**EXPERIMENT NO. 1**

Write PL/SQL code block to display the word “Hello!”

**PROGRAM**

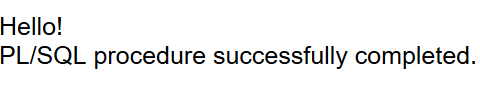
set serveroutput on

BEGIN

dbms\_output.put\_line('Hello!');

END;

**Output:**



**EXPERIMENT NO. 2**

Write PL/SQL code block which will get the salary of an employee with particular id from emp table and display it on screen.

**PROGRAM**

create table emp\_detail(id number, name char(30), salary number(6));

insert into emp\_detail values(101, 'sargam', 25000);

insert into emp\_detail values(102, 'harman', 30000);

insert into emp\_detail values(103, 'sargun', 50000);

insert into emp\_detail values(104, 'sehaj', 27000);

insert into emp\_detail values(105, 'shetal', 45000);

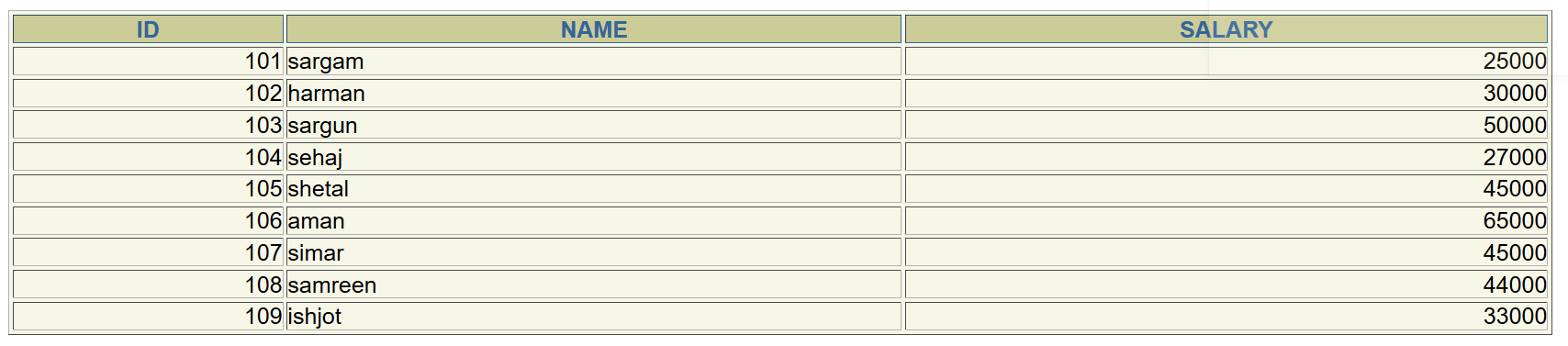
insert into emp\_detail values(106, 'aman', 65000);

insert into emp\_detail values(107, 'simar', 45000);

insert into emp\_detail values(108, 'samreen', 44000);

insert into emp\_detail values(109, 'ishjot', 33000);

select \* from emp\_detail;



**Method 1:**

set serveroutput on

DECLARE

emp\_rec emp\_detail%rowtype;

BEGIN

select \* into emp\_rec from emp\_detail where id = 104;

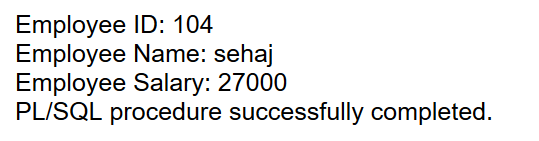
dbms\_output.put\_line('Employee ID: ' || emp\_rec.id);

dbms\_output.put\_line('Employee Name: ' || emp\_rec.name);

dbms\_output.put\_line('Employee Salary: ' || emp\_rec.salary);

END;

**Output:**



**Method 2:**

set serveroutput on

DECLARE

x varchar(10);

a varchar(10);

