Given the following Kotlin classes:

```
class LL {                                 class LLN (w:String,n:LLN?) {

 var head:LLN? = null                       var word = w
}                                           var next = n
                                           }
```

Write the methods printtail (), one in the LL class and one in the LLN class.

Given a linked list variable,

```
var l:LL?
```

the method `l!!.printtail()` should print the string in the tail node, the node furthest from the head. The LL printtail() method should call the method in the LLN class which should operate recursively, without loops.

LL:

```
fun printtail () {

  head?.printtail()
}
```

LLN:

```
 fun printtail () {

  if (next==null)
   println (word)
  else
   next!!.printtail()
 }
```