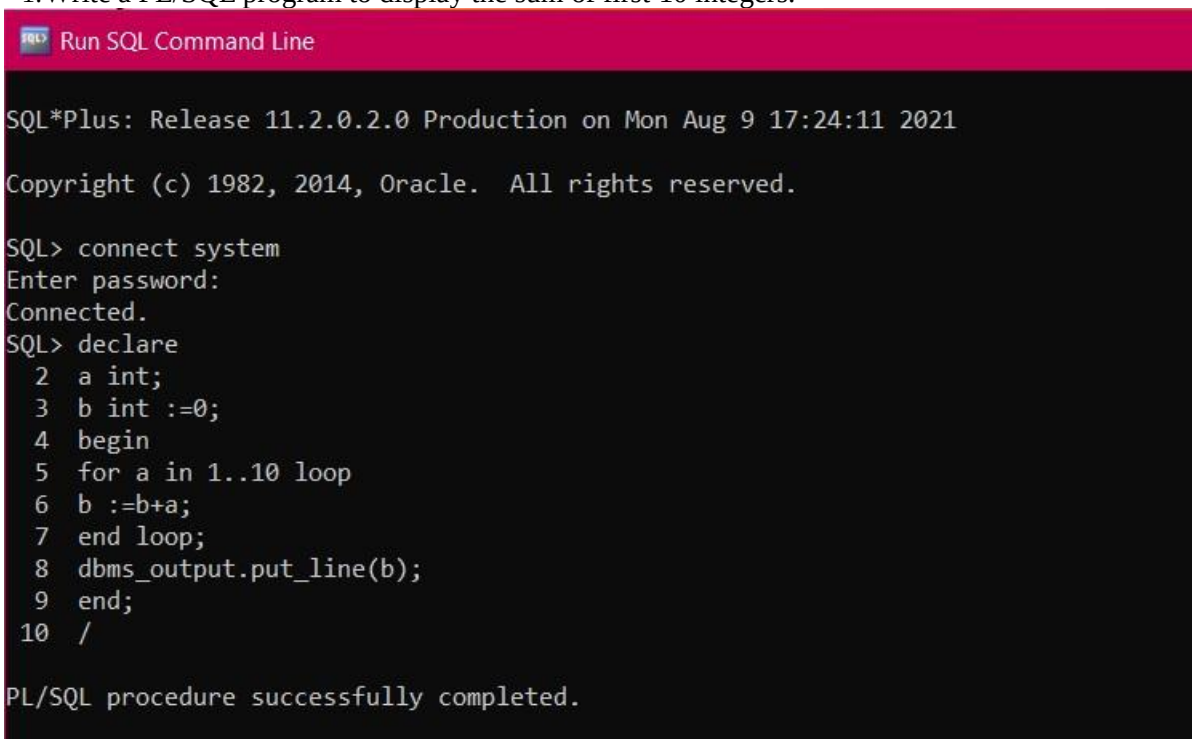
Name**:-**Aniket Prajapati

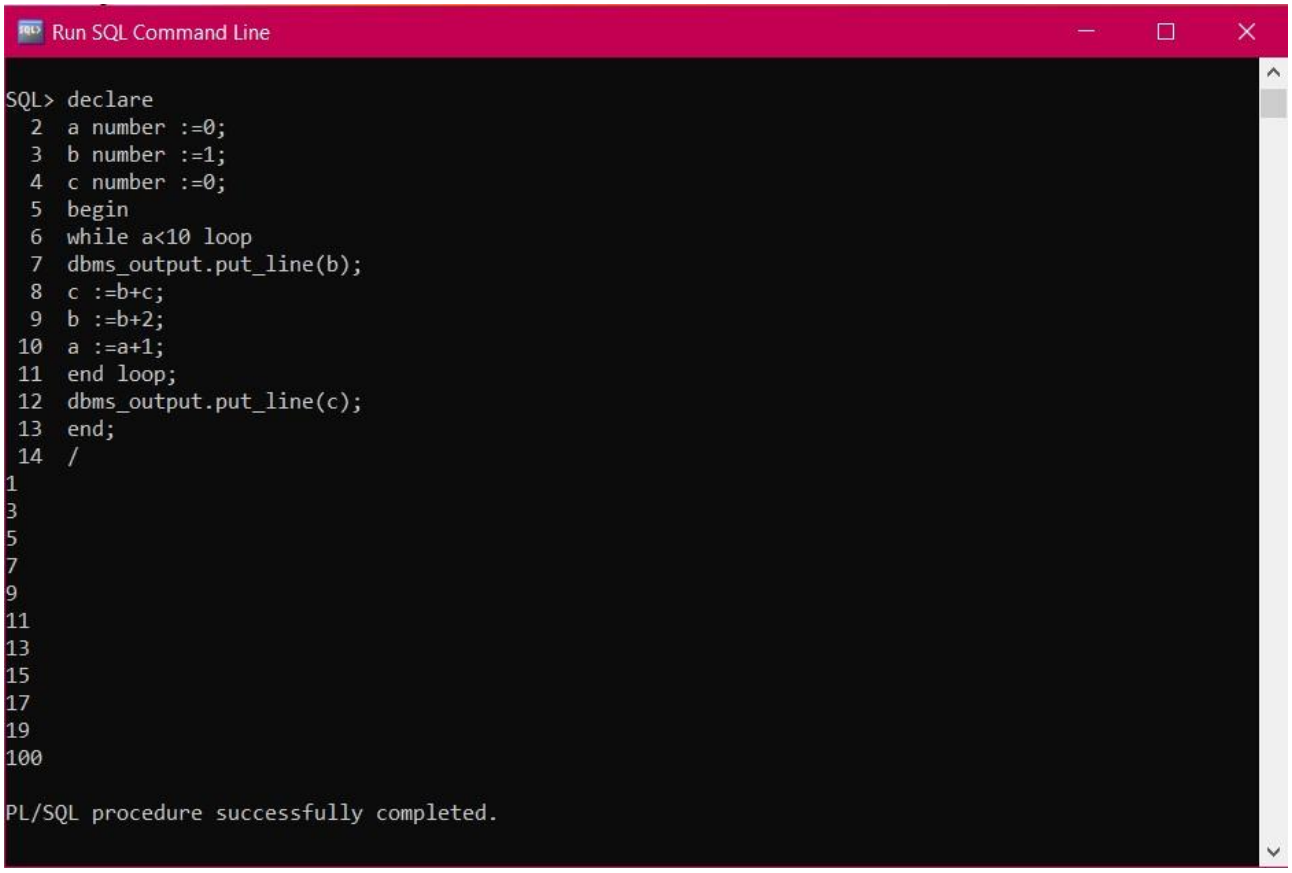
SYCS--351

PRACTICAL NO:2

1.Write a PL/SQL program to display the sum of first 10 integers.



2.Write a PL/SQL program to display the sum of first 10 odd numbers.


Run SQL Command Line
SQL> declare
2 a number :=0;
3 b number :=1;
4 c number :=0;
5 begin
6 while a<10 loop
7 dbms_output.put_line(b);
8 c :=b+c;
9 b :=b+2;
10 a :=a+1;
11 end loop;
12 dbms_output.put_line(c);
13 end;
14 /
1
3
5
7
9
11
13
15
17
19
100
PL/SQL procedure successfully completed.

3. Write a PL/SQL program to calculate the area of a circle and insert the area and radius in a table aoc (sno,radius,area) till radius is less than 10.

```
SQL> declare
  2  a number;
  3  radius number :=1;
  4  area number;
  5  pi number :=3.14;
  6  begin
  7  for a in 1..9 loop
  8  radius := a;
  9  area := radius * radius * pi;
 10  insert into aco values(a,radius,area);
 11  end loop;
 12  end;
 13  /

PL/SQL procedure successfully completed.
```

4. Write a PL/SQL program to reverse the number (234 as 432).

Run SQL Command Line

```
SQL> declare
  2  a number :=234;
  3  b number;
  4  begin
  5  b :=0;
  6  while a>0 loop
  7  b :=(b * 10) + mod(a,10);
  8  a :=floor(a/10);
  9  end loop;
 10  dbms_output.put_line(b);
 11  end;
 12  /
432

PL/SQL procedure successfully completed.
```

