# ASSIGNMENT – 7

1. **Problem Statement:** Write a PL/SQL code for finding factorial of a given number.

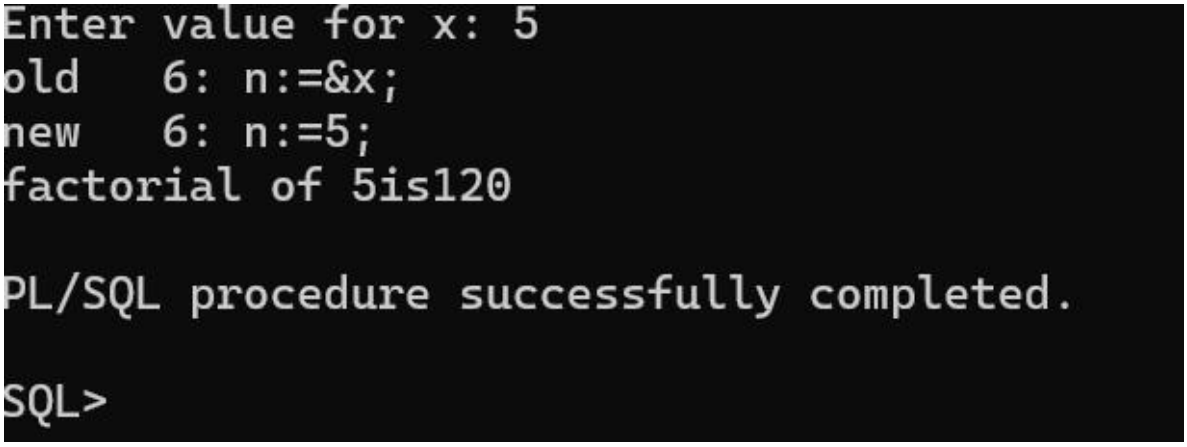
**Query:**

assignment7_1 - Notepad

File Edit Format View Help

```
set serveroutput ON
declare
n number;
i number;
f number:=1;
begin
n:=&X;
for i in 1..n
loop
f:=f*i;
end loop;
dbms_output.put_line('Factorial of ' ||n|| ' is ' ||f);
end;
/
```

**Output:**




2. **Problem Statement:** Write a PL/SQL code for finding sum of N numbers.

**Query:**

assignment7_2 - Notepad

File Edit Format View Help

```
set serveroutput ON
declare
n number;
i number;
s number:=0;
begin
n:=&X;
for i in 1..n
loop
s:=s+i;
end loop;
dbms_output.put_line('SUM of ' ||n|| ' is ' ||s);
end;
/
```

**Output:**

```
Enter value for x: 4
old   6: n:=&x;
new   6: n:=4;
sum of 4is10

PL/SQL procedure successfully completed.

SQL>
```

3. **Problem Statement:** Write a PL/SQL code for finds a given year is leap year or not.

**Query:**

assignment7_3 - Notepad

File Edit Format View Help

```
set serveroutput ON
declare
y number;
begin
    y:=&n;
    if(mod(y,400)=0)then
        dbms_output.put_line('The above year is Leap year');
    elsif((mod(y,4)=0) and (mod(4,100)!=0))
     then
        dbms_output.put_line('The above year Leap year');
    else
        dbms_output.put_line('The above year is not Leap year');
    end if;
end;
/
```

**Output:**

```
Enter value for n: 2024
old   4: y:=&n;
new   4: y:=2024;
Leap Year

PL/SQL procedure successfully completed.

SQL>
```

```
Enter value for n: 2023
old   4: y:=&n;
new   4: y:=2023;
Not a Leap Year

PL/SQL procedure successfully completed.

SQL>
```

4. **Problem Statement:** Write a PL/SQL code for finding maximum of three numbers (Input will be given by the user).

**Query:**

*assignment7_4 - Notepad

File Edit Format View Help

```
set serveroutput ON
declare
a number;
b number;
c number;
begin
    a:=&n;
    b:=&m;
    c:=&p;
    if((a>b) and (a>c))then
        dbms_output.put_line( 'THE HIGHEST NUMBER IS ' ||a);
    elsif((b>a) and (b>c))
     then
        dbms_output.put_line('THE HIGHEST NUMBER IS ' ||b);
    else
        dbms_output.put_line('THE HIGHEST NUMBER IS ' ||c);
    end if;
end;
/
```

**Output:**

```
Enter value for a: 12
old   6: y:=&a;
new   6: y:=12;
Enter value for b: 34
old   7: x:=&b;
new   7: x:=34;
Enter value for c: 4
old   8: z:=&c;
new   8: z:=4;
34 is max

PL/SQL procedure successfully completed.

SQL>
```

5. **<u>Problem Statement</u>:** Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 6 to 10. Store the radius and corresponding values of calculated area in an empty table named Areas, consisting of two columns Radius and Area.

**Query:**

*assignment7_5 - Notepad

File Edit Format View Help

```
set serveroutput ON
drop table areas;
create table areas(radius number(5,3),area number(10,3));
declare
r number;
pi constant number(4,2):=3.14;
area number(10,2);
begin
for r in 6..10
loop
area:=pi*power(r,2);
insert into areas values(r,area);
end loop;
end;
/
```

**Output:**

```
SQL> @D:\sidd57\ass_7_5.txt
SP2-0042: unknown command "ass_7_5" - rest of line ignored.

Table dropped.


Table created.


PL/SQL procedure successfully completed.

SQL> select * from Areas;

    RADIUS       AREA
---------- ----------
         6     113.04
         7     153.86
         8     200.96
         9     254.34
        10        314

SQL>
```

6. **Problem Statement:** Write a PL/SQL code block that will accept a client_no from the user and adds the amount of Rs. 1000 to bal_due column, has a minimum balance of Rs. 6000. The process is fire on client_master.

**Query:**

*assignment7_6 - Notepad

File Edit Format View Help

```
set serveroutput ON
declare
Cli_no varchar2(6):='&Client_number';
t_c_no number(10,2);
begin
select balance_due into t_c_no from Client_master where Client_number = Cli_no;
if t_c_no>=6000 then
        t_c_no:=t_c_no+1000;
        update Client_master set balance_due = t_c_no where Client_number = Cli_no;
        dbms_output.put_line('The balance due is updated Succesfully');
end if;
end;
/
```

**Output:**

```
SQL> @D:\sidd57\ass_7_6.txt
Enter value for cilent_no: C001
old   2: cli_No varchar2(6):='&Cilent_No';
new   2: cli_No varchar2(6):='C001';

PL/SQL procedure successfully completed.

SQL> @D:\sidd57\ass_7_6.txt
Enter value for cilent_no: C003
old   2: cli_No varchar2(6):='&Cilent_No';
new   2: cli_No varchar2(6):='C003';
The Balance is below 6000.

PL/SQL procedure successfully completed.

SQL>
```