PL*SQL

Exercise 4

1. Write a program containing a loop that iterates from 1 to 1000 using a variable *I*, which is incremented each time around the loop. The program should output the value of *I* every hundred iterations (i.e., the output should be 100, 200, etc). Display the output on the screen using dbms_output.put_line.

```
declare
v_num number;
begin
for i in 1..1000
loop
if MOD(i,100)=0 then
dbms_output.put_line(i);
end if;
end loop;
end;
```

2. Write a program that examines all the numbers from 1 to 999, displaying all those for which the sum of the cubes of the digits equal the number itself. Display the output on the screen using dbms_output.put_line.

```
declare

v_num number(5,0);

v_cube number(5,0);

j number(5,0);

i number(5,0);


begin


for i in 1..999

loop

j:=i;
```

```
v_cube:=0;


while (j<>0)

loop


v_num:=MOD(j,10);


j:=trunc(j/10);

v_cube:=v_cube+power(v_num,3);

end loop;


if v_cube=i  then

dbms_output.put_line(v_cube);

end if;

end loop;
end;
```

3. Write a PL*SQL block that reads in a minimum and maximum value for a radius, along with an increment factor, and generates a series of radii by repeatedly adding the increment to the minimum until the maximum is reached. For each value of the radius, compute and display the circumference, area, and volume of the sphere. (Be sure to include both the maximum and the minimum values.). Validate each of the input values to be sure they are positive. If the minimum is typed in place of the maximum, swap the values within the program, and continue execution. Display the results on the screen using dbms_output.put_line.

```
declare
v_min number(2,0);
v_max number(2,0);
v_ccm number(5,2);
v_ar number(5,2);
v_vol number(5,2);

begin
v_min:=:min;
v_max:=:max;
```

```
for i in v_min..v_max
loop
v_ccm:=2*3.14*i;
v_ar:=3.14*power(i,2);
v_vol:=(3.14*power(i,3)*4)/3;

dbms_output.put_line('radius is'||i||'circumference is'||v_ccm||'area
is'||v_ar||'volume is'||v_vol);

end loop;
end;
```

4. Allow any positive integer to be typed in. The program should count how many times the number has to be doubled before it reaches 1 million. Display the results on the screen using dbms_output.put_line.

```
declare
v_min number(2,0);
v_max number(2,0);
v_ccm number(5,2);
v_ar number(5,2);
v_vol number(5,2);
v_temp number(5,2);
begin
v_min:=:min;
v_max:=:max;


if v_min > v_max then
v_temp:=v_max;
v_max:=v_min;
v_min:=v_temp;
end if;


for i in v_min..v_max
loop
v_ccm:=2*3.14*i;
v_ar:=3.14*power(i,2);
v_vol:=(3.14*power(i,3)*4)/3;

dbms_output.put_line('radius is '||i||' circumference is '||v_ccm||' area is
'||v_ar||' volume is '||v_vol);
end loop;
end;
```

5. A *palindrome* is a word that is spelled the same forward and backward, such as *level*, *radar*, etc. Write a program to read in a five letter word from the user and determine whether it is a palindrome. Display the results on the screen using dbms_output.put_line.

```
declare
v_str1 varchar2(20);
v_str2 varchar2(20);
v_str3 varchar2(20);
flag number:=1;
begin
v_str1:=:v_str1;
for i in 1..2
loop

   v_str2:=substr(v_str1,i,1);
   v_str3:=substr(v_str1,6-i,1);
   if v_str2<>v_str3 then
   flag:=0;
   end if;
end loop;
if flag=0 then
dbms_output.put_line('String Is Not Pallindrome');
else
dbms_output.put_line('String Is Pallindrome');
end if;
end;
```

6. Modify the above program to accept a variable length word. This requires determining how many characters are read in.

```
declare
v_str1 varchar2(20);
v_str2 varchar2(20);
v_str3 varchar2(20);
v_len number(2,0);
flag number:=1;
begin
v_str1:=:v_str1;
v_len:=length(v_str1);
for i in 1..v_len/2
loop

   v_str2:=substr(v_str1,i,1);
   v_str3:=substr(v_str1,(v_len+1)-i,1);
   if v_str2<>v_str3 then
   flag:=0;
   end if;
end loop;
if flag=0 then
dbms_output.put_line('String Is Not Pallindrome');
else
dbms_output.put_line('String Is Pallindrome');
end if;
end;
```

7. Write a program to read in a number and print it out digit by digit, as a series of words. For example, the number 523 would be printed as "five two three". Use decode function within a for loop. Display the results on the screen using dbms_output.put_line.

```
declare
v_n number;
v_n1 number;
v_digit varchar2(20);
v_digit1 varchar2(20);
begin
v_n:=:v_n;
while (v_n<>0)
loop
 select decode(mod(v_n,10),0,'zero',
                  1,'one',
                  2,'two',
                  3,'three',
                  4,'four',
                  5,'five',
                  6,'six',
                  7,'seven',
                  8,'eight',
                  9,'nine')into v_digit1 from dual;
v_n:=trunc(v_n/10);
v_digit:=v_digit1||' '||v_digit;
end loop;
dbms_output.put_line(v_digit);
end;
```