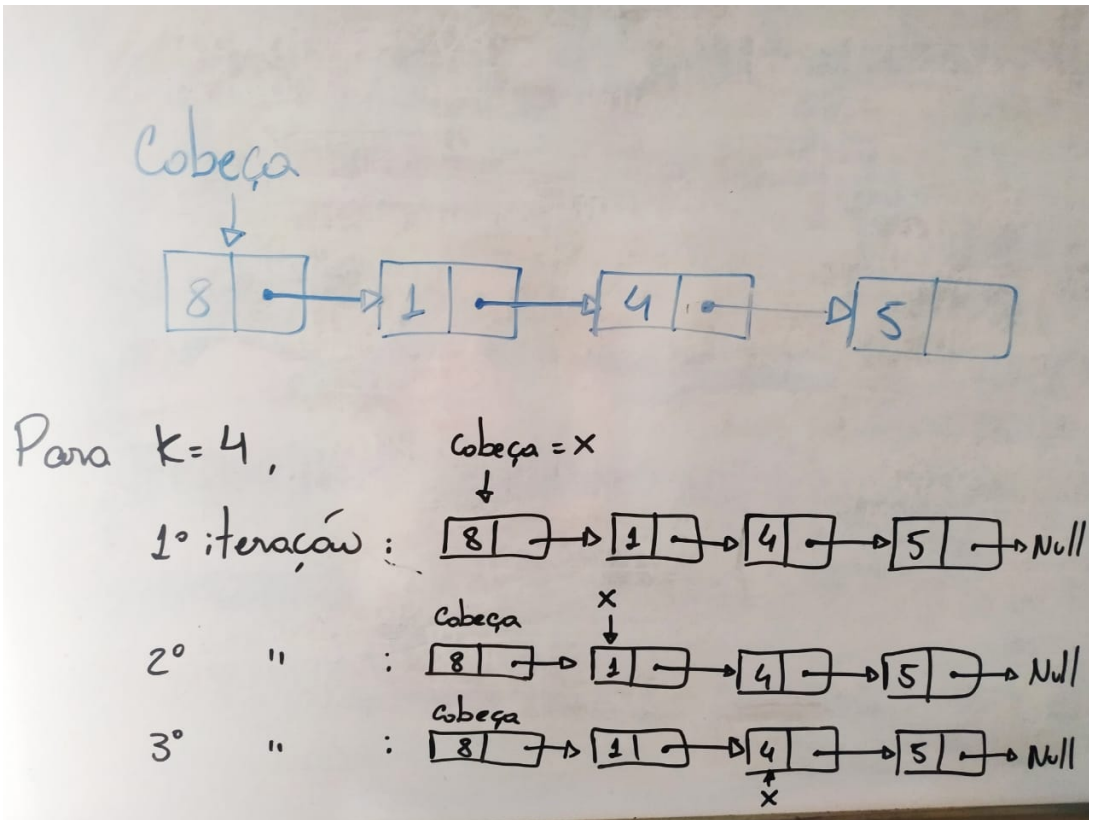
1. Buscar pelo primeiro nó que possui chave k na lista L.

Algoritmo 7.1: BUSCANALISTA(L, k)

