Write a PL/SQL block to find the largest of two numbers. Take the numbers as input during runtime.

```
declare
    a number := &a;
    b number := &b;
begin
    if (a>b)
       then
        dbms_output.put_line(a);
       else
        dbms_output.put_line(b);
    end if;
end;
```

Write a PL/SQL block to find the largest among three numbers. Take the no.s as input during runtime.

```
declare
    a number := &a;
    b number := &b;
    c number := &c;
begin
    if (a>b and a>c)
      then
        dbms_output.put_line(a);
    else if (b>a and b>c)
      then
        dbms_output.put_line(b);
      else
        dbms_output.put_line(c);
      end if;
end if;
end;
```

HAP Write a PL/SQL block to print 1...10.

```
declare
    i number;
begin
    for i in 1..10
      loop
      dbms_output.put_line(i);
      end loop;
end;
```

Write a PL/SQL block to print 1..10 in reverse.

```
declare
  a number := 10;
begin
    while(a>0)
      loop
        dbms_output.put_line(a);
        a := a-1;
      end loop;
end;
```

Write a PL/SQL block to print 1..10 using while loop.

```
declare
    a number := 1;
begin
    while(a<=10)
      loop
        dbms_output.put_line(a);
        a := a+1;
      end loop;
end;
```

Write a PL/SQL block to find whether the entered year is leap year or not.

```
declare
    l number := &l;
begin
    if (mod (l, 4) = 0 and mod (l, 100) <> 0 or
       mod (l, 400) = 0)
    then
    dbms_output.put_line ("leap year");
    else
     dbms_output.put_line('not leap year');
    end if;
end;
```

Write a PL/SQL block to find whether the entered number is an armstrong number or not.

```
declare
      n number;
      m number;
      s number := 0;
      i number;
      p number := 153;
begin
    n:=p;
    while (n != 0)
     loop
         m := mod (n, 10);
         s := s+ ( m * m * m);
         n := floor (n/10);
    end loop;
    if (s=p
```

```
then dbms_output.put_line('armstrong');
  else
    dbms_output.put_line ('not armstrong');
  end if;
end;
```

Write a PL/SQL block to display first 10 odd numbers.

```
declare
  i number;
begin
  for   i in 1..20
    loop
      if(mod(i,2)<>0)
      then
      dbms_output.put_line(i);
      end if;
    end loop;
end;
```

Write a PL/SQL block to display the first 10 no.s that are divisible by both 3 and 5.

```
declare
     i number := 1;
     c number := 1;
begin
    while(i<100)
      loop
        if(mod(i,2)<>0 and c<=10)
        then
        dbms_output.put_line(i);
        c:= c+1;
       end if;
         i:=i+1;
    end loop;
end;
```

Write a PL/SQL block to accept the marks of 3 subjects from student and calculate the average. If the average is less than 50%, print fail. If its 50-60% print second, if its 60-75% print first and if its average is 75% and above print distinction.

```
declare
    m number := &m;
    e number := &e;
    s number := &s;
    a number;
begin
    a := (m+e+s)/3;
    if (a >= 75)
    then
      dbms_output.put_line ('distinction');
    else if (a>60 and a<75)
      then
        dbms_output.put_line ('first');
      else if (a>50 and a<60)
        then
          dbms_output.put_line ('second');
          else if (a<50)
          then
            dbms_output.put_line ('fail');
          end if;
        end if;
      end if;
    end if;
end;
```

Write a PL/SQL block to reverse a number.

```
declare
  a number := 123;
  b number;
  r number;
begin
  while (a>0)
    loop
      b := mod(a,10);
      r := (r*10) + b;
    dbms_output.put_line(b);
    end loop;
end;
```