5. Write a PL/SQL program to print the length of entered string.

Run SQL Command Line

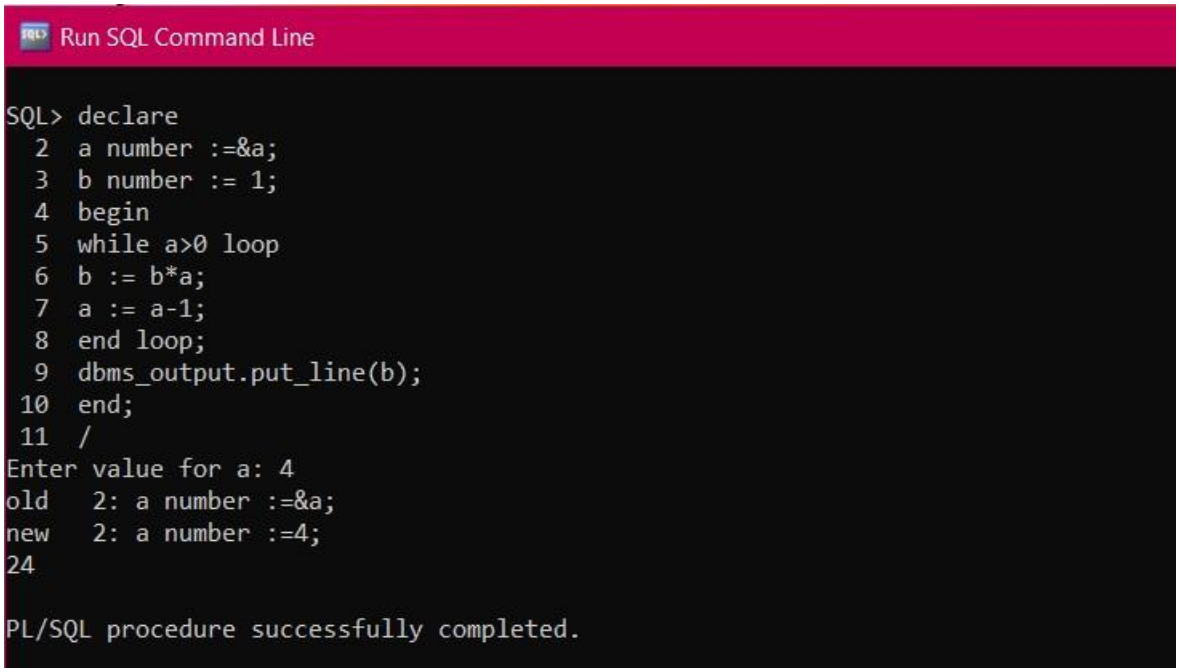
```
SQL> declare
  2  a varchar2(20) := '&a';
  3  begin
  4  dbms_output.put_line(length(a));
  5  end;
  6  /
Enter value for a: hello
old   2: a varchar2(20) := '&a';
new   2: a varchar2(20) := 'hello';
5

PL/SQL procedure successfully completed.
```

6. Write a PL/SQL program to count number of employees in dept 10, and if they are greater than 3 print the count otherwise do nothing.

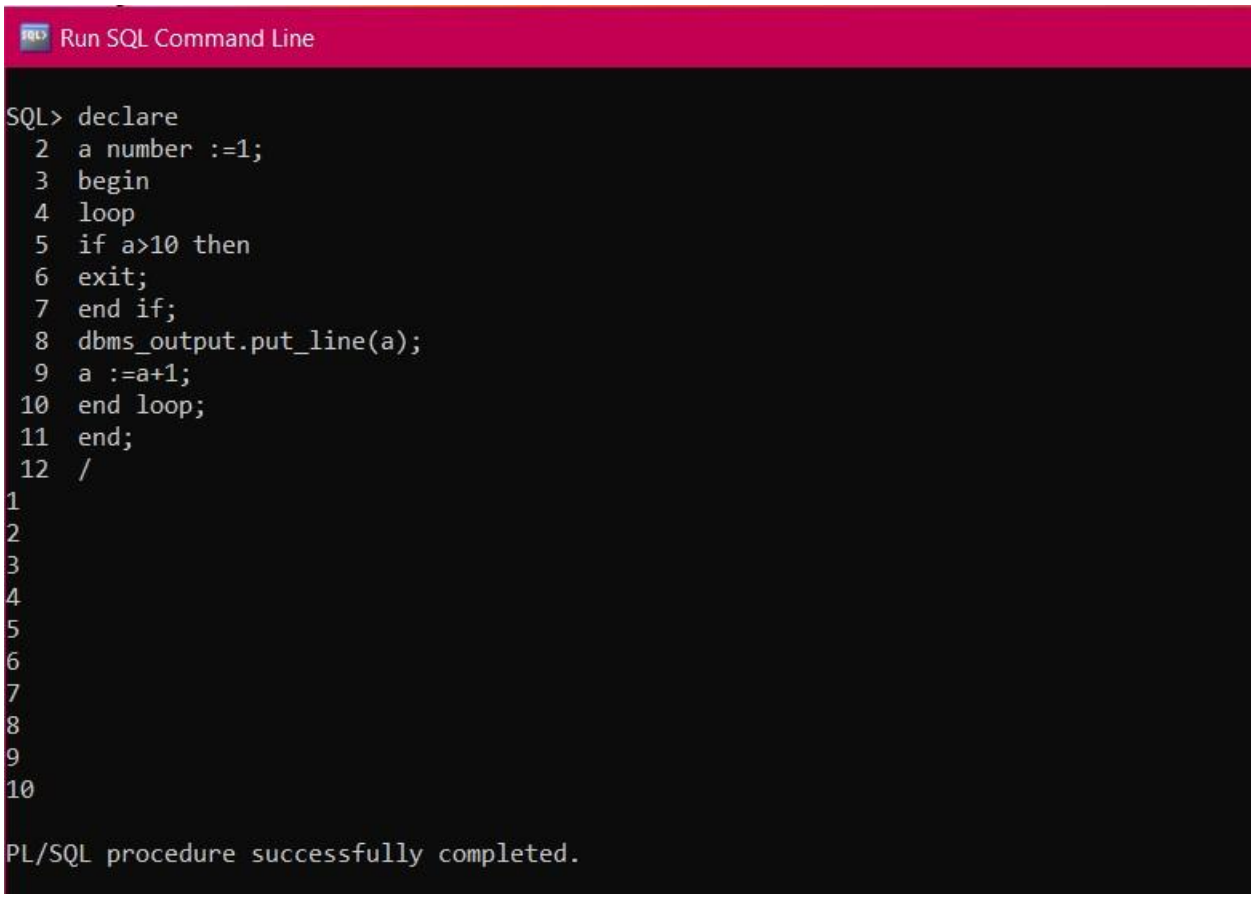
Run SQL Command Line

```
SQL> declare
  2  num number;
  3  begin
  4  select count(*) into num from aniket_emp351 where deptno = 10;
  5  if num > 3 then
  6  dbms_output.put_line(num || 'Employee in deptno 10');
  7  end if;
  8  end;
  9  /

PL/SQL procedure successfully completed.
```

7. Write a PL/SQL block to find the factorial of a number.



8. Write a block to display the number from 1 to 10 using unconstraint loop.



9. Write a PL/SQL block using CASE statement to accept the owner name from the user.

The user name can be SYS, SYSTEM, HR or SCOTT.

If the owner name is SYS then print the result is ‘The Owner is SYS’. If the owner name is SYSTEM then print the result is ‘The Owner is SYSTEM’.

If the owner name is HR then print the result is ‘The Owner is HR’. If the owner name is SCOTT then print the result is ‘The Owner is SCOTT’. Otherwise print ‘Invalid Choice’.

Run SQL Command Line

```
SQL> declare
  2  name varchar2(20) := '&name';
  3  begin
  4  case name
  5  when 'sys' then dbms_output.put_line('the owner is '||name);
  6  when 'system' then dbms_output.put_line('the owner is '||name);
  7  when 'hr' then dbms_output.put_line('the owner is '||name);
  8  when 'scott' then dbms_output.put_line('the owner is '||name);
  9  else
 10  dbms_output.put_line('Invalid Choise');
 11  end case;
 12  end;
 13  /
Enter value for name: sys
old   2: name varchar2(20) := '&name';
new   2: name varchar2(20) := 'sys';
the owner is sys

PL/SQL procedure successfully completed.
```

