class No:

def \_\_init\_\_(self, valor):

self.valor = valor

self.anterior = None

self.proximo = None

self.size = 0

class List:

def \_\_init\_\_(self):

self.inicioDaLista = None

self.ultimo = None

def printLista(self):

atual = self.inicioDaLista

while atual:

print(atual.valor, end=" -> ")

if atual == self.ultimo:

break

atual = atual.proximo

print('')

def inserirPrimeiro(self, element):

newNode = No(element)

if self.inicioDaLista is None:

self.inicioDaLista = newNode

self.ultimo = newNode

else:

newNode.proximo = self.inicioDaLista

self.inicioDaLista.anterior = newNode

self.inicioDaLista = newNode

self.ultimo.proximo = self.inicioDaLista

self.inicioDaLista.anterior = self.ultimo

def inserirUltimo(self, element):

newNode = No(element)

if self.inicioDaLista is None:

self.inicioDaLista = newNode

self.ultimo = newNode

else:

self.ultimo.proximo = newNode

newNode.anterior = self.ultimo

self.ultimo = newNode

self.ultimo.proximo = self.inicioDaLista

self.inicioDaLista.anterior = self.ultimo

def inserirEm(self, element, position):

newNode = No(element)

if position == 0:

self.inserirPrimeiro(element)

else:

atual = self.inicioDaLista

for i in range(position):

atual = atual.proximo

anterior = atual.anterior

anterior.proximo = newNode

newNode.anterior = anterior

atual.anterior = newNode

newNode.proximo = atual

atual = newNode

def BuscarEm(self, position):

atual = self.inicioDaLista

if position == 0:

print(atual.valor)

else:

for i in range(position):

atual = atual.proximo

print('Item da ',position," posição é: ",atual.valor)

def BuscarPor(self, element):

contador = 0

atual = self.inicioDaLista

while atual:

if atual.valor == element:

print('Valor',element,' está na posição:',contador)

break

elif atual == self.ultimo:

print('Valor não encontrado')

break

atual = atual.proximo

contador+=1

def deletarPrimeiro(self):

if self.inicioDaLista is not None:

atual = self.inicioDaLista

proximo = atual.proximo

proximo.anterior = self.ultimo

self.ultimo.proximo = proximo

self.inicioDaLista = proximo

def deletarUltimo(self):

if self.inicioDaLista is not None:

atual = self.ultimo

anterior = atual.anterior

anterior.proximo = self.inicioDaLista

self.inicioDaLista.anterior = anterior

self.ultimo = anterior

def deletarItmPos(self, position):

if self.inicioDaLista is not None:

if position == 0:

self.deletarPrimeiro()

else:

atual = self.inicioDaLista

for i in range(position):

atual = atual.proximo

if atual == self.ultimo:

self.deletarUltimo()

else:

anterior = atual.anterior

proximo = atual.proximo

anterior.proximo = proximo

proximo.anterior = anterior

def removerDetElem(self, element):

if self.inicioDaLista is not None:

if element == self.inicioDaLista.valor:

self.deletarPrimeiro()

elif element == self.ultimo.valor:

self.deletarUltimo()

else:

atual = self.inicioDaLista

while atual:

if atual.valor == element:

anterior = atual.anterior

proximo = atual.proximo

anterior.proximo = proximo

proximo.anterior = anterior

break

elif atual == self.ultimo:

print('Valor não encontrado')

break

atual = atual.proximo

def quantLista(self):

contador = 1

atual = self.inicioDaLista

while atual:

if atual == self.ultimo:

break

atual = atual.proximo

contador += 1

print(contador)

lista = List()

lista.inserirPrimeiro(2)

lista.inserirPrimeiro(1)

lista.inserirUltimo(8)

lista.inserirPrimeiro(3)

lista.inserirUltimo(4)

lista.inserirEm(10,3)

lista.printLista()

lista.BuscarEm(5)

lista.BuscarPor(4)

lista.deletarPrimeiro()

lista.printLista()

lista.deletarUltimo()

lista.printLista()

lista.deletarItmPos(6)

lista.printLista()

lista.removerDetElem(7)

lista.printLista()

lista.quantLista()