7:

```
declare
a number:=7;
begin
if(mod(a,2)=0) then
dbms_output.put_line('The number '|| a || ' is even. ');
else
dbms_output.put_line('The number '|| a || ' is odd. ');
end if;
end;
```

8:

```
declare
a number:=5;
i number:=1;
begin
if(a>0) then
for i in 1..a
loop
dbms_output.put_line(i);
end loop;
end if;
end;
```

9:

```
declare
a number :=1;
b number :=-2;
c number :=-15;
d number :=0;
r1 number :=0;
r2 number :=0;
begin
d:=b*b-4*a*c;
if(d>0) then
r1:=(-b+sqrt(d))/(2*a);
r2:=(-b-sqrt(d))/(2*a);
dbms_output.put_line('the roots are ' ||r1|| ' and' ||r2);
elsif(d=0) then
r1:=(-b/(2*a));
dbms_output.put_line('the roots are ' ||r1|| ' and' ||r1);
else
r1:=-b/(2*a);
r2:=sqrt(-d)/(2*a);
dbms_output.put_line('the roots are ' ||r1|| ' and ' ||' +i '|| r2||r1||' -i'||r2);
end if;
end ;
```

10:

```
declare
n number :=17;
i number :=1;
cnt number :=0;
begin
for i in 1..n
loop
if(mod(n,i)=0) then
cnt :=cnt+1;
end if;
end loop;
if(cnt=2) then
dbms_output.put_line(n ||'is prime');
else
dbms_output.put_line(n ||'is not prime');
end if;
end;
```

11:

```
DECLARE
    counter NUMBER;
    k NUMBER;
    a number :=9;
BEGIN
  FOR n IN 2..a LOOP
    counter := 0;
    k := floor(n/2);
    FOR i IN 2..k LOOP
        IF (mod(n, i) = 0 ) THEN
            counter := 1;
        END IF;
    END LOOP;
    IF (counter = 0) THEN
       DBMS_OUTPUT.PUT_LINE(n);
    END IF;
  END LOOP;
END;
```

12:

```
declare
n number:=15;
i number;

begin
for i in 1..10
loop
dbms_output.put_line(n||' x '||i||' = '||n*i);
```

```
end loop;
end;
```

13:

```
declare
    n number;
    m number;
    temp number:=0;
    rem number;
begin
    n:=1771;
    m:=n;
    while n>0
    loop
        rem:=mod(n,10);
        temp:=(temp*10)+rem;
        n:=trunc(n/10);
    end loop;
    if m = temp
    then
        dbms_output.put_line(m||' is a palindrome');
    else
        dbms_output.put_line(m||' is not a palindrome');
    end if;
end;
```

14:

```
DECLARE
    s VARCHAR2(10) := 'abccba';
    l VARCHAR2(20);
    t VARCHAR2(10);
BEGIN
    FOR i IN REVERSE 1..Length(s) LOOP
        l := Substr(s, i, 1);
    t := t||''||l;
    END LOOP;
    IF t = s THEN
      dbms_output.Put_line(t||''||' is palindrome');
    ELSE
      dbms_output.Put_line(t||''||' is not palindrome');
    END IF;
END;
```

15:

```
declare
    pn programmer.pname%type:='Anand';
    db programmer.dob%type;
begin
    select dob into db from programmer where pname= pn;
    dbms_output.put_line('Date of Birth: '||db);
exception
    when no_data_found then
    dbms_output.put_line('No data');
end;
```

16:

```
declare
    cursor s is select * from programmer;
    t s%rowtype;
begin
    open s;
    loop
        fetch s into t;
        exit when s%notfound;
    dbms_output.put_line('Pname: '||t.pname||' '||chr(13)||chr(10) ||' Date of Birth:
'||t.dob||chr(13)||chr(10) );
    end loop;
    close s;
end;
```

17:

```
declare
    pn software.pname%type:='Tulasi';
    cursor s is select * from software where pname = pn;
    t s%rowtype;
begin
    open s;
    loop
        fetch s into t;
        exit when s%notfound;
    dbms_output.put_line('title: '||t.title||chr(13)||chr(10));
    end loop;
    close s;
end;
```

18:

```
declare
    t software.title%type:='Vaccines';
    pn software.pname%type;
begin
    select pname into pn from software where title=t;
    dbms_output.put_line('Name: '||pn);
exception
    when no_data_found then
    dbms_output.put_line('No Data');
end;
```

19:

```
declare
    cursor s is select * from radius1;
    t s%rowtype;
begin
    open s;
    loop
        fetch s into t;
        exit when s%notfound;
        insert into circle1 values(t.radius,3.14*t.radius*t.radius,2*3.14*t.radius);
    end loop;
end;
```

20:

```
create or replace procedure prod(a in number, b in number) as
    c number;
begin
    c:=a*b;
    dbms_output.put_line('Product of '||a||' and '||b||' is '||c);
exception
    when no_data_found then
    dbms_output.put_line('No Data');
end;

declare
    a number:=10;
    b number:=20;
begin
    prod(a,b);
end;
```

21:

```
create or replace procedure db(p in programmer.pname%type) as
    d programmer.dob%type;
begin
    select dob into d from programmer where pname=p;
    dbms_output.put_line('DOB is: '||d);
exception when no_data_found then
    dbms_output.put_line('No Data');
end;

declare
    x programmer.pname%type:='Tulasi';
begin
    db(x);
end;
```

