

1. Fibonacci series

```
--Fibonacci Series
declare
    a number := 0; --starting number
    b number :=1; -- second number
    c number; -- to store a addition of a,b values
    n number := :n; -- To Give the range for fibonacci
begin
    DBMS_OUTPUT.PUT_LINE('First 10 Fibonacci Numbers');
    DBMS_OUTPUT.PUT_LINE(a);
    DBMS_OUTPUT.PUT_LINE(b);
    for i in 3..n
    loop
        c:=a+b;
        DBMS_OUTPUT.PUT_LINE(c);
        a := b;
        b := c;
    end loop;
end;
```

First 10 Fibonacci Numbers

0
1
1
2
3
5
8
13
21
34

2. Palindrome number or not

```
--polindrome number or not
declare
    num number := :n;
    rev_num number;
    temp number:=0;
    rem number;
begin
    rev_num := num; --121, 12, 1, 0
    while num>0
    loop
        rem:=mod(num,10); -- mod(121,10)=1, mod(21,10) = 2, mod(1,10)= 1
        temp:=(temp*10)+rem; -- 0*10+1=1, 1*10+2 = 12, 12*10+1 = 121
        num:=trunc(num/10); -- 121/10 =12, 12/10=1, 1/10 = 0
    end loop;
    if rev_num = temp then
        dbms_output.put_line(temp||' is a polindrome');
    else
        dbms_output.put_line(temp||' is not a polindrome');
    end if;
end;
```

Script Output x

Task completed in 0.04 seconds

121 is a polindrome

PL/SQL procedure successfully completed.

3. Reverse of an number

```
-- Reverse a number
declare
    num number := :n;
    rev_num number := 0;
begin
    while num>0
    loop
        rev_num := (rev_num *10) + mod(num,10);
        num := trunc(num/10);
    end loop;
    dbms_output.put_line('Reverse of the number is '||rev_num);
end;
```

Script Output x

Task completed in 0.141 seconds

Reverse of the number is 54321

PL/SQL procedure successfully completed.

4. Reverse of a string

```

--Reverse a string
declare
    str varchar2(300) :=:s;
    rev_str varchar2(300);
begin
    for i in reverse 1..length(str)
    loop
        rev_str :=rev_str || substr(str,i,1);
    end loop;
    dbms_output.put_line('Reversed string is: '||rev_str);
end;

```

Script Output x
 Task completed in 0.059 seconds
 Reversed string is: hsejaR

PL/SQL procedure successfully completed.

5. Armstrong Number

```

-- Armstrong Number
declare
    n number := :n; --153
    s number :=0;
    r number ;
    len number; --3
    m number; -- 153
begin
    m:= n; --153
    len := length(to_char(n)); --3

    while n>0 --153, 15, 1, 0 = false loop over
    loop
        r := mod(n,10); -- mod(153,10) = 3 & mod(15,10) = 5, mod(1,10) = 1
        s := s + power(r,len); -- 0+ pow(3,3)=27, 27 +pow(5,3)=152, 152+pow(1,3)= 153 s = 153
        n := trunc(n / 10); --153/10=15, 15/10=1 , 1/10 = 0
    end loop;
    if m = s -- m = 153 and s = 153
    then
        dbms_output.put_line(m ||' is a armstrong number');
    else
        dbms_output.put_line(m ||' is not a armstrong number');
    end if;
end;

```

Script Output x
 Task completed in 0.077 seconds
 153 is a armstrong number

PL/SQL procedure successfully completed.

```
Script Output x
Task completed in 0.053 seconds
1535 is not a armstrong number

PL/SQL procedure successfully completed.
```

6. Sum of digits is equal to given number, here number = 25

```
1 set serveroutput on;
2 -- Sum of digits is equal to gieven number, here number = 25
3 declare
4     n number;
5     m number;
6     s number := 0;
7 begin
8     for i in 1..999
9     loop
10         n := i;
11         while n>0
12         loop
13             m := mod(n,10);
14             s := s + m;
15             n := trunc(n/10);
16         end loop;
17         if s = 25 then
18             dbms_output.put_line(i);
19         end if;
20         s:=0;
21     end loop;
22 end;
```

```
Script Output x
Task completed in 0.034 seconds
799
889
898
979
988
997

PL/SQL procedure successfully completed.
```

7. Sum of digits in a given number

```

3  --sum of digits in a given number
4  declare
5      n number:= :num;
6      total_sum number := 0;
7      rem number;
8      num number;
9  begin
10     num := n;
11     while num <> 0
12     loop
13         rem := mod(num,10);
14         total_sum := total_sum + rem ;
15         num := trunc(num/10);
16     end loop;
17     dbms_output.put_line('the sum of digits in '||n||' is '||total_sum);
18 end;

```

Script Output x
Task completed in 0.06 seconds
the sum of digits in 1235897 is 35

PL/SQL procedure successfully completed.

