2024
```

7. Write a program in PL/SQL to print the prime numbers between 1 to 50.

```
declare
s NUMBER := 0;
f NUMBER :=0;
begin
for i in 1..50 LOOP
        f :=0;
        for j in 2..i/2 LOOP
        if MOD(i,j)=0 then
        f :=1;
        end if;
        end LOOP;
        if f=0 then
        dbms_output.put_line(i || ' is Prime');
        end if;
```

```
end LOOP;

End;
```

```
Statement processed.
1 is Prime
2 is Prime
3 is Prime
5 is Prime
7 is Prime
11 is Prime
13 is Prime
17 is Prime
19 is Prime
23 is Prime
29 is Prime
31 is Prime
37 is Prime
41 is Prime
43 is Prime
47 is Prime
```

8. Write a program in PL/SQL to print the sum of digits of a number [eg: 635=14].

```
declare
num NUMBER:=325;
res NUMBER:=0;
begin
while num>0 LOOP
        res:=res + MOD(num,10);
        num:=FLOOR(num/10);
end LOOP;


dbms_output.put_line('result is = ' || res);
end;
```

```
Statement processed.
result is = 10
```