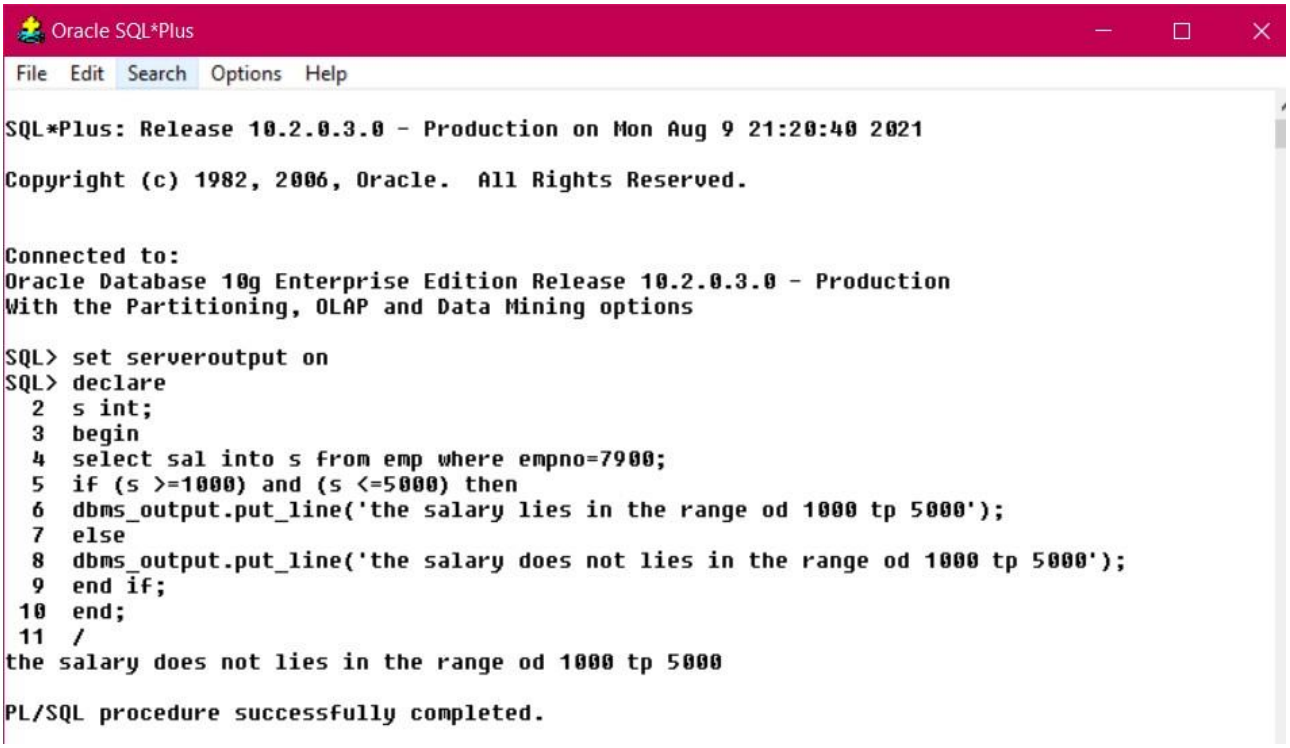
10. Write a PL/SQL block to find factorial of a number which is accepted by the user and store it under the table fac(num,fact).

```
Run SQL Command Line

PL/SQL procedure successfully completed.

SQL> create  table fac(a number,b number);

Table created.

SQL> declare
  2  a number := &a;
  3  b number := 1;
  4  c number;
  5  begin
  6  c := a;
  7  while c>0 loop
  8  b := b*c;
  9  c := c-1;
 10  end loop;
 11  insert into fac values(a,b);
 12  end;
 13  /
Enter value for a: 5
old   2: a number := &a;
new   2: a number := 5;

PL/SQL procedure successfully completed.
```

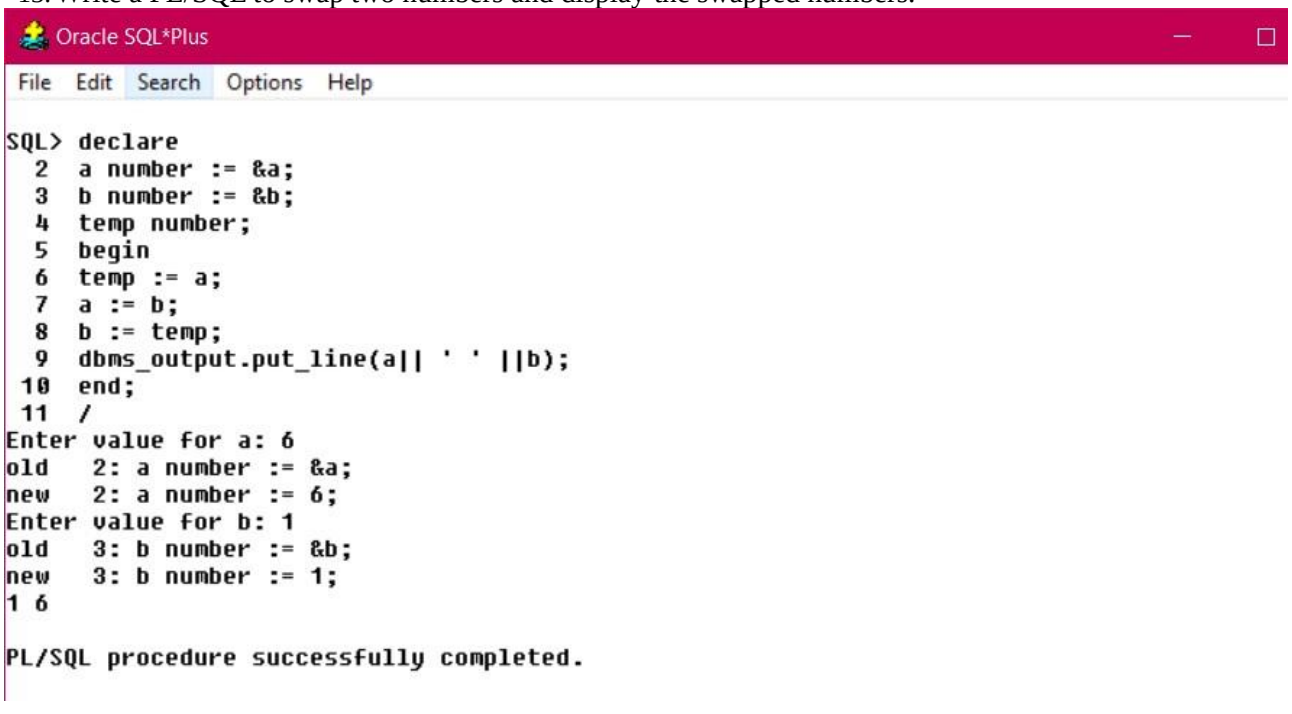
11. Write a PL/SQL to read a number and check whether it is greater than 100 or not and print appropriate message.

```
Run SQL Command Line

SQL> declare
  2  a number :=&a;
  3  begin
  4  if a>100 then
  5  dbms_output.put_line('the given number is greater than 100');
  6  else
  7  dbms_output.put_line('the given number is smaller than 100');
  8  end if;
  9  end;
 10  /
Enter value for a: 52
old   2: a number :=&a;
new   2: a number :=52;
the given number is smaller than 100

PL/SQL procedure successfully completed.
```

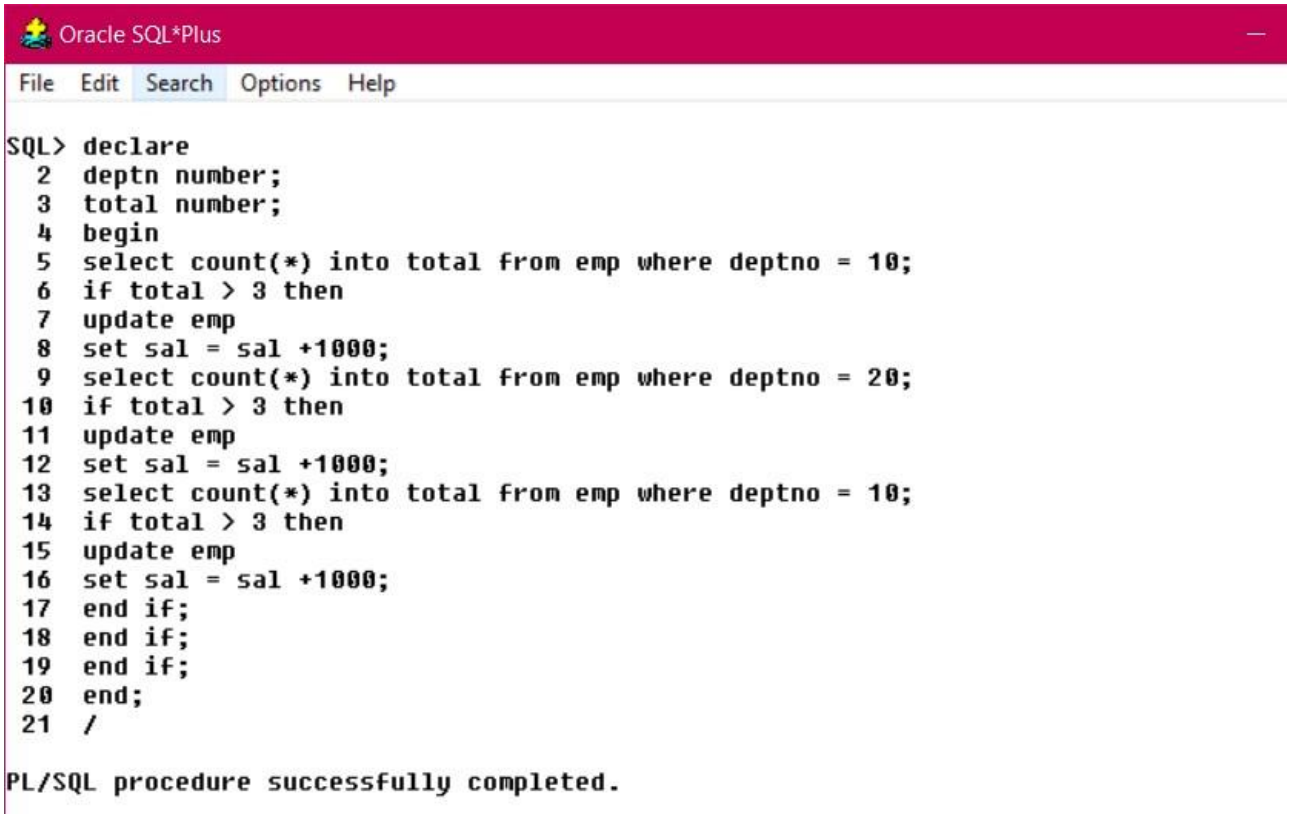
12. Write a PL/SQL to read the salary of an employee 7900 and display the appropriate message if it lies in the range of 1000 and 5000.