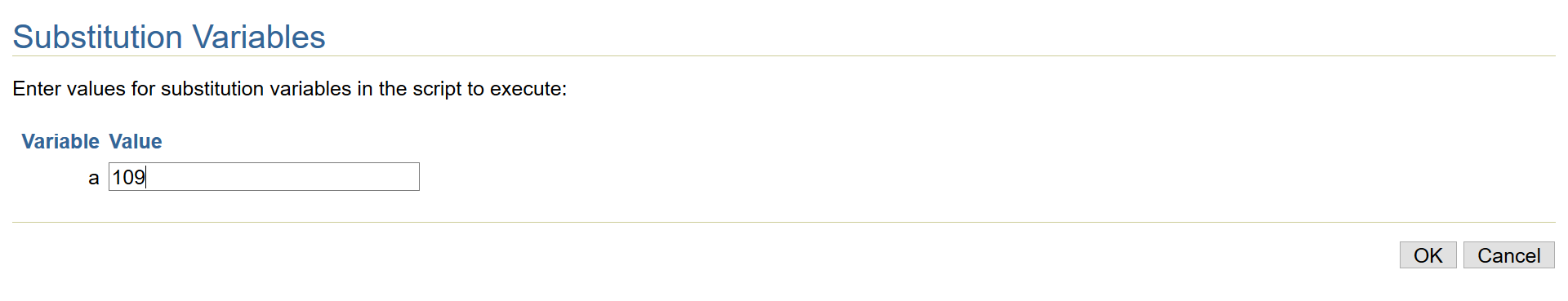
BEGIN

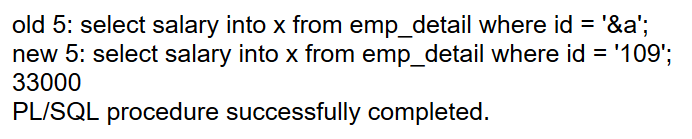
select salary into x from emp\_detail where id = '&a';

dbms\_output.put\_line(x);

END;

**Output:**





**EXPERIMENT NO. 3**

Write PL/SQL code block to calculate the sum by taking input of two numbers.

**PROGRAM**

set serveroutput on

DECLARE

no1 number(7);

no2 number(7);

result number(7);