22:

```
create or replace function addi(a in number, b in number) return number as
    c number;
begin
    c:=a+b;
    return (c);
end;

declare
    x number:=10;
    y number:=20;
    z number;
begin
    z:=addi(x,y);
    dbms_output.put_line('Sum of '||x||' and '||y||' is '||z);
end;
```

23:

```
create or replace function getdb(p in programmer.pname%type) return date a
s
    d programmer.dob%type;
begin
    select dob into d from programmer where pname=p;
    return (d);
end;

declare
    x programmer.pname%type:='Altaf';
    r programmer.dob%type;
begin
    r:=getdb(x);
    dbms_output.put_line('Date Of Birth:'||r);
end;
```

24:

```
create or replace procedure getpname(sp studies.splace%type) as
    cursor s is select * from studies where splace = sp;
    t s%rowtype;
begin
    open s;
    loop
    fetch s into t;
    exit when s%notfound;
    dbms_output.put_line(t.pname);
    end loop;
    close s;
end;

-- declare
--     a number:=10;
--     b number:=20;
-- begin
--     prod(a,b);
-- end;
```

25:

```
create or replace function total_dcost_func(p programmer.pname%type) return numbe
r as
    cursor s is select * from software where pname = p;
    t s%rowtype;
    total number;
begin
    total:=0;
    open s;
    loop
    fetch s into t;
    exit when s%notfound;
    total:=total+t.dcost;
    end loop;
    close s;
    return total;
end;

declare
    pn programmer.pname%type:='Revathi';
    res number;
begin
    res:=total_dcost_func(pn);
    dbms_output.put_line('Total development cost is '||res);
end;
```

26:

```
create or replace package my_pack1 is
procedure product_table(a in number);
function product(a in number,b in number) return number;
end;
create or replace package body my_pack1 as
procedure product_table(a in number) as
    i number;
begin
    i:=1;
    while(i<=10)
    loop
        dbms_output.put_line(a||'*'||i||'='||a*i);
        i:=i+1;
    end loop;
```

```
end product_table;

function product(a in number,b in number) return number as
    c number;
begin
    c:=a+b;
    return (c);
end product;
end;
declare
    x number:=10;
    y number:=2;
    z number;
begin
    my_pack1.product_table(x);
    z:=my_pack1.product(x,y);
    dbms_output.put_line('Product of '||x||' and '||y||' is '||z);
end;
```

27:

```
create or replace package pack1 is
procedure p1(p in programmer.pname%type);
function f1(t in software.title%type) return software.pname%type;
end;

create or replace package body pack1 as
procedure p1(p in programmer.pname%type) as
    s programmer.salary%type;
begin
    select salary into s from programmer where pname=p;
    dbms_output.put_line('Salary: '||s);
end p1;

function f1(t in software.title%type) return software.pname%type as
    x software.pname%type;
begin
    select pname into x from software where title=t;
    return (x);
end f1;
end;
```

```
declare
    a programmer.pname%type:='Anand';
    b software.title%type:='Pharachutes';
    r software.pname%type;
begin
    pack1.p1(a);
    r:=pack1.f1(b);
    dbms_output.put_line('PName of given Project is '||r);
end;
```

28:

```
create or replace package pack2 is
procedure p2(x in software.dev_d%type);
function f2(y in studies.pname%type) return studies.splace%type;
end;
create or replace package body pack2 as
procedure p2(x in software.dev_d%type) as
    cursor s is select * from software where dev_d=x;
    t s%rowtype;
begin
    open s;
    loop
        fetch s into t;
        exit when s%notfound;
        dbms_output.put_line(t.title);
    end loop;
    close s;
end p2;

function f2(y in studies.pname%type) return studies.splace%type as
    r studies.splace%type;
begin
    select splace into r from studies where pname=y;
    return (r);
end f2;
end;
declare
    a software.dev_d%type:='Basic';
    b studies.pname%type:='Anand';
    r studies.splace%type;
```

```
begin
    pack2.p2(a);
    r:=pack2.f2(b);
    dbms_output.put_line('Splace is '||r);
end;
```

29:

```
declare
    a programmer.pname%type:='Anand';
    datbirth programmer.dob%type;
    datjoin programmer.doj%type;
    invalid_age_exception exception;
begin
    select dob into datbirth from programmer where pname=a;
    select doj into datjoin from programmer where pname=a;
    if ((datjoin-datbirth)/365)>5 then
        dbms_output.put_line(a||' is eligible for Promotion.');
    else
        raise invalid_age_exception;
    end if;
exception
    when invalid_age_exception then
        dbms_output.put_line('Sorry '||a||' is not eligible for Promotion.');
end;
```

30:

```
create table student1(rollno number(3) primary key,
                        sname varchar2(15),
                        marks1 number(3),
                        marks2 number(3),
                        total number(4));
create or replace trigger auto_cal before insert on student1
for each row
declare
begin
:new.total:=:new.marks1+:new.marks2;
end;
insert into student1 values(1,'Ramesh',44,67,null);
select * from student1;
```

31:

```
create table prog as select pname,salary from programmer;
create table update_prog(pname varchar2(20),old_salary number(7,2),new_sal
ary number(7,2),dt date,tim varchar2(10));
create or replace trigger update_status after update on prog for each row
begin
    insert into update_prog values(:old.pname,:old.salary,:new.salary,sysd
ate,substr(current_timestamp,11,8));
end;
update prog set salary=salary+200 where pname='Anand'
select * from update_prog;
```

32:

```
create table student(roll number(3) primary key,
                        sname varchar2(15),
                        age number(3));
create or replace trigger age_change before insert on student
begin
    update student set age=age+1;
end;
insert into student values(1,'Siva',0);
select * from student;
```