PL/SQL

Q1: Write a PL/SQL program to find the factorial of a given number

```
set  serveroutput on;
declare
fact number:=1;
n number:=&n;
begin
for i in 1..n
loop
 fact:=fact*i;
 end loop;
 dbms_output.put_line('factorial='||fact);
 end;
 /
```

```
SQL> set  serveroutput on;
SQL> declare
  2  fact number:=1;
  3  n number:=&n;
  4  begin
  5  for i in 1..n
  6  loop
  7   fact:=fact*i;
  8   end loop;
  9   dbms_output.put_line('factorial='||fact);
 10   end;
 11   /
Enter value for n: 5
old   3: n number:=&n;
new   3: n number:=5;
factorial=120

PL/SQL procedure successfully completed.
```

Q2: Write a PL/SQL program to check whether the given no is prime or not

```
set  serveroutput on;
declare
i number:=2;
f number:=1;
n number:=&n;
begin
```

```
for i in 2..n/2
loop
if n mod i=0
then
f:=0;
exit;
end if;
end loop;

if f=1
then
dbms_output.put_line('prime');
else
dbms_output.put_line('not prime');
end if;
end;
/
```

```
SQL> set  serveroutput on;
SQL> declare
  2  i number:=2;
  3  f number:=1;
  4  n number:=&n;
  5  begin
  6  for i in 2..n/2
  7  loop
  8  if n mod i=0
  9  then
 10  f:=0;
 11  exit;
 12  end if;
 13  end loop;
 14
 15  if f=1
 16  then
 17  dbms_output.put_line('prime');
 18  else
 19  dbms_output.put_line('not prime');
 20  end if;
 21  end;
 22  /
Enter value for n: 4
old   4: n number:=&n;
new   4: n number:=4;
not prime

PL/SQL procedure successfully completed.
```

Functions

1) Write a PL/SQL program to Check whether a number is Armstrong or not using functions

```
create or replace function arms(n in number)
return number is
    r number;
    s number;
    a number;
    l number;

begin
a:=n;
s:=0;
l:=length(n);
    while a>0
    loop
        r:=mod(a,10);
        s:=s+power(r,l);
        a:=trunc(a/10);
    end loop;
    return s;
    end;
```

```
    /


set serveroutput on;
declare
    n number:=&n;
    s number;
    begin
    s:=arms(n);
    if s=n
    then
        dbms_output.put_line('armstrong number');
    else
        dbms_output.put_line('not armstrong number');
    end if;
end;
/
```

```
SQL> create or replace function arms(n in number)
  2  return number is
  3      r number;
  4      s number;
  5      a number;
  6      l number;
  7
  8  begin
  9  a:=n;
 10  s:=0;
 11  l:=length(n);
 12      while a>0
 13      loop
 14          r:=mod(a,10);
 15          s:=s+power(r,l);
 16          a:=trunc(a/10);
 17      end loop;
 18      return s;
 19      end;
 20      /

Function created.

SQL>
SQL> set serveroutput on;
SQL> declare
  2      n number:=&n;
  3      s number;
  4      begin
  5      s:=arms(n);
  6      if s=n
  7      then
  8          dbms_output.put_line('armstrong number');
  9      else
 10          dbms_output.put_line('not armstrong number');
 11      end if;
 12  end;
 13  /
Enter value for n: 1634
old   2:     n number:=&n;
new   2:     n number:=1634;
armstrong number

PL/SQL procedure successfully completed.
```

2) Create table that contains itemid,item_name & price of several items sold in a grocery shop, Using functions retrieve the item name & price from table when itemid is given as input.

   create or replace function grows(id number)

```
return number as
   s number;
   n number;
   nm grow.itemname%type;
   p grow.price%type;
 begin
 s:=id;
 select itemname,price into nm,p from grow where itemid=s;
 dbms_output.put_line('Item Name'||nm);
 dbms_output.put_line('Price'||p);
 return 0;
 end;
 /

declare
n number:=&n;
p number;
begin
p:=grows(n);
end;
/
```

```
SQL> create or replace function grows(id number)
  2  return number as
  3      s number;
  4      n number;
  5      nm grow.itemname%type;
  6      p grow.price%type;
  7   begin
  8   s:=id;
  9   select itemname,price into nm,p from grow where itemid=s;
 10   dbms_output.put_line('Item Name'||nm);
 11   dbms_output.put_line('Price'||p);
 12   return 0;
 13   end;
 14   /

Function created.

SQL> declare
  2  n number:=&n;
  3  p number;
  4  begin
  5  p:=grows(n);
  6  end;
  7  /
Enter value for n: 2
old   2: n number:=&n;
new   2: n number:=2;
Item Namebox
Price180

PL/SQL procedure successfully completed.
```

3) Write a PL/SQL function called POW that takes two numbers as argument and return the value of the first number raised to the power of the second .

```
create or replace function pow(a number,b number)
return number as
p number;
begin
select power(a,b) into p from dual;
```

```
return p;
end;
/

declare
a number:=&a;
b number:=&b;
begin
dbms_output.put_line('Power is '||pow(a,b));
end;
/
```

```
SQL> create or replace function pow(a number,b number)
  2  return number as
  3  p number;
  4  begin
  5  select power(a,b) into p from dual;
  6  return p;
  7  end;
  8
  9  /

Function created.

SQL> declare
  2  a number:=&a;
  3  b number:=&b;
  4  begin
  5  dbms_output.put_line('Power is '||pow(a,b));
  6  end;
  7  /
Enter value for a: 3
old   2: a number:=&a;
new   2: a number:=3;
Enter value for b: 8
old   3: b number:=&b;
new   3: b number:=8;
Power is 6561

PL/SQL procedure successfully completed.
```