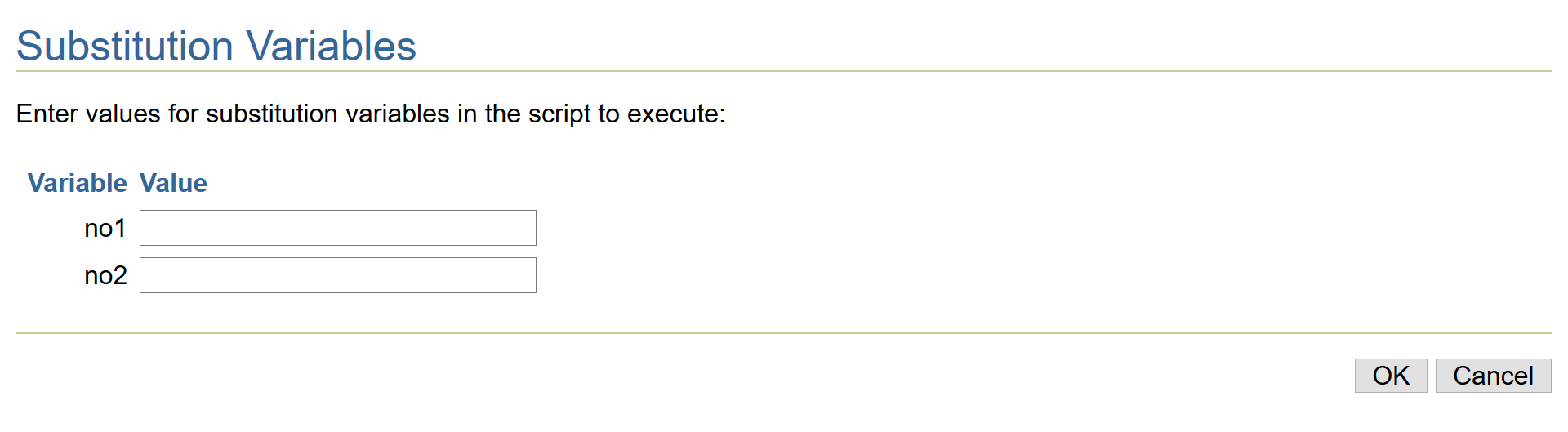
BEGIN

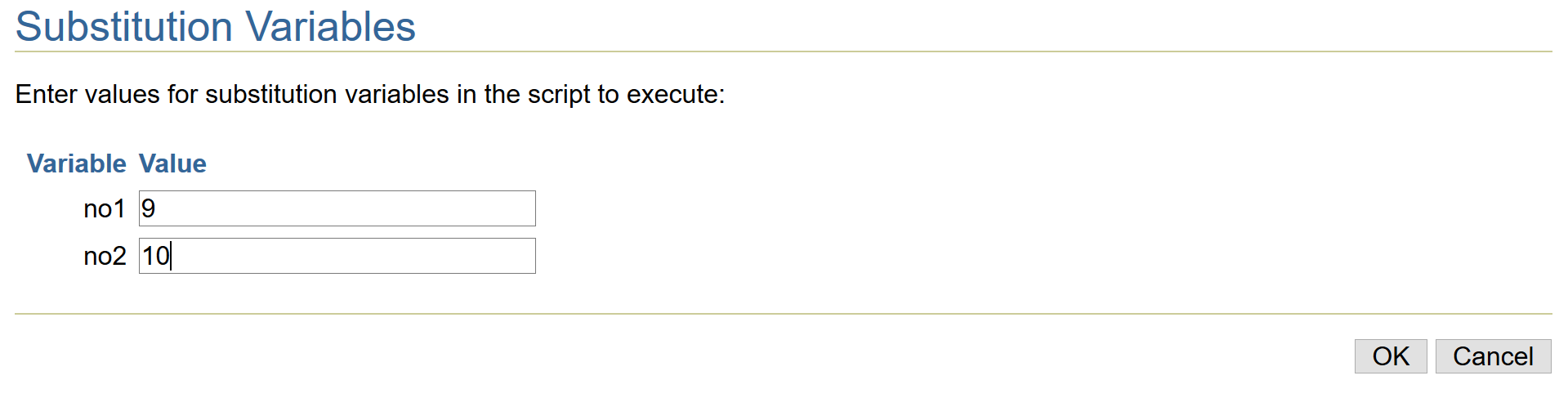
result:= &no1 + &no2;

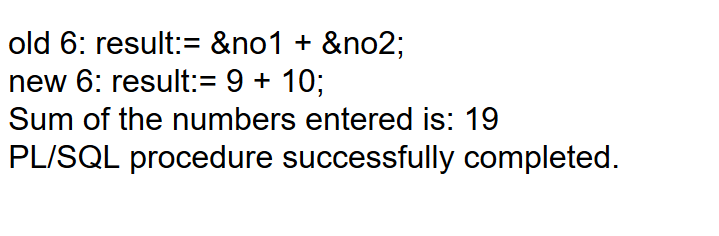
dbms\_output.put\_line('Sum of the numbers entered is: '||result);

END;

**Output:**







**EXPERIMENT NO. 4**

Write PL/SQL code block which creates two variables in the outer block and assign their product to the third variable created in the inner block.

**PROGRAM**

set serverouput on

DECLARE

no1 number(7);

no2 number(7);

BEGIN

no1 := &no1;

no2 := &no2;

DECLARE

result number(15);

BEGIN

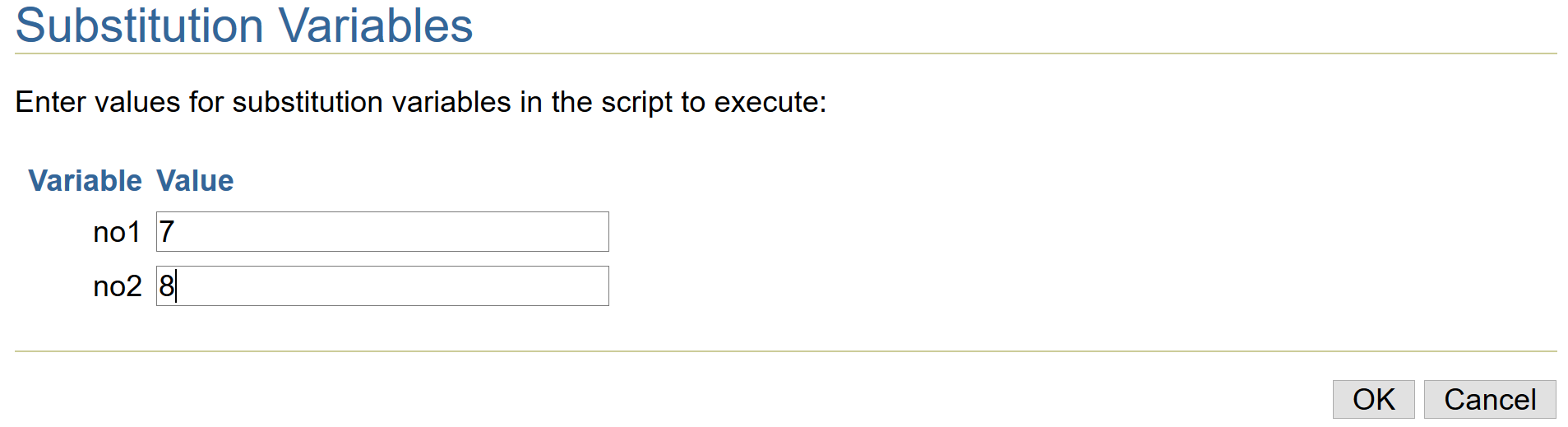
result := no1 \* no2;