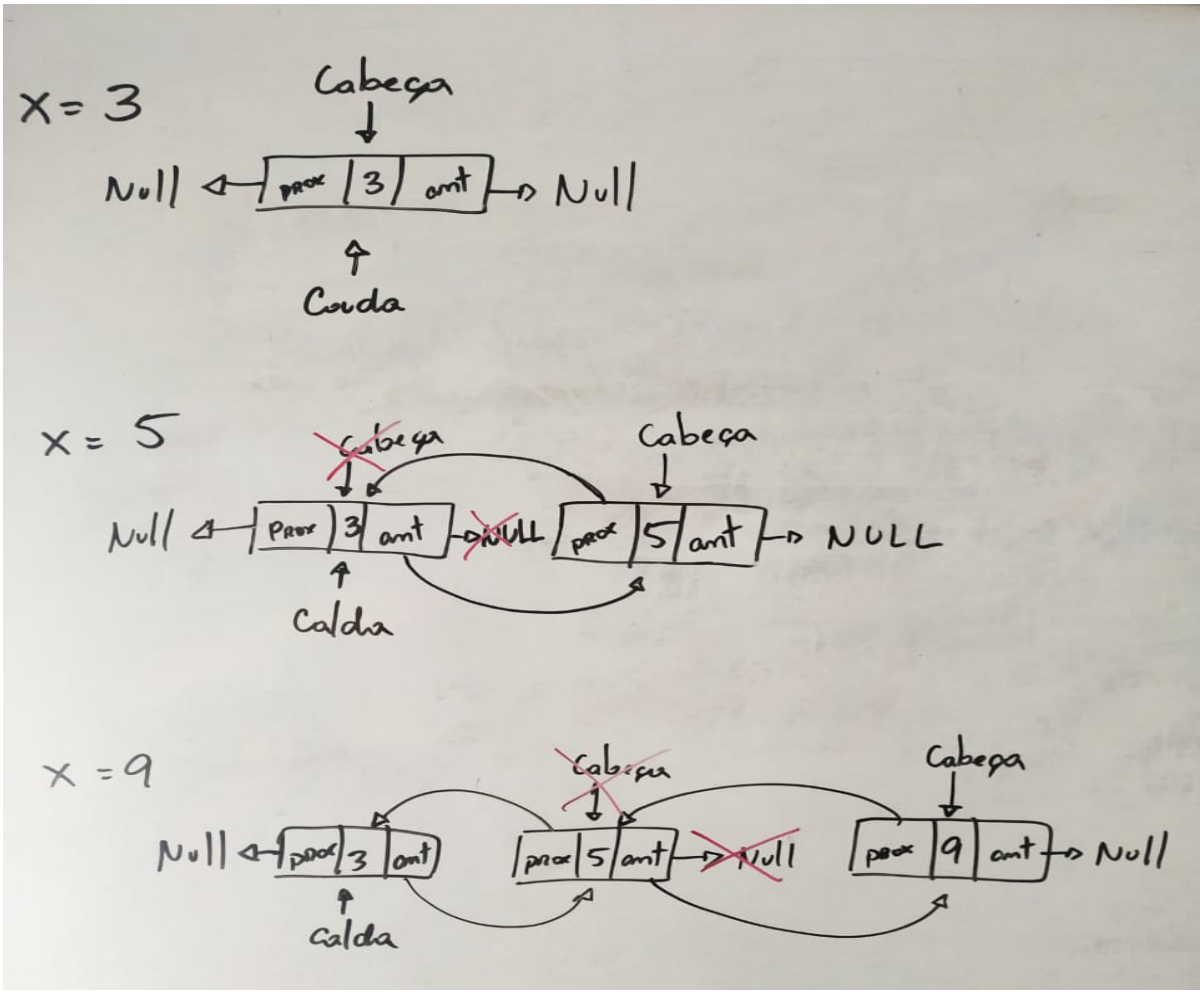
```
1 x = L.cabeca
2 enquanto x ≠ null e x.chave ≠ k faça
3     x = x.proximo
4 devolve x
```



2. Inserir um nó x no início da lista L.

Algoritmo 7.2: INSERENOINICIOLISTA(L, x)

