

Exercice 1 : Triangle de Pascal

La fonction pascal s'écrit comme suit :

```
def pascal(n: Int): List[List[Int]] = {
  def shiftLeft(xs: List[Int]): List[Int] = xs::List(0);
  def shiftRight(xs: List[Int]): List[Int] = 0::xs;
  def iter(i: Int, acc: List[List[Int]]): List[List[Int]] =
    if (i == n)
      acc
    else {
      val xs = acc.head;
      val ys = shiftLeft(xs) zip shiftRight(xs) map (p => p._1 + p._2);
      iter(i + 1, ys::acc)
    }
  if (n < 1) List(()) else iter(1, List(List(1)))
}
```

Exercice 2 : Preuve par induction structurelle

Pour démontrer l'égalité (pour mémoire xs a le type `List[A]`)

```
replaceHead(xs.last, replaceLast(xs.head, xs))  
= replaceLast(xs.head, replaceHead(xs.last, xs))
```

on effectue une preuve par induction structurelle sur xs .

Pour cela on traite tout d'abord le cas de base pour chaque côté de l'égalité avant de passer à l'étape d'induction.

1. Cas de base : $xs = y::Nil$

Pour le côté gauche on a :

```
replaceHead((y::Nil).last, replaceLast((y::Nil).head, y::Nil))  
= (selon définitions de ::.last et ::.head)  
  replaceHead(y, replaceLast(y, y::Nil))  
= (selon 2ième clause de replaceLast)  
  replaceHead(y, y::Nil)  
= (selon 2ième clause de replaceHead)  
  y::Nil
```

Pour le côté droit on a :

```
replaceLast((y::Nil).head, replaceHead((y::Nil).last, y::Nil))  
= (selon définitions de ::.head et ::.last)  
  replaceLast(y, replaceHead(y, y::Nil))  
= (selon 2ième clause de replaceHead)  
  replaceLast(y, y::Nil)  
= (selon 2ième clause de replaceLast)  
  y::Nil
```

2. Etape d'induction : $xs = y::ys$

Pour le côté gauche on a :

```
replaceHead((y::ys).last, replaceLast((y::ys).head, y::ys))  
= (selon définitions de ::.last et ::.head)  
  replaceHead(ys.last, replaceLast(y, y::ys))  
= (selon 3ième clause de replaceLast)  
  replaceHead(ys.last, y::replaceLast(y, ys))  
= (selon 2ième clause de replaceHead)  
  ys.last::replaceLast(y, ys)
```

Pour le côté droit on a :

```
replaceLast((y::ys).head, replaceHead((y::ys).last, y::ys))  
= (selon définitions de ::.head et ::.last)  
  replaceLast(y, replaceHead(ys.last, y::ys))  
= (selon 2ième clause de replaceHead)  
  replaceLast(y, ys.last::ys)  
= (selon 3ième clause de replaceLast)  
  ys.last::replaceLast(y, ys)
```

QED