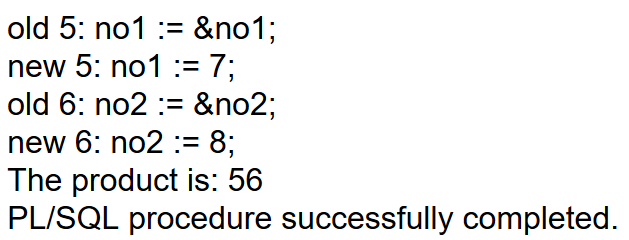
dbms\_output.put\_line('The product is: '||result);

END;

END;

**Output:**





**EXPERIMENT NO. 5**

Write PL/SQL code block to declare a record called employee\_rec based on user-defined datatype.

**PROGRAM**

set serveroutput on

DECLARE

type employee is record

( id number(4),

name varchar(30),

salary number(6));

employee\_rec employee;

BEGIN

select id, name, salary into employee\_rec.id, employee\_rec.name, employee\_rec.salary from emp\_detail where id = 109;

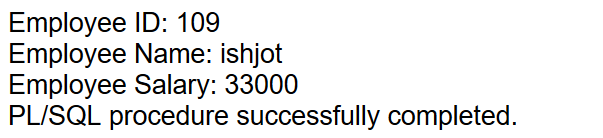
dbms\_output.put\_line('Employee ID: ' || employee\_rec.id);

dbms\_output.put\_line('Employee Name: ' || employee\_rec.name);

dbms\_output.put\_line('Employee Salary: ' || employee\_rec.salary);

END;

**Output:**



**EXPERIMENT NO. 6**

Write PL/SQL code block for a procedure that has four sections. Each section should output a statement use labels and goto command to output the section message in the following order.

Section 3

Section 2

Section 1

Section 4

**PROGRAM**

set serveroutput on

BEGIN

dbms\_output.put\_line('Gursimar Kaur');

dbms\_output.put\_line('1820036');

goto section3;

<<section1>>

dbms\_output.put\_line('Section 1');

goto section2;

<<section2>>

dbms\_output.put\_line('Section 2');

goto section4;

<<section3>>

dbms\_output.put\_line('Section 3');

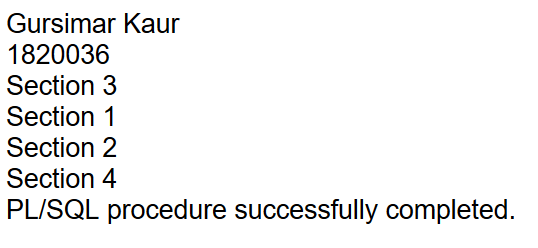
goto section1;

<<section4>>

dbms\_output.put\_line('Section 4');

END;

**Output:**



**EXPERIMENT NO. 7**

Write PL/SQL code block which use the relational operators to compare character values for equality or inequality.

**PROGRAM**

set serveroutput on

DECLARE

Name varchar(20);

Company varchar(30);

Introduction CLOB;

Choice varchar(1);

BEGIN

Name := 'Gursimar';

Company := 'Google';

Introduction := 'Smart and talented';

choice := '&choice';

if choice = 'y' then

dbms\_output.put\_line(Name);

dbms\_output.put\_line(Company);

dbms\_output.put\_line(Introduction);

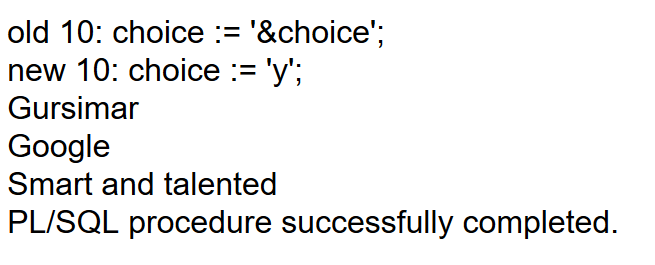
else

dbms\_output.put\_line('wrong input');

end if;

END;

**Output:**



**EXPERIMENT NO. 8**

Write PL/SQL code block which use the if statement.

