KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY (KIIT)

Deemed to be University U/S 3 of the UGC Act, 1956

DBMS LAB ASSIGNMENT 8

Name :- Abhishek Kumar Tiwari

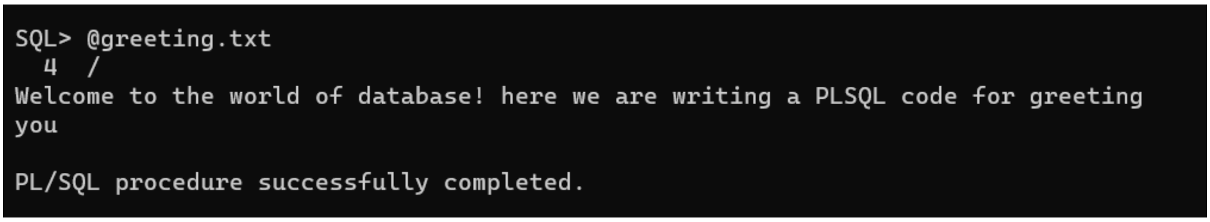
Date:- 26th April 2023

Roll No:- 21051449

Sec:- CSE-19

1.Write a PLSQL program to display a greeting text using.

```
begin
dbms_output.put_line('Welcome to the world of database! here we are writing a PLSQL code for greeting you ');
end;
```

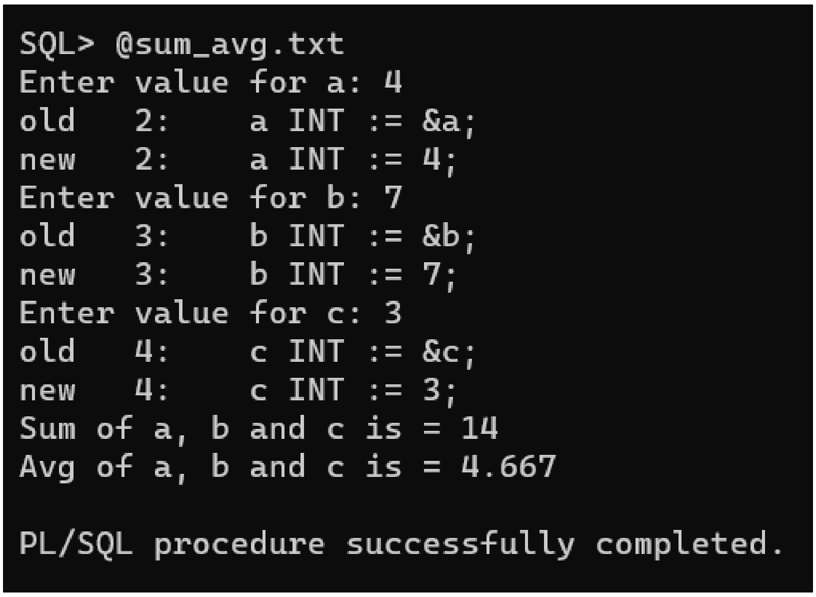


```
SQL> @greeting.txt
  4  /
Welcome to the world of database! here we are writing a PLSQL code for greeting
you

PL/SQL procedure successfully completed.
```

2. Write a PLSQL program to display sum and avg of 3 numbers.

```
DECLARE
  a INT := &a;
  b INT := &b;
  c INT := &c;
  sum1 INT;
  avg1 NUMBER(12,3);
BEGIN
  sum1 := a + b + c;
  avg1 := sum1 / 3;
  dbms_output.put_line('Sum of a, b and c is = ' || sum1);
  dbms_output.put_line('Avg of a, b and c is = ' || avg1);
END;
/
```

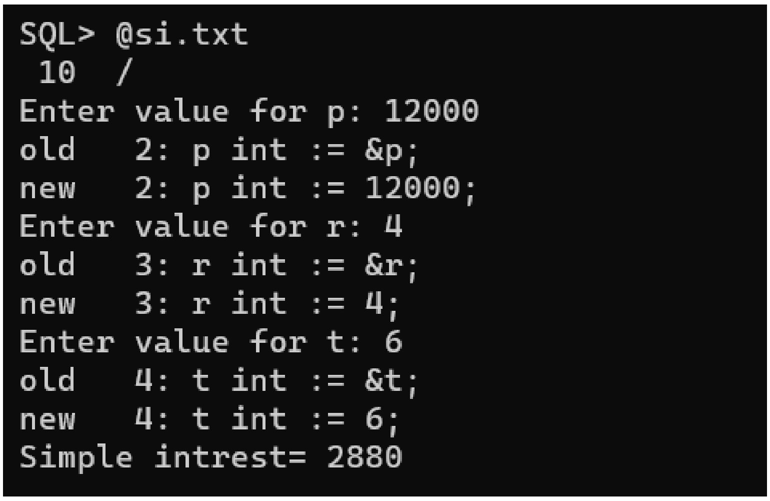


```
SQL> @sum_avg.txt
Enter value for a: 4
old   2:     a INT := &a;
new   2:     a INT := 4;
Enter value for b: 7
old   3:     b INT := &b;
new   3:     b INT := 7;
Enter value for c: 3
old   4:     c INT := &c;
new   4:     c INT := 3;
Sum of a, b and c is = 14
Avg of a, b and c is = 4.667

PL/SQL procedure successfully completed.
```