8. Perfect number or not

```

-- Perfect Number or not
declare
    num number:= :n;
    sum_of_divisors number :=0;
begin
    for i in 1..num/2
    loop
        if num mod i = 0 then
            sum_of_divisors := sum_of_divisors + i;
        end if;
    end loop;
    if sum_of_divisors = num then
        dbms_output.put_line(num|| ' is a perfect number');
    else
        dbms_output.put_line(num|| ' is not a perfect number');
    end if;
end;

```

Script Output x
Task completed in 0.069 seconds
496 is a perfect number

PL/SQL procedure successfully completed.

9. First 20 even numbers

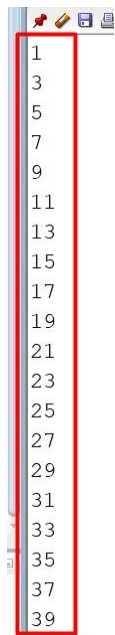
```
-- first 20 even numbers
declare
    num number := 0;
    counter number := 0;
begin
    while counter < 20
    loop
        num := num + 2;
        dbms_output.put_line(num);
        counter := counter + 1;
    end loop;
end;
```

PL/SQL procedure successfully completed.

2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40

10. First 20 odd numbers

```
-- first 20 odd numbers
declare
    num number := 1;
    counter number := 0;
begin
    while counter < 20
    loop
        if num mod 2 <> 0 then
            dbms_output.put_line(num);
            counter := counter + 1;
        end if;
        num := num + 1;
    end loop;
end;
```



11. Retrieve all employee details using for loop

```
begin
  dbms_output.put_line('emp_no' || chr(9) || 'Emp_name' || chr(9) || 'Job Nmae' || chr(9)
  || 'Manager_Id' || chr(9) || 'Salary' || chr(9) || 'Commision' || chr(9) || 'deptno');
  for i in (select empno, ename, job, mgr, hiredate, sal, comm, deptno from emp)
  loop

    dbms_output.put_line(i.empno || chr(9) ||
                          i.ename || chr(9) ||
                          i.job || chr(9) || chr(9) || chr(9) ||
                          i.mgr || chr(9) ||
                          i.sal || chr(9) || chr(9) || chr(9) ||
                          i.comm || chr(9) ||
                          i.deptno || chr(9));

  end loop;
end;
/
```

emp_no	Emp_name	Job Nmae	Manager_Id	Salary	Commision	deptno
7369	SMITH	CLERK	7902	26000		20
7499	ALLEN	SALESMAN	7698	25000	300	30
7521	WARD	SALESMAN	7698	1250	500	30
7566	JONES	MANAGER	7839	2975		20
7654	MARTIN	SALESMAN	7698	1250	1400	30
7698	BLAKE	MANAGER	7839	2850		30
7782	CLARK	MANAGER	7839	2450		10
7788	SCOTT	ANALYST	7566	3000		20
7839	KING	PRESIDENT		5000		10
7844	TURNER	SALESMAN	7698	1500	0	30
7876	ADAMS	CLERK	7788	1100		20
7900	JAMES	CLERK	7698	950		30
7902	FORD	ANALYST	7566	3000		20
7934	MILLER	CLERK	7782	1300		10

12. Triangle pattern

```

Worksheet Query Builder
-- Triangle Pattern
declare
    num number := :n;
    i number ;
    j number ;
begin
    for i in 1..num
    loop
        for j in 1..i
        loop
            dbms_output.put('*');
        end loop;
        dbms_output.new_line;
    end loop;
end;

```

Script Output x Task completed in 0.211 seconds

```

*
**
***
****
*****

```


13. Triangle pattern in reverse

```
-- Triangle Pattern in reverse
declare
    num number := :n;
    i number ;
    j number ;
begin
    for i in reverse 1..num
    loop
        for j in 1..i
        loop
            dbms_output.put('*');
        end loop;
        dbms_output.new_line;
    end loop;
end;
```

Script Output x

Task completed in 0.074 seconds

```
*****
****
***
**
*
```