13. Write a PL/SQL to swap two numbers and display the swapped numbers.

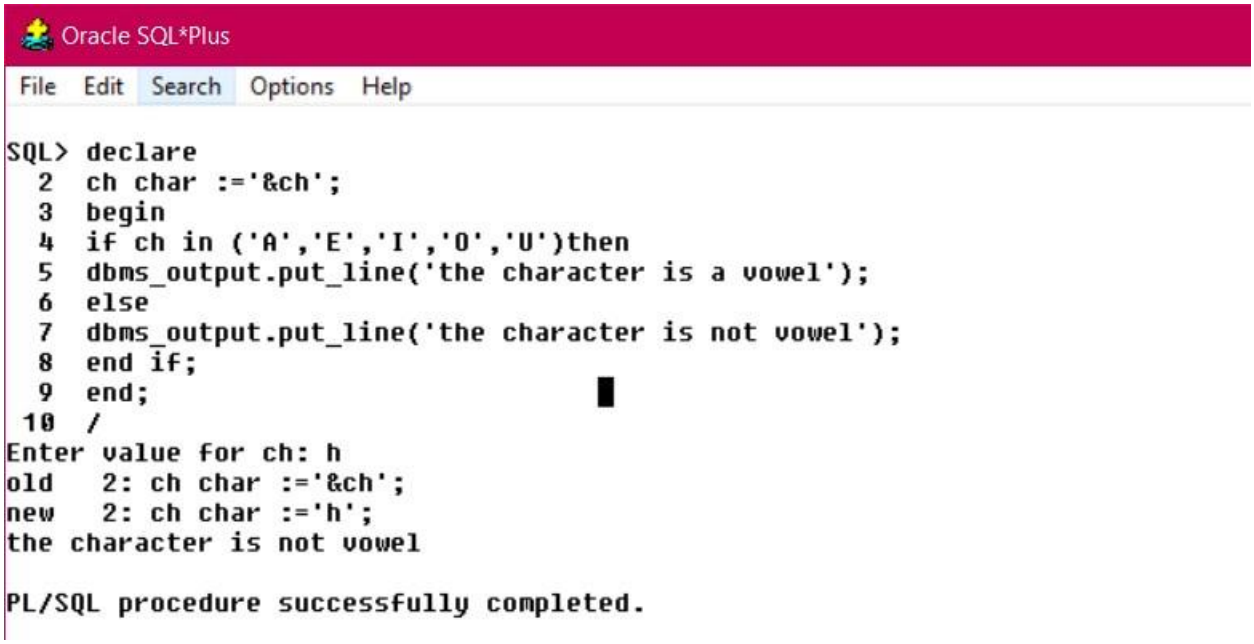


14. Write a PL/SQL block to update the salary of the employee with 1000 when total number of employees in a particular department is greater than 3.

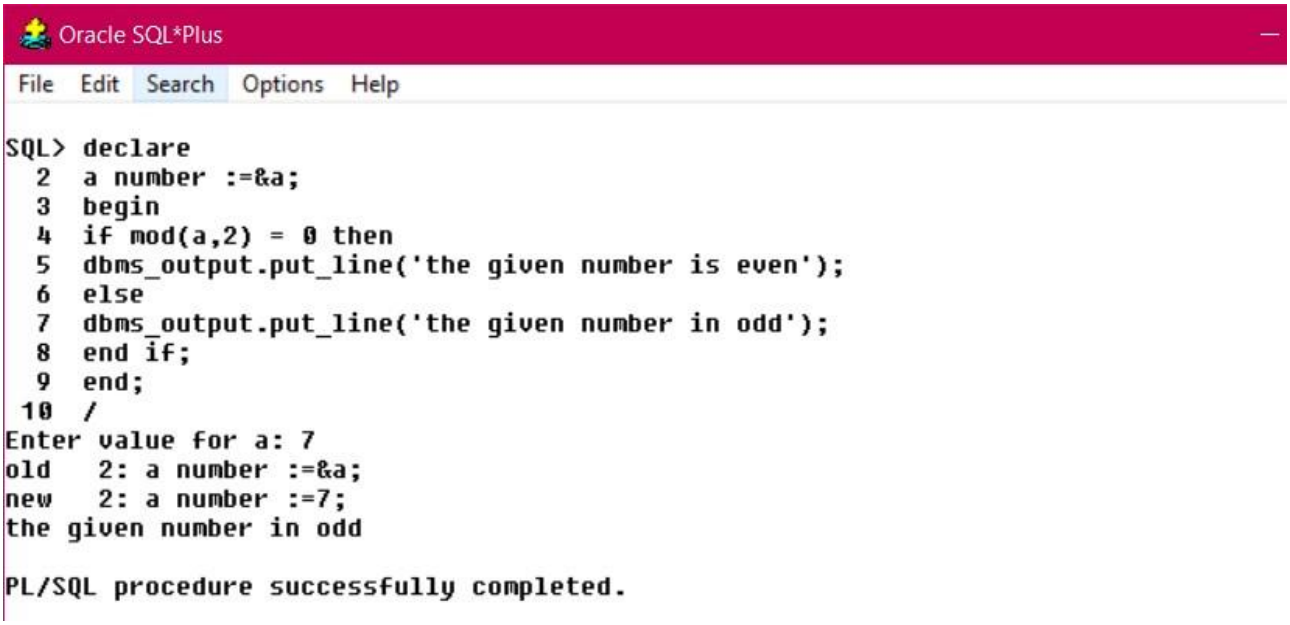


15. Write a PL/SQL block to delete the records of the table employee by accepting the table name from the user.

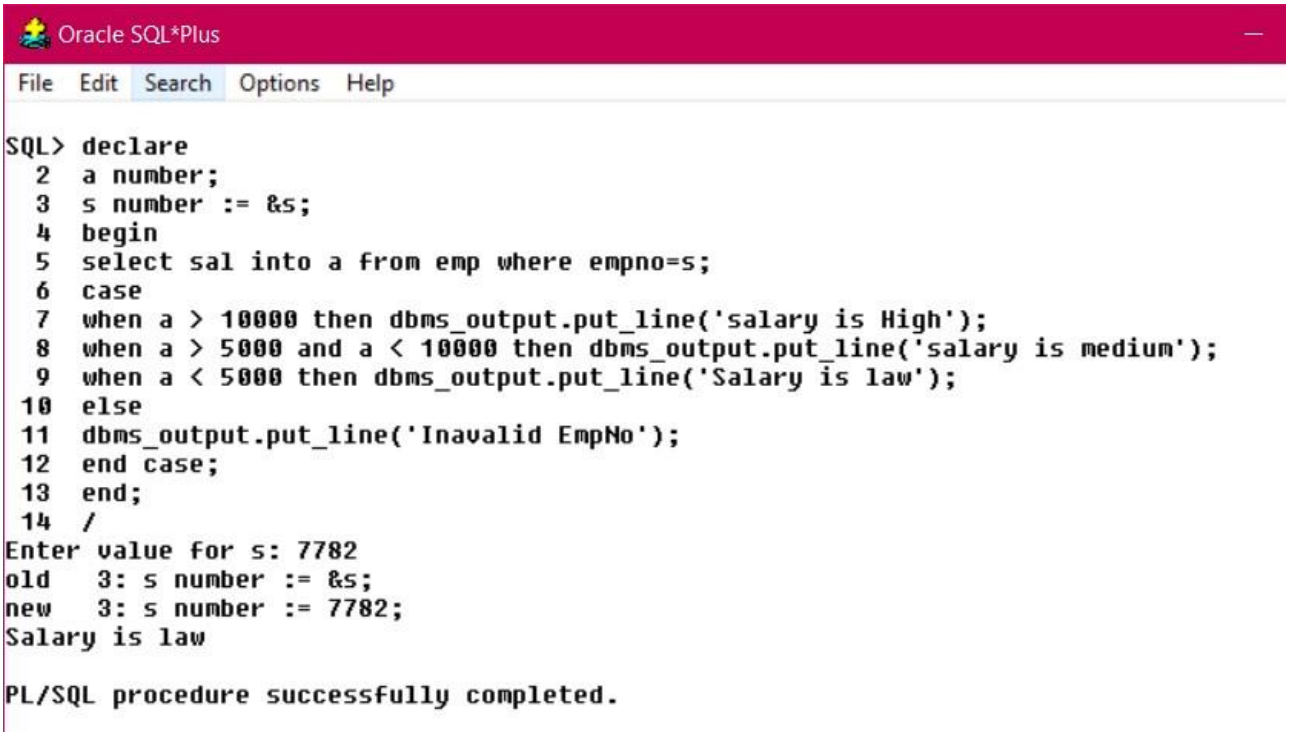
16. Write a PL/SQL to check whether the character entered is a vowel or not.



