**Q1:**

DECLARE

AREA NUMBER(10,2);

RADIUS NUMBER(10,2);

PI CONSTANT NUMBER(10,2):= 3.14;

BEGIN

RADIUS := &RADIUS;

AREA := PI \* RADIUS \* RADIUS;

DBMS\_OUTPUT.PUT\_LINE('AREA OF CIRCLE IS ' || AREA);

END;

/

**Q2:**

declare

no number;

result number;

begin

no := &no;

dbms\_output.put\_line('Square of number is ' || (no \* no) );

result := 1;

result := no \* no \* no;

dbms\_output.put\_line('Cube of number is ' || result );

result := 1;

result := no \* 2;

dbms\_output.put\_line('Double of number is ' || result );

end;

**Q3:**

declare

    a number;

    b number;

    temp number;

begin

    a:=5;

    b:=10;

    dbms\_output.put\_line('before swapping:');

    dbms\_output.put\_line('a='||a||' b='||b);

    temp:=a;

    a:=b;

    b:=temp;

    dbms\_output.put\_line('after swapping:');

    dbms\_output.put\_line('a='||a||' b='||b);

end;

/

**Q4:**

declare  
    ename varchar2(15);  
    basic number;  
    da number;  
    hra number;  
    pf number;  
    netsalary number;  
begin  
    ename:='&ename';  
    basic:=&basic;  
    da:=basic \* (30/100);  
    hra:=basic \* (10/100);  
        if (basic < 8000)  
    then  
        pf:=basic \* (8/100);  
        elsif (basic >= 8000 and basic <= 16000)  
    then  
        pf:=basic \* (10/100);  
             
end if;  
        netsalary:=basic + da + hra -pf;  
  
dbms\_output.put\_line('Employee name : ' || ename);  
dbms\_output.put\_line('Providend Fund : ' || pf);  
dbms\_output.put\_line('Net salary : ' || netsalary);  
end;

**Q5:**

DECLARE

CURSOR emp\_cur (emp\_job\_nm VARCHAR2, job\_max\_sal NUMBER, dt\_of\_hire DATE DEFAULT '31-DEC-99' ) IS

SELECT last\_name, first\_name, (salary - job\_max\_sal) overpayment

FROM employees

WHERE job\_id = emp\_job\_nm

AND salary > job\_max\_sal

AND hire\_date > dt\_of\_hire

ORDER BY salary;

PROCEDURE emp\_excesspaid IS

last\_name\_ employees.last\_name%TYPE;

first\_name\_ employees.first\_name%TYPE;

paid\_excess employees.salary%TYPE;

BEGIN

LOOP

FETCH emp\_cur INTO last\_name\_, first\_name\_, paid\_excess;

EXIT WHEN emp\_cur%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(last\_name\_ || ', ' || first\_name\_ ||

' (by ' || paid\_excess || ')');

END LOOP;

END emp\_excesspaid;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('---------------------------------');

DBMS\_OUTPUT.PUT\_LINE('Extra Salary paid to Programmers:');

DBMS\_OUTPUT.PUT\_LINE('---------------------------------');

OPEN emp\_cur('IT\_PROG', 6000);

emp\_excesspaid;

CLOSE emp\_cur;

DBMS\_OUTPUT.PUT\_LINE('-----------------------------------');

DBMS\_OUTPUT.PUT\_LINE('Extra Salary paid to Stock Manager hired after 2005:');

DBMS\_OUTPUT.PUT\_LINE('----------------------------------------------------');

OPEN emp\_cur('ST\_MAN', 5000,'31-DEC-05');

emp\_excesspaid;

CLOSE emp\_cur;

END;

/

**Q6:**  
declare

n number:=&n;

begin

if mod(n,2)=0

then

dbms\_output.put\_line('number is even');

else

dbms\_output.put\_line('number is odd');

end if;

end;

/

**Q7:**

declare

a number:=10;

b number:=12;

c number:=5;

begin

dbms\_output.put\_line('a='||a||' b='||b||' c='||c);

if a>b AND a>c

then

dbms\_output.put\_line('a is greatest');

else

if b>a AND b>c

then

dbms\_output.put\_line('b is greatest');

else

dbms\_output.put\_line('c is greatest');

end if;

end if;

end;

/

**Q8:**

DECLARE

st\_month NUMBER(2) := 1;

no\_o\_emp NUMBER(3);

BEGIN

dbms\_output.Put\_line(Rpad('Month No', 20)

||Rpad('Month Name', 20)

|| 'Number of Employees');

dbms\_output.Put\_line('-------------------------------------------------------------');

FOR month IN 1 .. 12 LOOP

SELECT Count(\*)

INTO no\_o\_emp

FROM employees

WHERE To\_char(hire\_date, 'mm') = month;

dbms\_output.Put\_line(Rpad(To\_char(month, '00'), 20)

||Rpad(To\_char(To\_date(month, 'MM'), 'MONTH'), 20)

|| To\_char(no\_o\_emp, '999'));

END LOOP;

END;

/

**Q9:**

DECLARE

grd CHAR(1);

BEGIN

-- Accept value for grade

grd := '&new\_grd';

CASE grd

WHEN 'A' THEN dbms\_output.Put\_line('Your Grade is: Outstanding');

WHEN 'B' THEN dbms\_output.Put\_line('Your Grade is: Excellent');

WHEN 'C' THEN dbms\_output.Put\_line('Your Grade is: Very Good');

WHEN 'D' THEN dbms\_output. Put\_line('Your Grade is: Average');

WHEN 'F' THEN dbms\_output.Put\_line('Your Grade is: Poor');

ELSE dbms\_output.Put\_line('No such grade in the list.');

END CASE;

END;

/

**Assignment-2**

// C program to find the number of days in a given month

#include<stdio.h>

int main()

{

//fill the code

int year, month;

scanf("%d %d",&month,&year);

if(month == 1 || month == 3 || month == 5 || month == 7 || month == 8 || month == 10 || month == 12)

printf("Number of days is 31");

else if((month == 2) && ((year%400==0) || (year%4==0 && year%100!=0)))

{

printf("Number of days is 29");

}

else if(month == 2)

{

printf("Number of days is 28");

}

else

printf("Number of days is 30");

return 0;

}

**Q2:**

declare

n number;

i number;

rev number:=0;

r number;

begin

n:=&n;

while n>0

loop

r:=mod(n,10);

rev:=(rev\*10)+r;

n:=trunc(n/10);

end loop;

dbms\_output.put\_line('reverse is '||rev);

end;

/

**Q3:**

declare

n number:=5;

i number;

j number;

begin

for i in 1..n

loop

for j in 1..i

loop

dbms\_output.put('\*');

end loop;

dbms\_output.new\_line;

end loop;

end;

/

**Q4:**

declare

area number(5,2);

radius number(1):=3;

pi constant number(3,2):=3.14;

begin

while radius<=7

loop

area:=pi\*radius\*radius;

insert into areas values (radius,area);

radius:=radius+1;

end loop;

end;

**Output:**

PL/SQL procedure successfully completed.

SQL> select \* from areas;

RADIUS AREA

---------- ----------

3 28.26

4 50.24

5 78.5

6 113.04

7 153.86