3. Write a PLSQL program to display simple interest with required input.

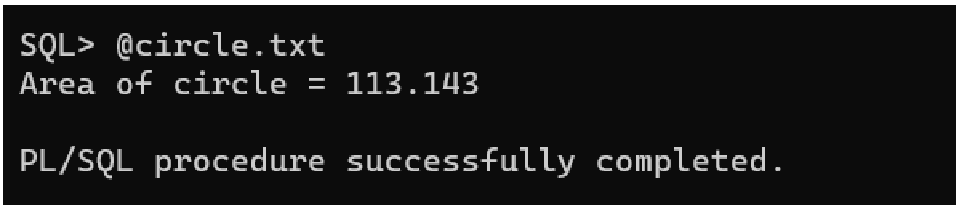
```
declare
p int := &p;
r int := &r;
t int := &t;
si number;
begin
si:= (p*r*t)/100;
dbms_output.put_line('Simple intrest= '|| si);
end;
```



```
SQL> @si.txt
 10  /
Enter value for p: 12000
old   2: p int := &p;
new   2: p int := 12000;
Enter value for r: 4
old   3: r int := &r;
new   3: r int := 4;
Enter value for t: 6
old   4: t int := &t;
new   4: t int := 6;
Simple intrest= 2880
```

4. Write a PLSQL program to display area of circle by accepting radius.

```
declare
r int := 6;
area decimal(12,3);
begin
area := (22/7)*r*r;
dbms_output.put_line('Area of circle = '|| area);
end;
/
```

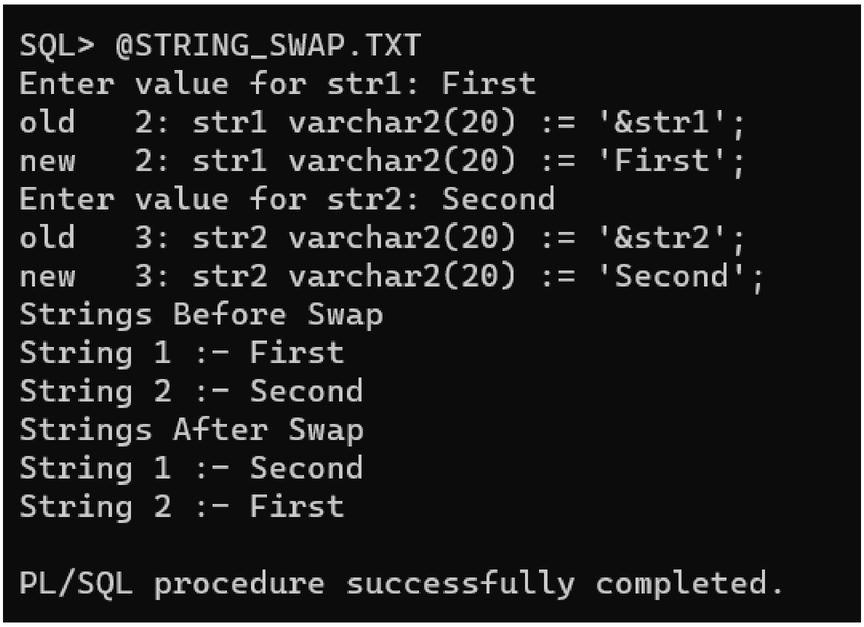


```
SQL> @circle.txt
Area of circle = 113.143

PL/SQL procedure successfully completed.
```

5. Write a PLSQL program to accept two strings and swap.

```
declare
str1 varchar2(20) := '&str1';
str2 varchar2(20) := '&str2';
temp varchar2(20);
begin
dbms_output.put_line('Strings Before Swap');
dbms_output.put_line('String 1 :- '|| str1);
dbms_output.put_line('String 2 :- '|| str2);
temp := str1;
str1 := str2;
str2 := temp;
dbms_output.put_line('Strings After Swap');
dbms_output.put_line('String 1 :- '|| str1);
dbms_output.put_line('String 2 :- '|| str2);
end;
/
```

