

PL/SQL

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

```
declare
n number;
fact number:=1;
begin
n:=&n;
for i in 1..n
loop
fact:=fact*i;
end loop;
dbms_output.put_line(chr(10)||'Factorial of '||n||' is '||fact);
end;
```

Output

```
Enter value for n: 6
old   5: n:=&n;
new   5: n:=6;

Factorial of 6 is 720

PL/SQL procedure successfully completed.
```

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

```
declare
n number;
c number:=1;
begin
n:=&n;

if n=1
then
```

```

dbms_output.put_line('Not prime');

else
for i in 1..n/2
loop
if mod(n,i)=0
then
c:=c+1;
end if;
end loop;

if c=2
then
dbms_output.put_line('prime');
else
dbms_output.put_line('Not prime');
end if;
end if;

end;

```

Output

```

Enter value for n: 7
old   5: n:=&n;
new   5: n:=7;
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 armstr(x in number)

return number as

z number;

s number:=0;

r number;

l number;

n number;

begin

n:=x;

l:=length(n);

while n>0

loop

r:=mod(n,10);

s:=s+power(r,l);

n:=trunc(n/10);

end loop;

z:=s;

return z;

end;

declare

n number;

c number;

begin

```

n:=&n;
c:=armstr(n);
if c=n
then
dbms_output.put_line('Armstrong');
else
dbms_output.put_line('Not Armstrong');
end if;
end;

```

Output

```

Enter value for n: 153
old 5: n:=&n;
new 5: n:=153;
Armstrong

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 table item(item_id varchar(20),item_name varchar(20),price int);
```

```
insert into item values('p101','bread',30);
```

```
insert into item values('p102','cheese',65);
```

```
insert into item values('p103','egg',13);
insert into item values('p104','soup',99);
insert into item values('p105','oil',110);
```

```
declare
id varchar(10);
begin
id:='&id';
dbms_output.put_line(getdata(id));
end;
```

```
create or replace function getdata(id varchar2)
return number as
p item.item_name%type;
q item.price%type;
begin
select item_name,price into p,q from item where item_id=id;
dbms_output.put_line('Item name  item price');
dbms_output.put_line(p||'      '||q);
return 0;
end;
```

Output

```

Enter value for id: p103
old 4: id:='&id';
new 4: id:='p103';
Item name    item price
egg          13
0
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 powr(a number,b number)

return number as

r number;

begin

r:=power(a,b);

return r;

end;

/

declare

x number;

y number;

begin

x:=&x;

y:=&y;

```
dbms_output.put_line('result'||powr(x,y));  
end;
```

Output

```
Enter value for x: 3  
old 5: x:=&x;  
new 5: x:=3;  
Enter value for y: 3  
old 6: y:=&y;  
new 6: y:=3;  
result=27  
  
PL/SQL procedure successfully completed.
```