(1) Implementar a função `retornaUltimo`

```
> retornaUltimo [1,2,3]
3
```

(2) Implementar a função `pegaPosicao`

```
> pegaPosicao 3 [10,2,3,4,5]
3
```

(3) Implementar a função `pega`:

```
> pega 3 [1,2,3,4,5]
[1,2,3]
```

(4) Implementar a função `retira`:

```
> retira 3 [1,2,3,4,5]
[4,5]
```

(5) Implementar uma função que cacula a média dos elementos de uma lista

(6) Implementar a função `pegaMaiores`

```
> pegaMaiores 3 [10,2,3,4,5]
[10,4,5]
```

(7) Implementar a função `contaMaiores`

```
> pegaMaiores 3 [10,2,3,4,5]
3
```

(8) Implementar a função `contatena`

```
> concatena "abc" "123"
"abc123"

> concatena [1,2] [3,4,5]
[1,2,3,4,5]
```

(9) Implementar a função `intercala`

```
> intercala [1,2,3,4] [10,20,30]
[1,10,2,20,3,30,4]
```

(10) Implementar a função `compress` que elimina elementos consecutivos

```
> compress "aaaabccaadeeee"
"abcade"
```

(11) Implementar a função `pack` que coloca elementos consecutivos em sublistas

```
> pack ['a', 'a', 'a', 'a', 'b', 'c', 'c', 'a',
             'a', 'd', 'e', 'e', 'e', 'e']
["aaaa","b","cc","aa","d","eeee"]
```

(12) Implementar a função `encode`:

```
> encode "aaaabccaadeeee"
[(4,'a'),(1,'b'),(2,'c'),(2,'a'),(1,'d'),(4,'e')]
```

(13) Implementar a função `dupli`, que duplica os elementos de uma lista:

```
> dupli [1, 2, 3]
[1,1,2,2,3,3]
```

(14) Implementar a função `repli`, que replica os elementos de uma lista:

```
> repli 3 "abc"
"aaabbbccc"
```

(15) Implementar a função `drobEvery`, que retira cada nth elemento de uma lista

```
*Main> dropEvery 3 "abcdefghik"
"abdeghk"
```

(16) Implementar a função `split`

```
*Main> split 3 "abcdefghik"
("abc", "defghik")
```

(17) Implementar a função `slice`, que devolve um pedaço interno de uma lista:

```
*Main> slice 3 7 ['a','b','c','d','e','f','g','h','i','k']
"cdefg"
```

(18) Implementar a função `rotate`

```
*Main> rotate ['a','b','c','d','e','f','g','h'] 3
"defghabc"

*Main> rotate ['a','b','c','d','e','f','g','h'] (-2)
"ghabcdef"
```

(19) Implementar a função `removeAt`

```
*Main> removeAt 2 "abcd"
('b',"acd")
```