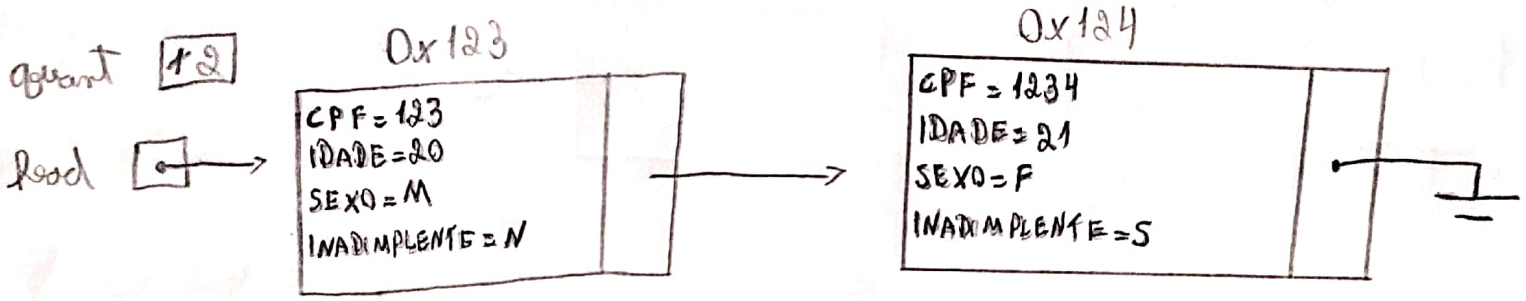
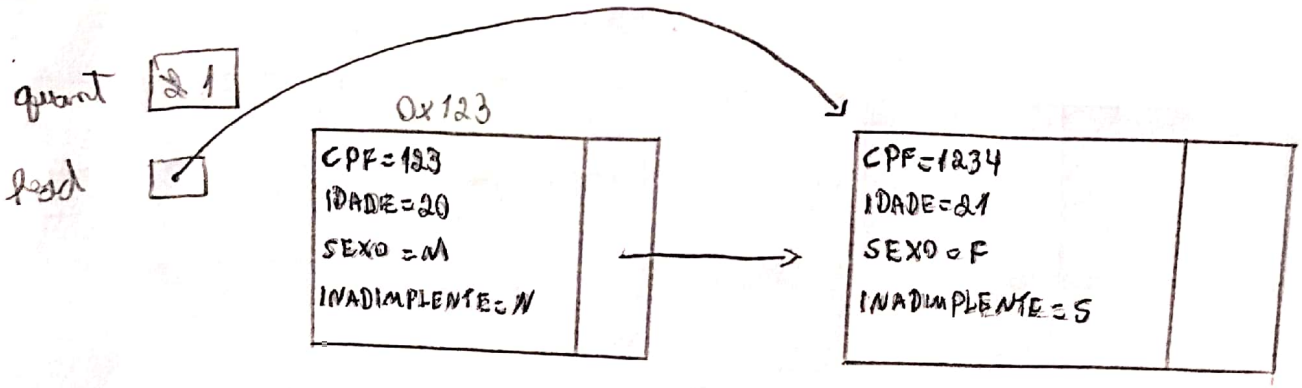
Método Insert

```
1 void Encadeada::insert(){
2     Pessoa p;
3     p.preencherCampos();
4     Nodo* novo = new Nodo();
5     novo->setItem(p);
6     novo->setProx(head);
7     head = novo;
8     quant++;
9 }
```