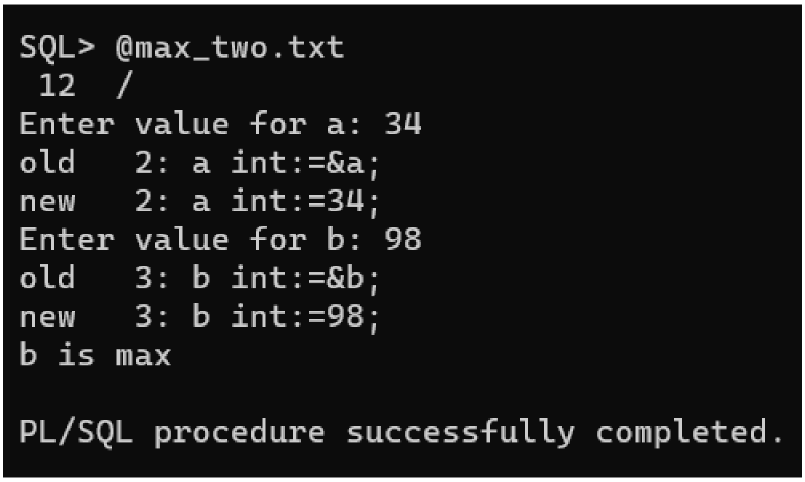


```
SQL> @STRING_SWAP.TXT
Enter value for str1: First
old   2: str1 varchar2(20) := '&str1';
new   2: str1 varchar2(20) := 'First';
Enter value for str2: Second
old   3: str2 varchar2(20) := '&str2';
new   3: str2 varchar2(20) := 'Second';
Strings Before Swap
String 1 :- First
String 2 :- Second
Strings After Swap
String 1 :- Second
String 2 :- First

PL/SQL procedure successfully completed.
```

6. Write a PLSQL program to display greatest among 2 numbers.

```
declare
a int:=&a;
b int:=&b;
begin
if(a>b)
then
dbms_output.put_line('a is max');
else
dbms_output.put_line('b is max');
end if;
end;
```

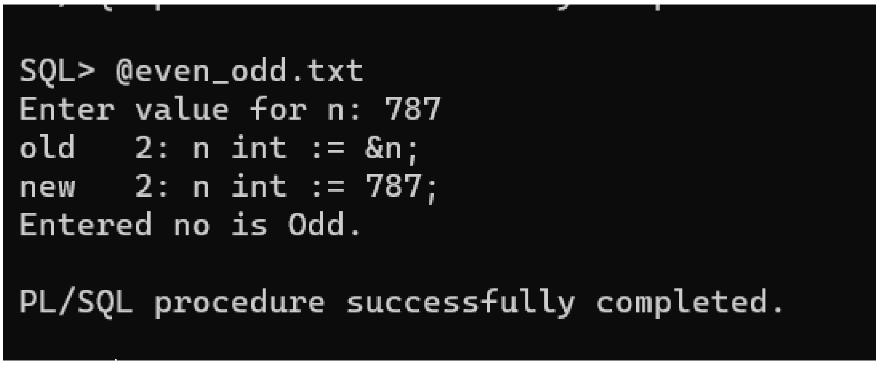


```
SQL> @max_two.txt
 12  /
Enter value for a: 34
old   2: a int:=&a;
new   2: a int:=34;
Enter value for b: 98
old   3: b int:=&b;
new   3: b int:=98;
b is max

PL/SQL procedure successfully completed.
```

7. Write a PLSQL program to check a number is even or odd.

```
declare
n int := &n;
begin
if((n)mod(2)=0)
then
dbms_output.put_line('Entered no is Even.');
else
dbms_output.put_line('Entered no is Odd.');
end if;
end;
/
```

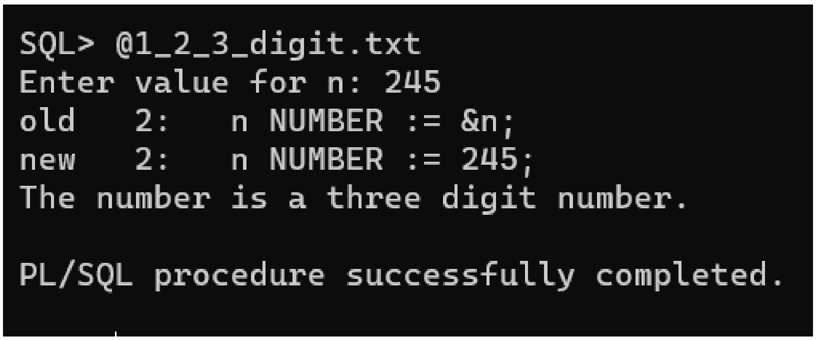


```
SQL> @even_odd.txt
Enter value for n: 787
old   2: n int := &n;
new   2: n int := 787;
Entered no is Odd.

PL/SQL procedure successfully completed.
```