**PROGRAM**

set serveroutput on

DECLARE

no1 number(4);

no2 number(4);

no3 number(4);

min\_no number(4);

Procedure find\_min(x In number, y In number, z In number, min Out number) IS

BEGIN

if x<y then

if x<z then

min := x;

else

min := z;

end if;

else

if y<z then

min := y;

else

min := z;

end if;

end if;

end find\_min;

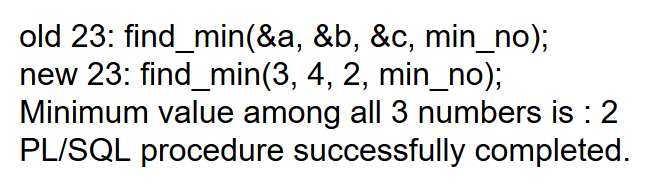
BEGIN

find\_min(&a, &b, &c, min\_no);

dbms\_output.put\_line('Minimum value among all 3 numbers is : '||min\_no);

END;

**Output:**



**EXPERIMENT NO. 9**

Write PL/SQL code block which demonstrate the use of “SIMPLE LOOP”.

**PROGRAM**

set serveroutput on

declare

num number:=10;

begin

dbms\_output.put\_line('Gursimar Kaur');

dbms\_output.put\_line('1820036');

loop

dbms\_output.put\_line('Number: ' ||num);

num:= num - 1;

if num = 0 then

Exit;

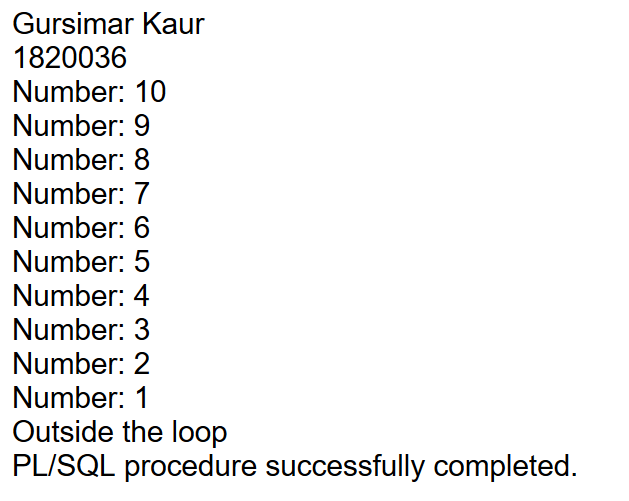
end if;

end loop;

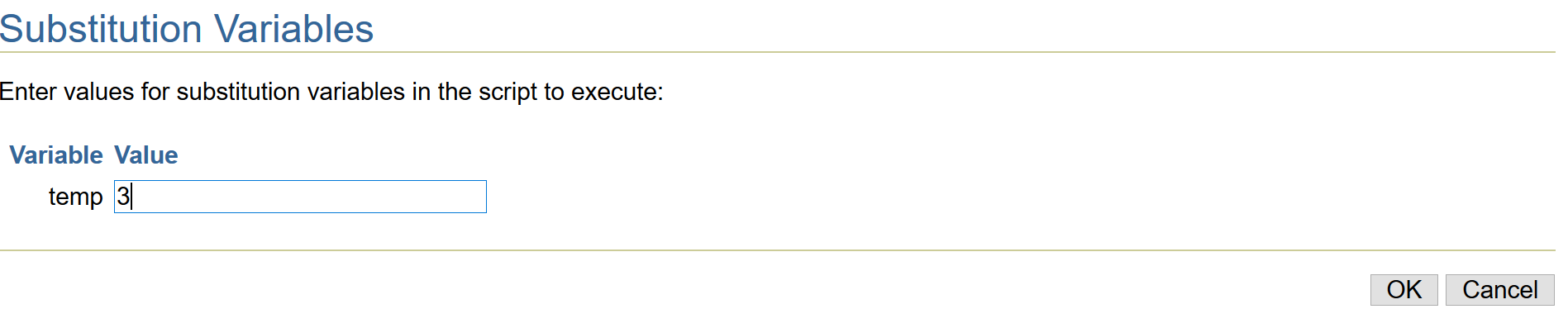
dbms\_output.put\_line('Outside the loop');