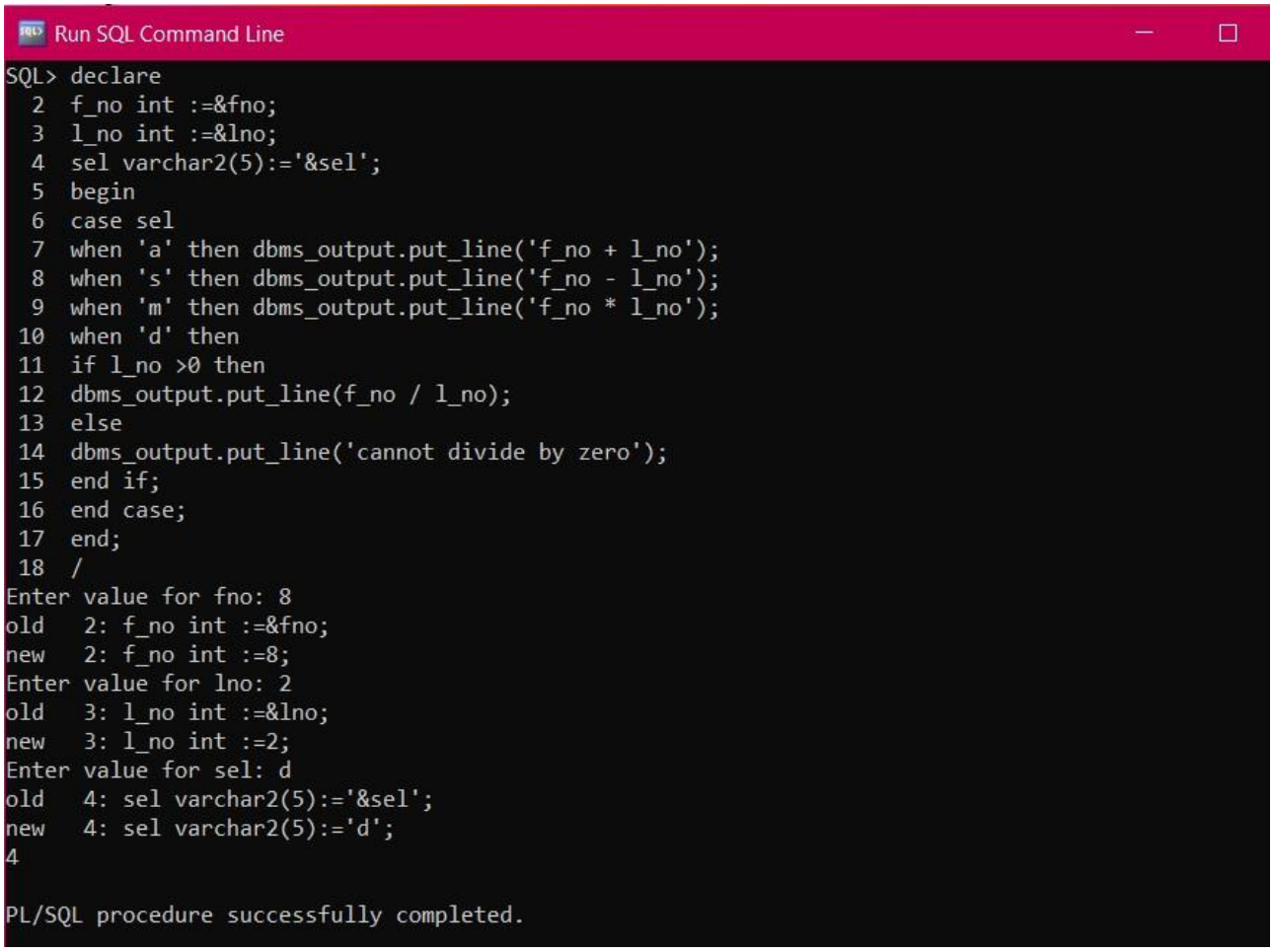
17. Write a PL/SQL to check whether a number is even or odd.



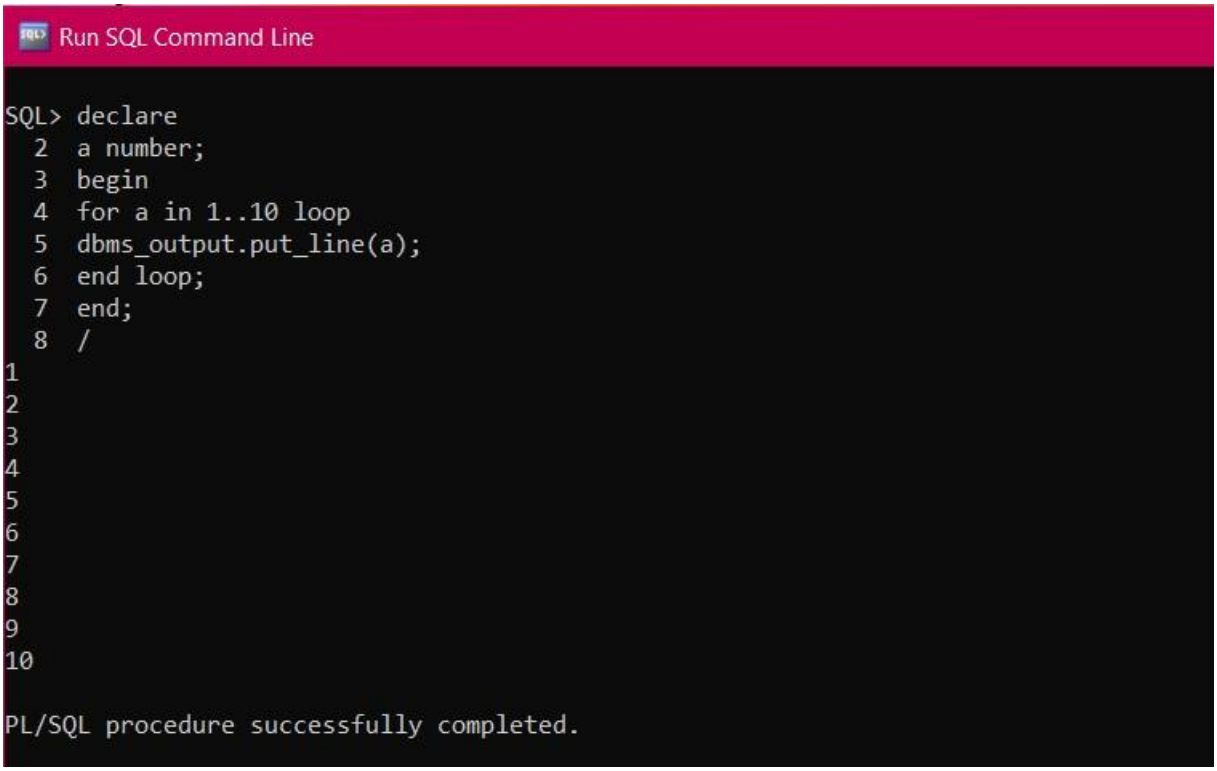
18. Write a PL/SQL block using case statement to print the salary as high if it is greater then 10000, moderate if it is between 5000 and 10000 and low if it is less than 5000.The salary has been taken as an input of a specific employee whose empid is accepted by the user.



