**PL/SQL**

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

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

**Functions**

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

**Program**

create or replace function amt(n in number)

return number is

create or replace function amt(n in number)

return number as

aa number;

s number;

r number;

len number;

begin

aa:=n;

len:= length(to\_char(aa));

s:=0;

while aa>0

loop

r:=aa mod 10;

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

aa:=trunc(aa/10);

end loop;

return s;

end;

/

declare

n number:=&n;

s number;

begin

s:=amt(n);

if s=n

then

dbms\_output.put\_line('Its a armstrong number!!');

else

dbms\_output.put\_line('Its not a armstrong number!!');

end if;

end;

/

1. 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.

**program**

declare

id number;

p number;

begin

dbms\_output.put\_line('Enter id:');

id:=&id;

p:=getprice(id);

dbms\_output.put\_line('Price:'||p);

end;

/

create or replace function getprice(id in number)

return number as

p item.price%type;

i number;

begin

i:=id;

select price into p from item where id=i;

return p;

end;

/

1. 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 pow2(a in number,b in number)

return number as

s number;

aa number;

bb number;

begin

aa:=a;

bb:=b;

s:=power(aa,bb);

return s;

end;

/

declare

a number:=&a;

b number:=&b;

s number;

begin

s:=pow2(a,b);

dbms\_output.put\_line('a pow b is:'||s);

end;

/