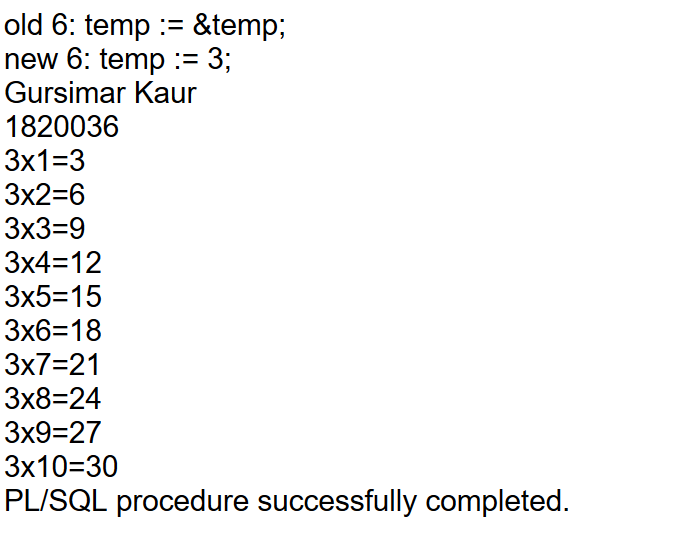
end;

**Output:**



**Output:**





**EXPERIMENT NO. 10**

Write PL/SQL code block which demonstrate the use of “FOR LOOP”.

**PROGRAM**

set serveroutput on

declare

temp number(4);

begin

dbms\_output.put\_line('Gursimar Kaur');

dbms\_output.put\_line('1820036');

temp := &temp;

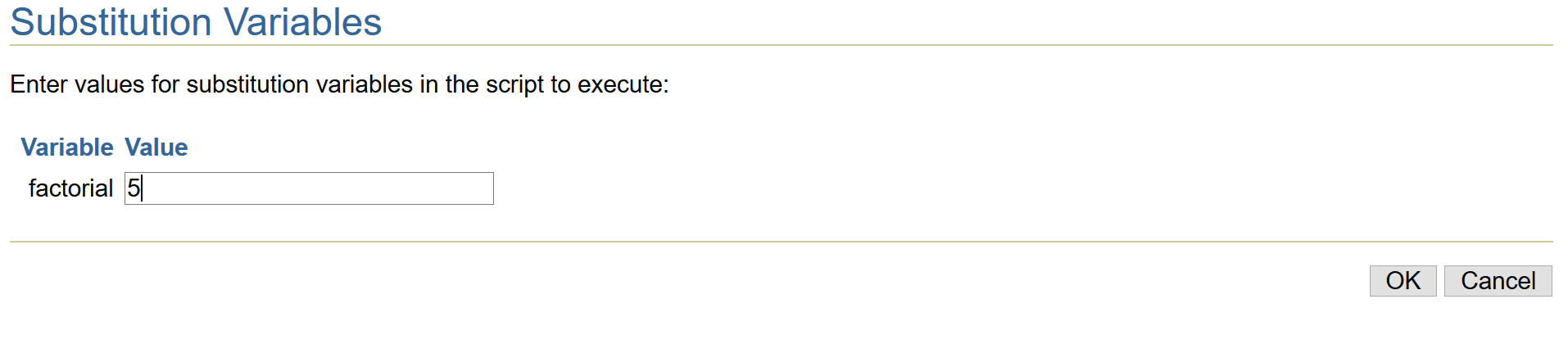
for n in 1..10 loop

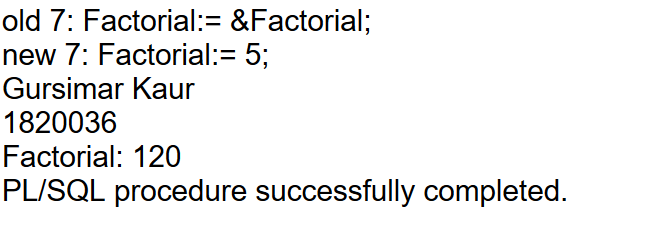
dbms\_output.put\_line(temp ||'x' ||n ||'=' || temp \* n);

end loop;

end;

**Output:**





**EXPERIMENT NO. 11**

Write PL/SQL code block which demonstrate the use of “WHILE LOOP”.