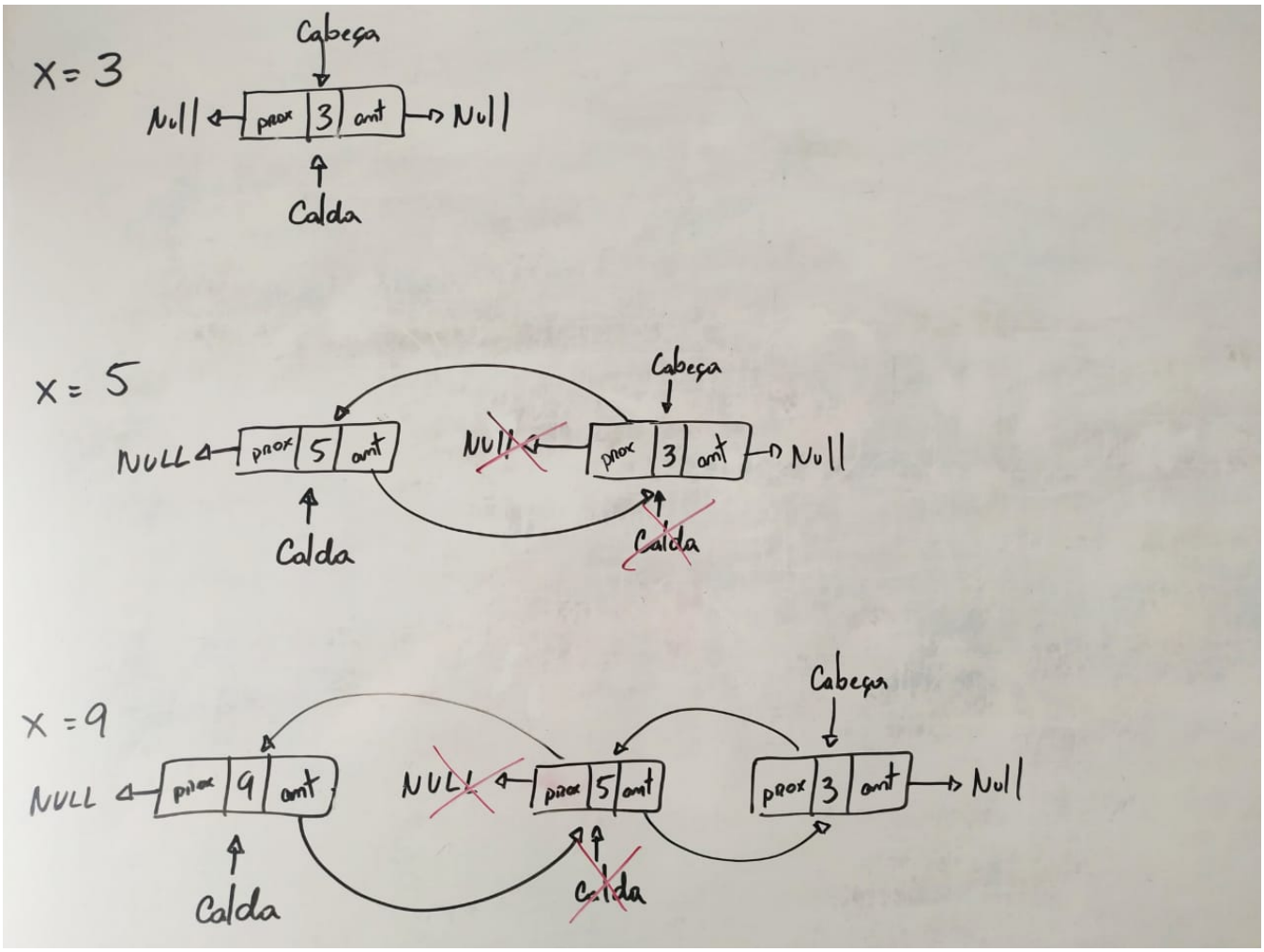
```
1 x.anterior = null
2 x.proximo = L.cabeca
3 se L.cabeca ≠ null então
4 |  L.cabeca.anterior = x
5 senão
6 |  L.cauda = x
7 L.cabeca = x
```



3. Inserir um nó x no fim da lista L.

Algoritmo 7.3: INSERENOFIMLISTA(L, x)

```
1 x.anterior = L.cauda
2 x.proximo = null
3 se L.cauda ≠ null então
4     L.cauda.proximo = x
5 senão
6     L.cabeca = x
7 L.cauda = x
```



4. Removendo um nó com chave k em uma lista L.

Algoritmo 7.4: REMOVEDALISTA(L, k)

```
1  x = L.cabeca
2  enquanto x ≠ null e x.chave ≠ k faça
3      x = x.proximo
4  se x = null então
5      devolve null
   /* Verificando se x é a cauda de L                                  */
6  se x.proximo == null então
7      L.cauda = x.anterior
8  senão
9      x.proximo.anterior = x.anterior
   /* Verificando se x é a cabeça de L                                 */
10 se x.anterior == null então
11     L.cabeca = x.proximo
12 senão
13     x.anterior.proximo = x.proximo
```