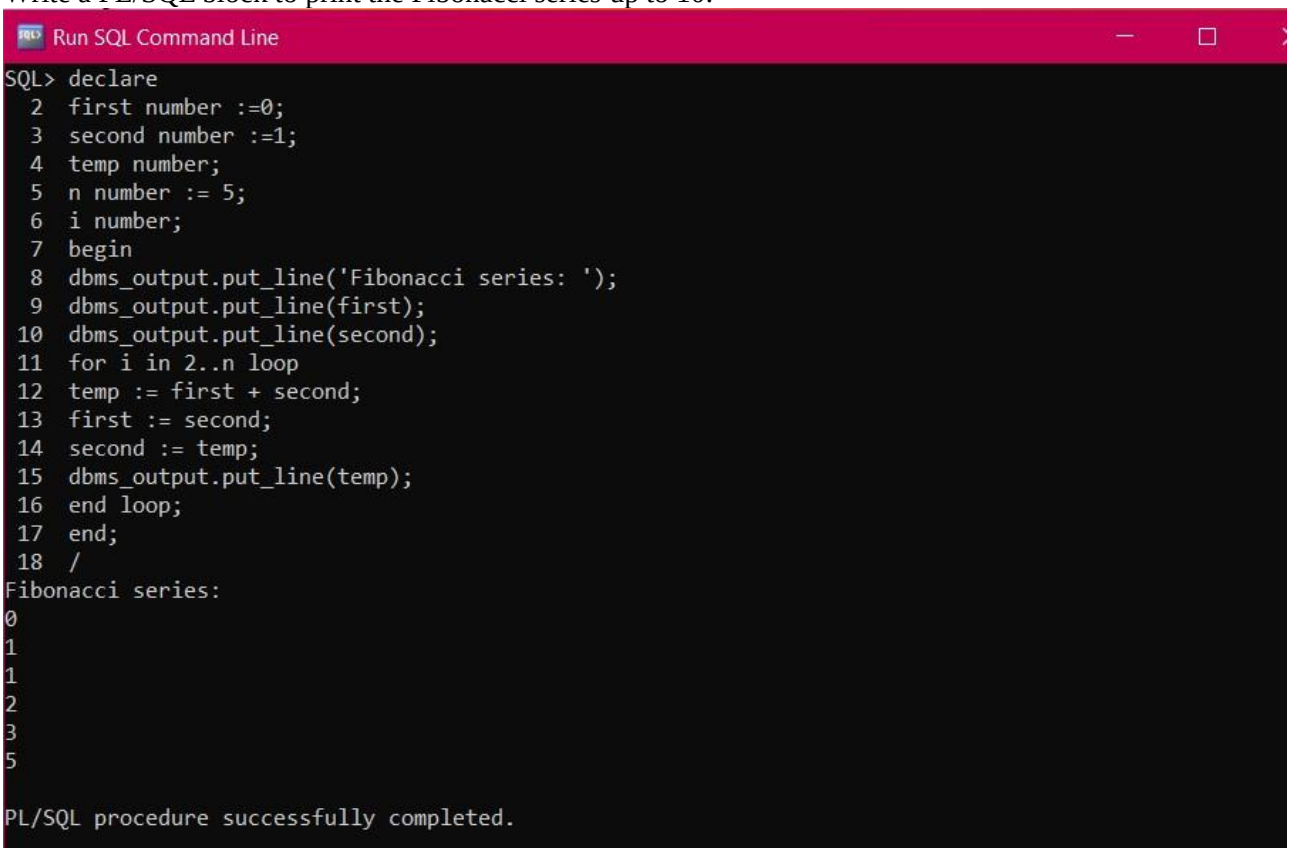
19. Write a PL/SQL block using case statement to perform addition, subtraction, multiplication and division for the individual choices a ,s ,m, d.The division can only take place if the divisor is greater than 0 else error message should be printed.



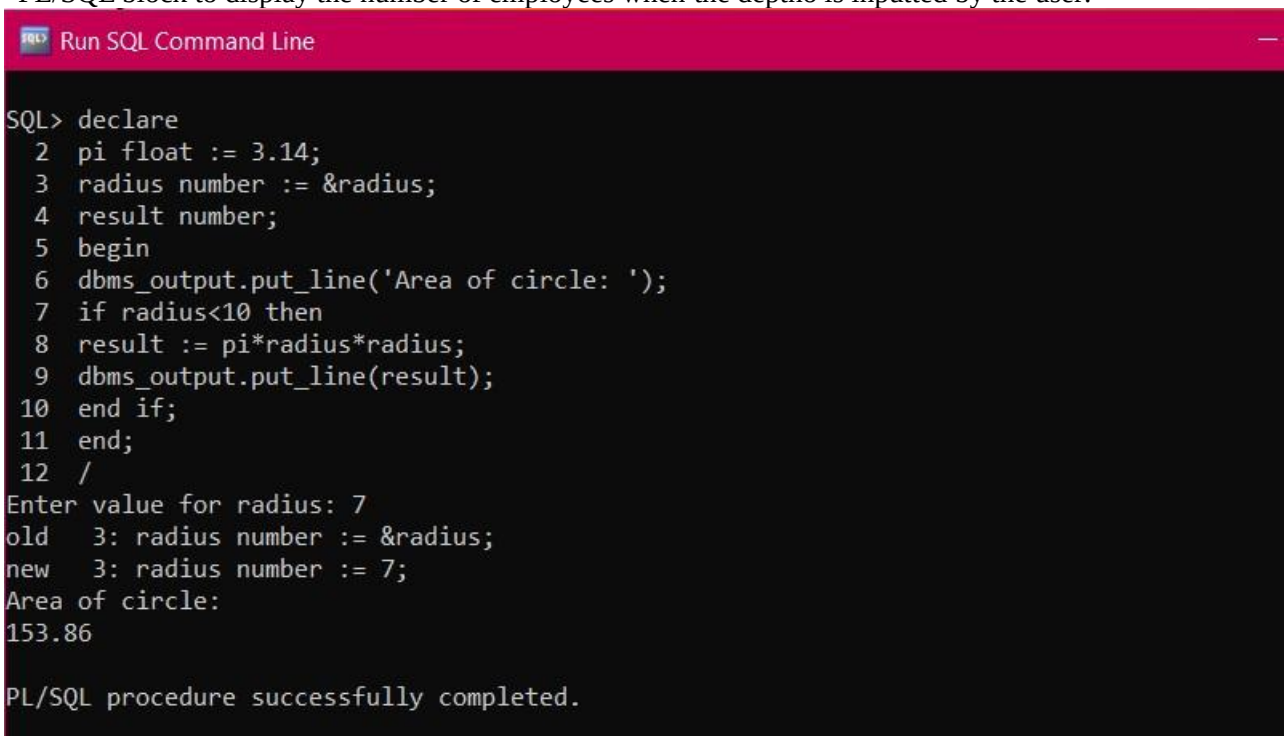
20. Write a PL/SQL block to print the numbers from 1 to 10 using While and For Loop.


Run SQL Command Line
SQL> declare
2 a number;
3 begin
4 for a in 1..10 loop
5 dbms_output.put_line(a);
6 end loop;
7 end;
8 /
1
2
3
4
5
6
7
8
9
10
PL/SQL procedure successfully completed.

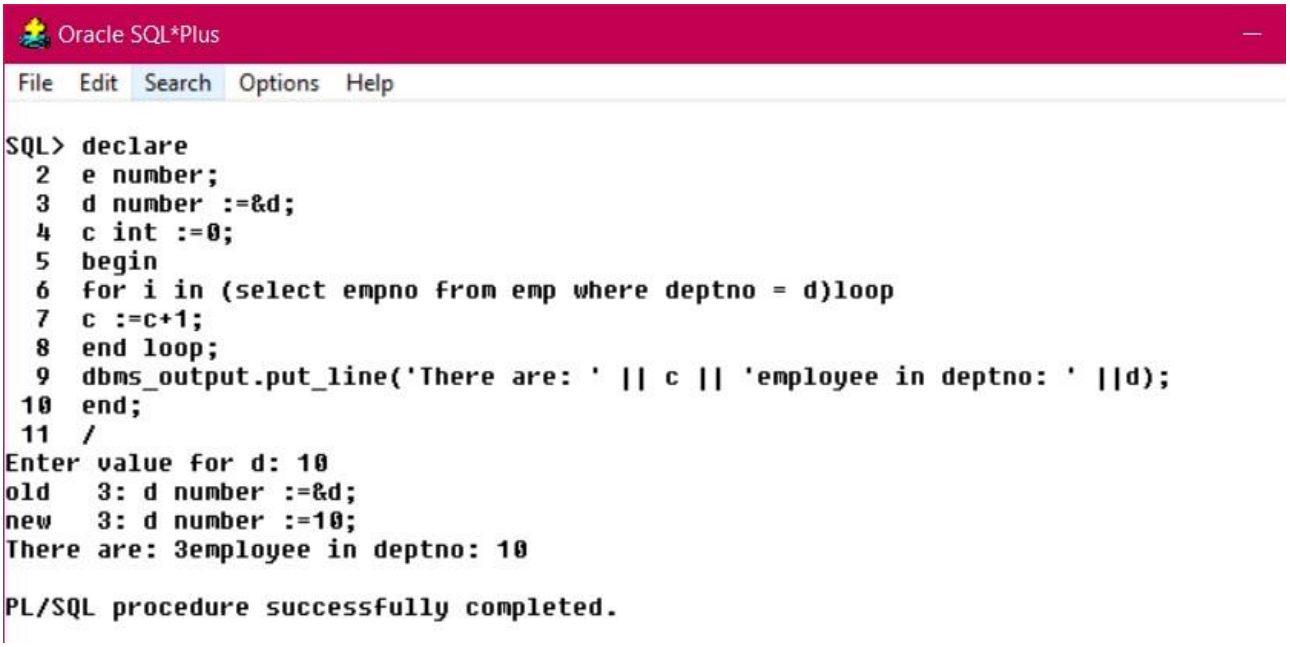
21. Write a PL/SQL block to print the Fibonacci series up to 10.