**PROGRAM**

set serveroutput on

declare

Factorial number(4);

result number(4);

begin

dbms\_output.put\_line('Gursimar Kaur');

dbms\_output.put\_line('1820036');

Factorial:= &Factorial;

result := 1;

while Factorial>0

loop

result := Factorial\*result;

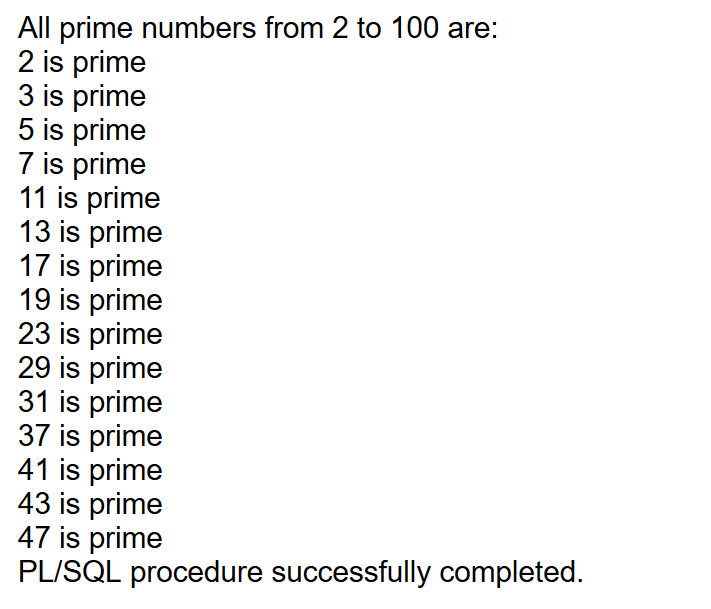
Factorial := Factorial-1;

end loop;

dbms\_output.put\_line('Factorial: ' || result);

end;

**Output:**



**EXPERIMENT NO. 12**

Write PL/SQL code block which demonstrate the use of “NESTED LOOP”.

**PROGRAM**

DECLARE

i number(3);

j number(3);

BEGIN

i := 2;

LOOP

j:= 2;

LOOP

exit WHEN ((mod(i, j) = 0) or (j = i));

j := j +1;

END LOOP;

IF (j = i ) THEN

dbms\_output.put\_line(i || ' is prime');

END IF;

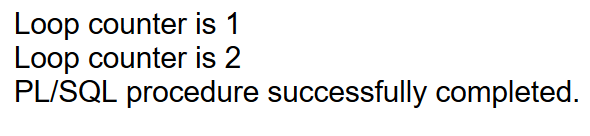
i := i + 1;

exit WHEN i = 50;

END LOOP;

END;

**Output:**



**EXPERIMENT NO. 13**

Write PL/SQL code block which demonstrate the use of “LABELLING LOOP”.

**PROGRAM**

BEGIN

<<looplabel>>

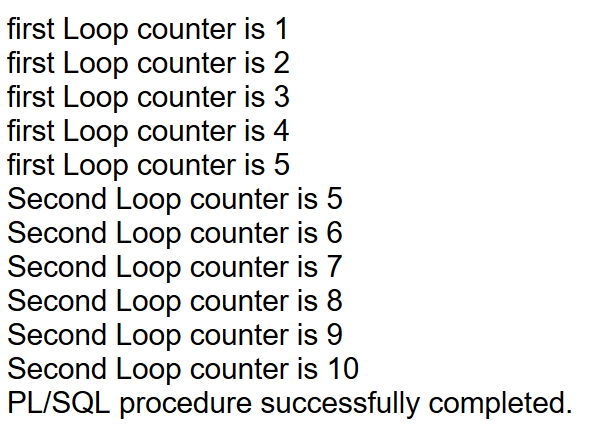
FOR i IN 1..2 LOOP

DBMS\_OUTPUT.PUT\_LINE('Loop counter is ' || i);

END LOOP looplabel;

END;

**Output:**



**EXPERIMENT NO. 13**

Write PL/SQL code block which demonstrate the use of “LABELLING LOOP”.

**PROGRAM**

BEGIN

<<firstloop>>

FOR i IN 1..10 LOOP

DBMS\_OUTPUT.PUT\_LINE('first Loop counter is ' || i);

if i = 5 then

goto secondloop;

end if;

end loop firstloop;

<<secondloop>>

for i in 5..10 loop

dbms\_output.put\_line('Second Loop counter is ' || i);

end loop secondloop;

END;