8. Write a PLSQL program to check a number is single/two/three or more than three digit number.

```
DECLARE
  n NUMBER := &n;
BEGIN
  IF n < 10 THEN
    DBMS_OUTPUT.PUT_LINE('The number is a single digit number.');
  ELSIF n < 100 THEN
    DBMS_OUTPUT.PUT_LINE('The number is a two digit number.');
  ELSIF n < 1000 THEN
    DBMS_OUTPUT.PUT_LINE('The number is a three digit number.');
  ELSE
    DBMS_OUTPUT.PUT_LINE('The number is more than three digit number.');
  END IF;
END;
/
```

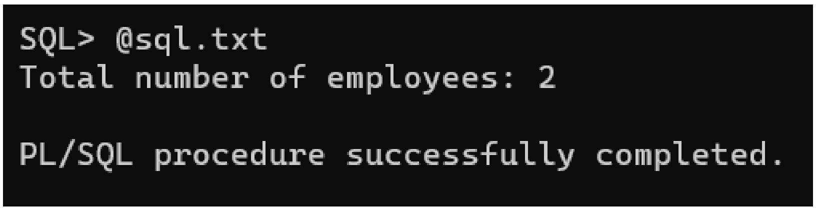


```
SQL> @1_2_3_digit.txt
Enter value for n: 245
old   2:    n NUMBER := &n;
new   2:    n NUMBER := 245;
The number is a three digit number.

PL/SQL procedure successfully completed.
```

9. Execute any sql query/command through PLSQL program.

```
DECLARE
  query_str VARCHAR2(200) := 'SELECT COUNT(*) FROM employees';
  result NUMBER;
BEGIN
  EXECUTE IMMEDIATE query_str INTO result;
  DBMS_OUTPUT.PUT_LINE('Total number of employees: ' || result);
END;
/
```

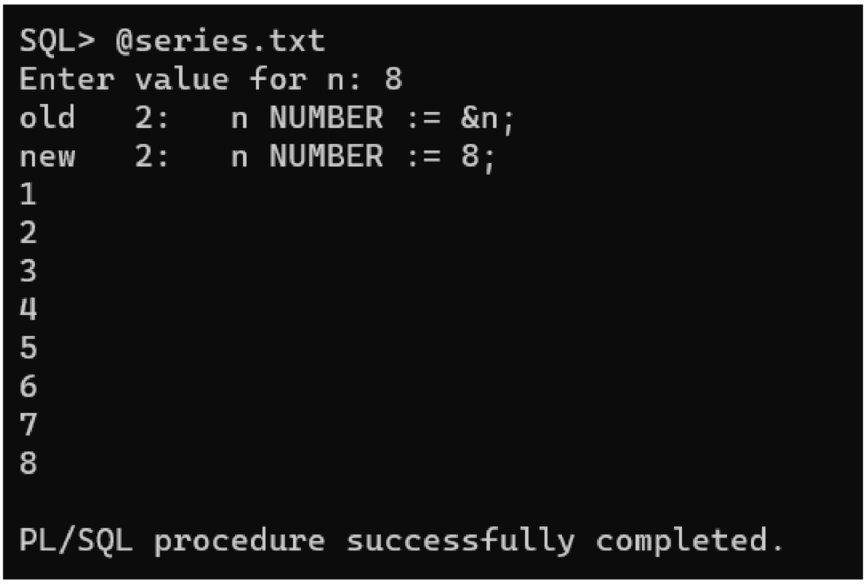


```
SQL> @sql.txt
Total number of employees: 2

PL/SQL procedure successfully completed.
```

10. Write a PLSQL program to display a series of numbers upto n.

```
DECLARE
  n NUMBER := &n;
BEGIN
  FOR i IN 1..n LOOP
    DBMS_OUTPUT.PUT_LINE(i || ' ');
  END LOOP;
END;
/
```



```
SQL> @series.txt
Enter value for n: 8
old   2:   n NUMBER := &n;
new   2:   n NUMBER := 8;
1
2
3
4
5
6
7
8

PL/SQL procedure successfully completed.
```