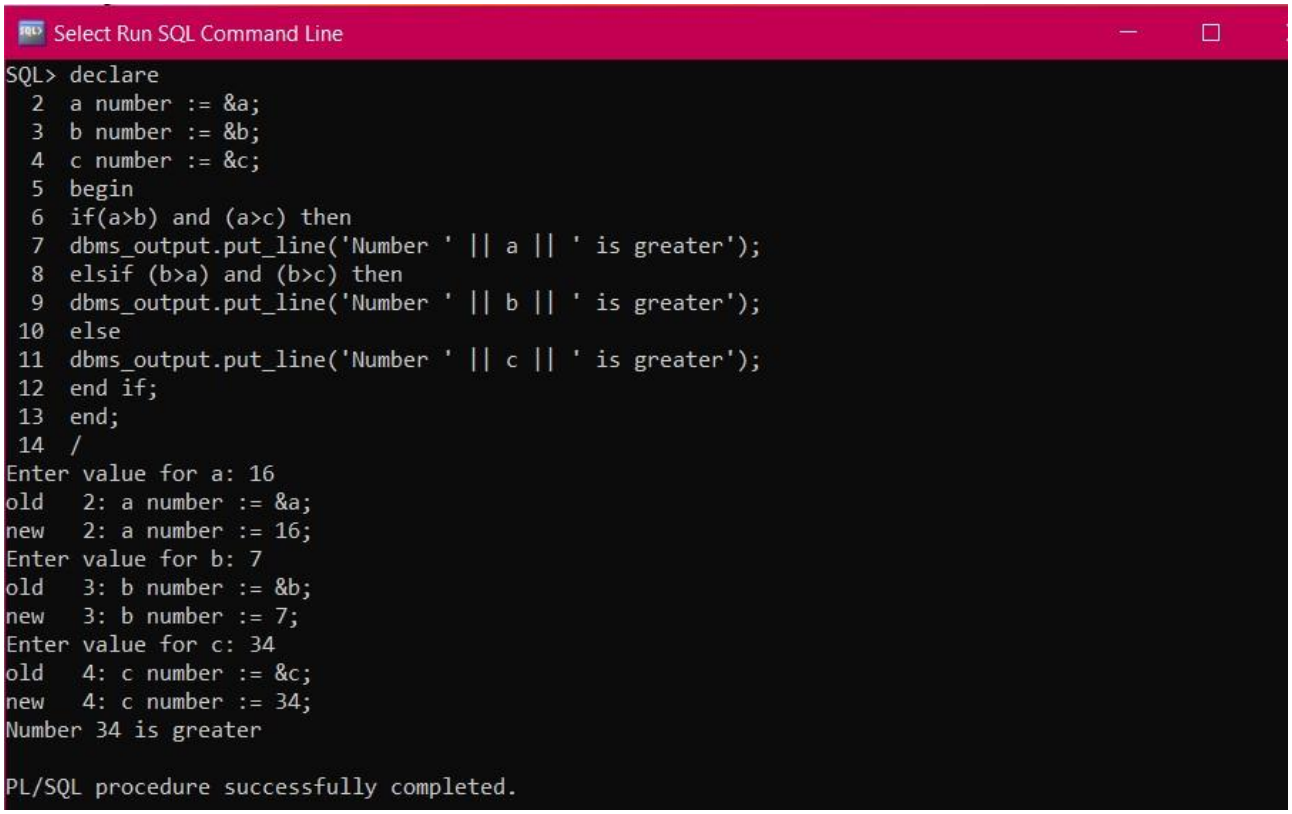
22. Write a PL/SQL block to calculate the area of a circle till radius less than 10. 23.Write a PL/SQL block to display the number of employees when the deptno is inputted by the user.



23. Write a PL/SQL block to display the number of employees when the deptno is inputted by the user.



24. Write a PL/SQL block to print greatest among three numbers.



25. Write a PL/SQL block to display the appropriate day of the week according to the choice made by the user.

Select Run SQL Command Line

```
SQL> declare
  2  dnum number :=&dnum;
  3  begin
  4  case dnum
  5  when 1 then dbms_output.put_line('sunday');
  6  when 2 then dbms_output.put_line('Monday');
  7  when 3 then dbms_output.put_line('Tuesday');
  8  when 4 then dbms_output.put_line('Wednesday');
  9  when 5 then dbms_output.put_line('Thursday');
 10  when 6 then dbms_output.put_line('Friday');
 11  when 7 then dbms_output.put_line('Satarday');
 12  else dbms_output.put_line('Not a number of day in week');
 13  end case;
 14  end;
 15  /
Enter value for dnum: 7
old   2: dnum number :=&dnum;
new   2: dnum number :=7;
Satarday

PL/SQL procedure successfully completed.
```

26. Create a PL/SQL block that has four sections. Each section should output a statement. Use labels and the Goto command to output the section messages in the following order:

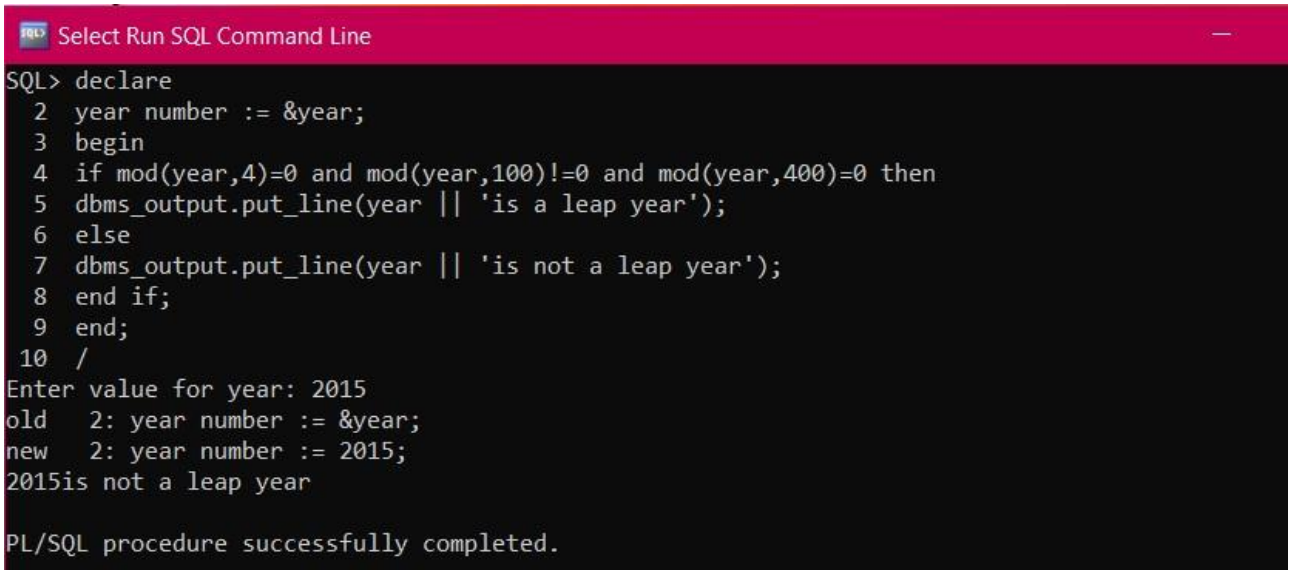
Section 3

Section 2

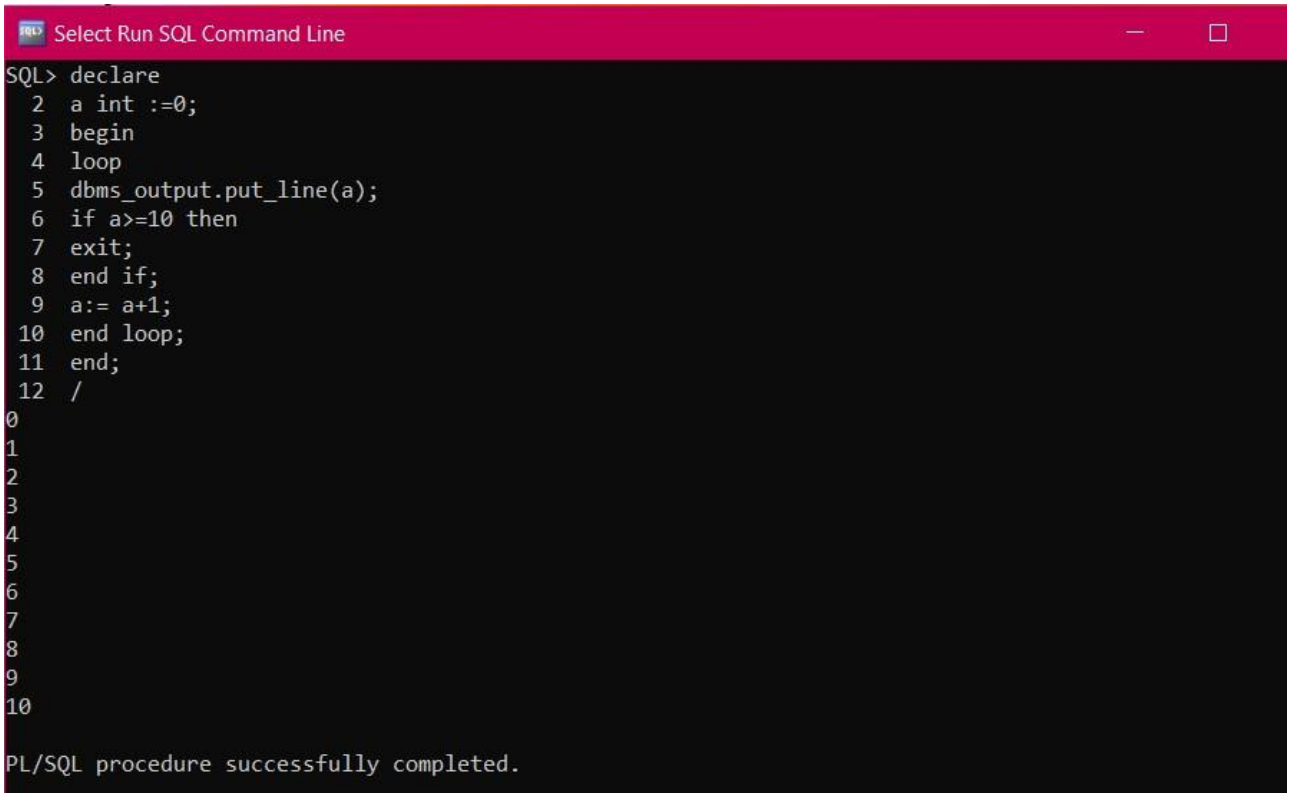
Section 1

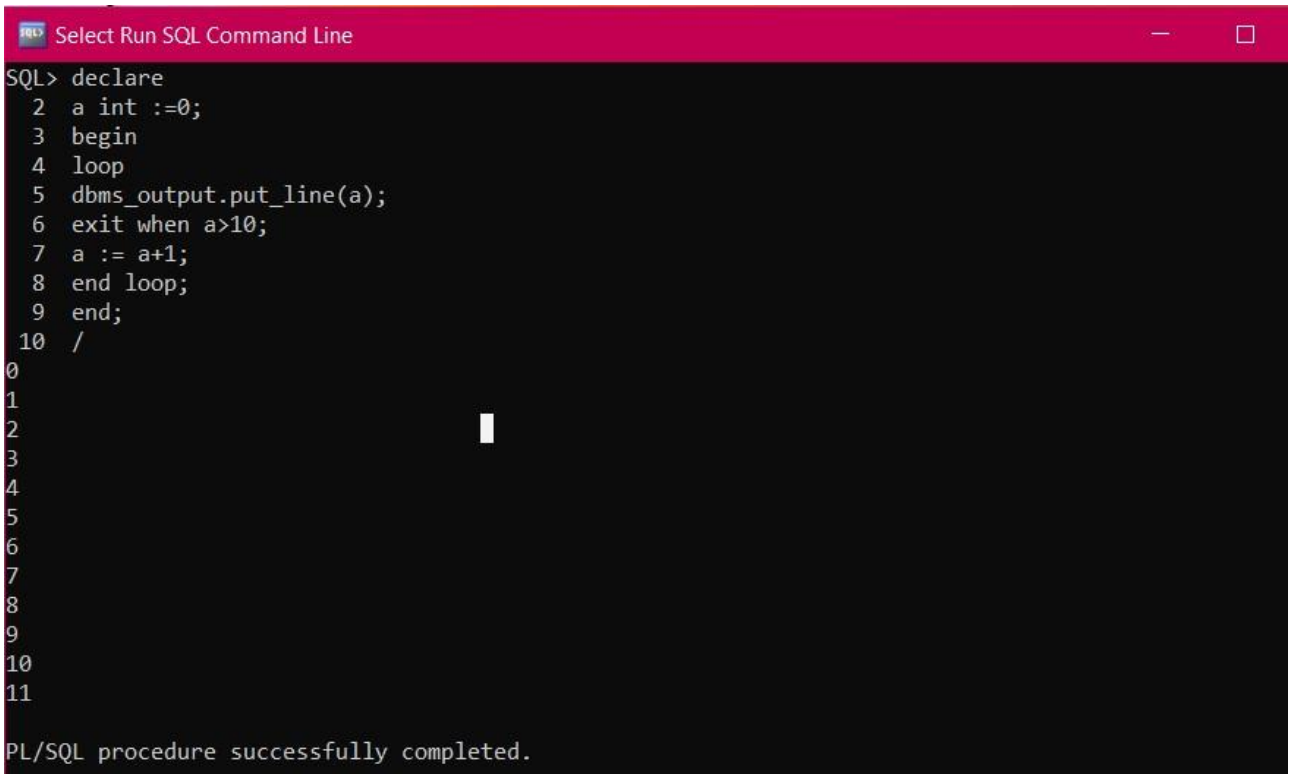
Section 4

27. Write a PL/SQL block to check whether the entered year is a leap year or not.



28. Write a PL/SQL block to display the numbers from 1 to 10 using EXIT and EXIT WHEN statement.


Select Run SQL Command Line
SQL> declare
2 a int :=0;
3 begin
4 loop
5 dbms_output.put_line(a);
6 if a>=10 then
7 exit;
8 end if;
9 a:= a+1;
10 end loop;
11 end;
12 /
0
1
2
3
4
5
6
7
8
9
10
PL/SQL procedure successfully completed.


Select Run SQL Command Line
SQL> declare
2 a int :=0;
3 begin
4 loop
5 dbms_output.put_line(a);
6 exit when a>10;
7 a := a+1;
8 end loop;
9 end;
10 /
0
1
2
3
4
5
6
7
8
9
10
11
PL/SQL procedure successfully completed.

29. Write a PL/SQL block to accept job from EMP table.

Give the following raise in the salary: -

By 9% if job is clerk.

By 8% if job is manager.

By 7% if job is salesman.

Update the salary of the EMP table.

```
SQL> declare
  2  a varchar2(10):='&a';
  3  begin
  4  case a
  5  when 'CLERK'then
  6  update emp
  7  set sal=+(sal*(9/100)) where job=a;
  8  dbms_output.put_line('salary is updated');
  9  when 'MANAGER' then
 10  update emp
 11  set sal=sal+(sal*(8/100)) where job=a;
 12  dbms_output.put_line('salary is updated');
 13  when 'SALESMAN' then
 14  update emp
 15  set sal=sal+(sal*(7/100)) where job=a;
 16  dbms_output.put_line('salary is updated');
 17  end case;
 18  end;
 19  /
Enter value for a: MANAGER
old   2: a varchar2(10):='&a';
new   2: a varchar2(10):='MANAGER';

PL/SQL procedure successfully completed.

SQL>
```

30. Write a PL/SQL block to get the details of marks (rollno, marks1, marks2, grade) out of 100 for marks1 and marks2 respectively.

Display the grade in table marks using if statement as specified below

If stud_percent > 70 then grade is ‘A’

If stud_percent > 60 and <70 then grade is ‘B’ else give grade ‘C’.

```
SQL> declare
  2  roll number := &roll;
  3  m1 number := &m1;
  4  m2 number := &m2;
  5  grade char;
  6  stud_percent number;
  7  begin
  8  stud_percent := ((m1+m2)/200) * 100;
  9  if stud_percent > 70 then
 10  insert into mark values(roll,m1,m2,'A');
 11  elsif stud_percent >60 and stud_percent > 70 then
 12  insert into mark values(roll,m1,m2,'B');
 13  else
 14  insert into mark values(roll,m1,m2,'C');
 15  end if;
 16  end;
 17  /
Enter value for roll: 7
old   2: roll number := &roll;
new   2: roll number := 7;
Enter value for m1: 80
old   3: m1 number := &m1;
new   3: m1 number := 80;
Enter value for m2: 90
old   4: m2 number := &m2;
new   4: m2 number := 90;

PL/SQL procedure successfully completed.
```

31. Write a PL/SQL block to book a ticket for a movie. The tickets are of two type's deluxe rows (D) and Ordinary rows (O).

While booking the ticket the customer may ask 'D' or 'O' and number of ticket. For deluxe the rate is 350 and for ordinary 200.

Find the total amount that the customer will pay and number of tickets (using case statement).

```
SQL> declare
  2  t_type char(1) := '&t_type';
  3  no_tickets number := '&no_tickets';
  4  begin
  5  case t_type
  6  when 'd' then dbms_output.put_line('Total price: ' || no_tickets*350);
  7  when 'o' then dbms_output.put_line('Total price: ' || no_tickets*200);
  8  else
  9  dbms_output.put_line('Not a valid ticket type');
 10  end case;
 11  end;
 12  /
Enter value for t_type: o
old   2: t_type char(1) := '&t_type';
new   2: t_type char(1) := 'o';
Enter value for no_tickets: 2
old   3: no_tickets number := '&no_tickets';
new   3: no_tickets number := '2';

PL/SQL procedure successfully completed.
```