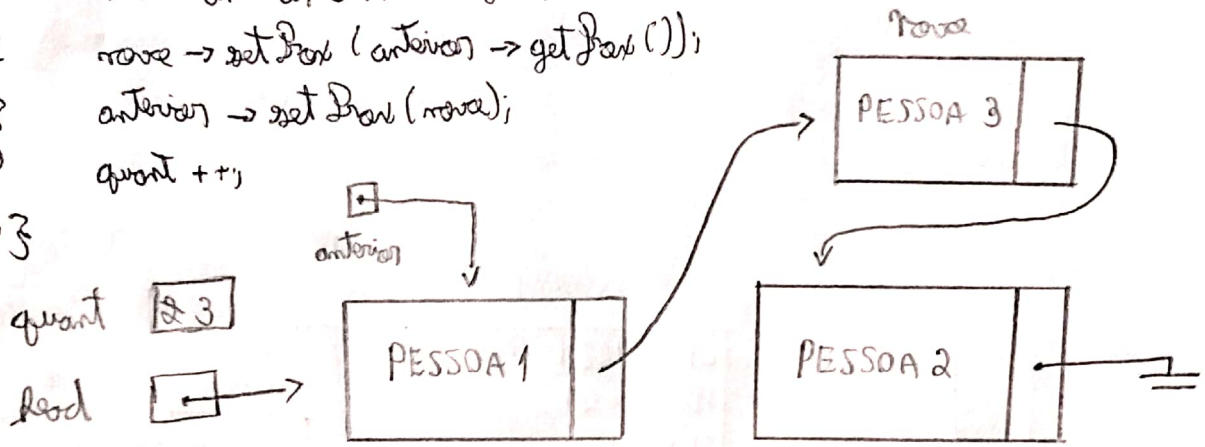
Método Remove

```
1 void Encadeada::remove(){
2     head = head->getProx();
3     quant--;
4 }
```



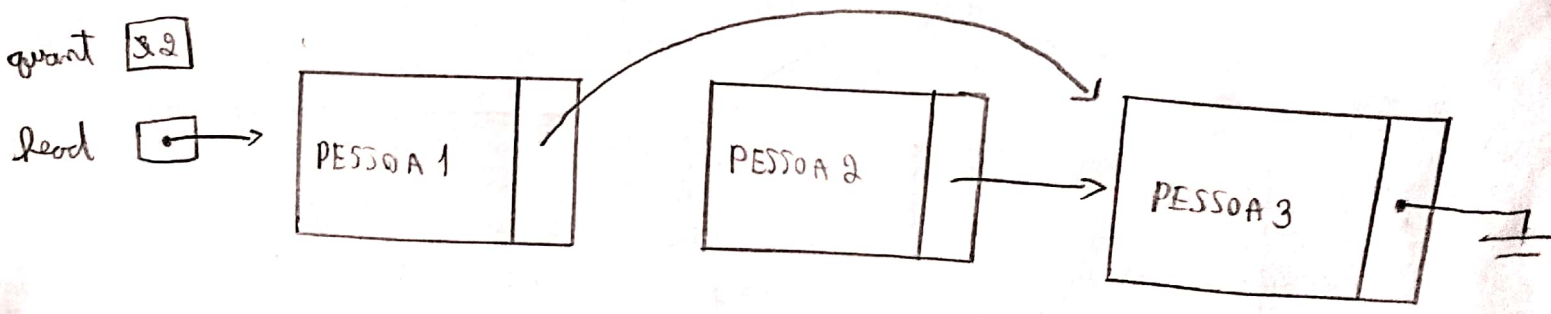
Método Insert na Posição

```
1  void Encadeada::insertPosicao(int n){
2      Pessoa p;
3      p.preencherCampos();
4      Nodo* nova = new Nodo();
5      nova->setItem(p);
6      Nodo* anterior = this->getElemento(n-1);
7      nova->setProx(anterior->getProx());
8      anterior->setProx(nova);
9      quant++;
10 }
```



Método Remove na Posição

```
1  void Encadeada::removePosicao(int n){
2      Nodo* anterior = this->getElemento(n-1);
3      Nodo* saida = anterior->getProx();
4      anterior->setProx(saida->getProx());
5      quant--;
6  }
```



```
1  Node* EncadeadaGet Elemento(int n){
2      Node* p = head;
3      int i = 1;
4      while(i <= n-1 && p->getProx() != NULL){
5          p = p-> getProx();
6          i++;
7      }
8      if(i == n){
9          return p;
10     }else{
11         return NULL;
12     }
13 }
```