UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática
Prof. Dr. Ivan Carlos Alcântara de Oliveira Estruturas de Dados I

Atividade N2 (2) - em dupla ou três

Nome	**TIA**
Cleverson Pereira da Silva	32198531
Gustavo Teixeira dos Santos	32197020
Natália de Fátima Teixeira	42129397

1. Desenvolver os métodos remove e busca e fazer testes.
2. Desenvolver um método **bool atualiza(int chave, elemento dado)** que atualiza o conteúdo (dado) do nó com a chave passada como parâmetro. Fazer testes!
3. Fazer um método para LDECO chamado: **LDL* intercala (LDL *Lista1, LDL *Lista2)** que combina a lista 1 com a lista 2 gerando uma terceira lista ordenada, retornando-a como resultado.
   Fazer um programa que testa este método.

No início do relatório incluir pelo menos 2 testes de cada um dos exercícios.

Colocar o código fonte completo em um Apêndice (no final do Relatório).

Não esquecer de enviar em separado o código fonte completo desenvolvido em cada exercício.

RELATÓRIO DE LABORATÓRIO

1. **Atividade:**
   a. **busca():**

```
Lista:
Lista: [ 1 2 3 5 12 22 23 25 43 55]
Digite o a chave do item que deseja buscar: 22

O valor da chave que escolheu é: 22

sh: 1: pause: not found
>
```

```
Lista:
Lista: [ 1 2 3 5 12 22 23 25 43 55]
Digite o a chave do item que deseja buscar: 55

O valor da chave que escolheu é: 55
```

b. remove():

```
Lista:
Lista: [ 1 2 3 5 12 22 23 25 43 55]
Digite o a chave do item que deseja remover: 22
Nova lista: Lista: [ 1 2 3 5 12 23 25 43 55]

sh: 1: pause: not found
```

```
Lista:
Lista: [ 1 2 3 5 12 22 23 25 43 55]
Digite o a chave do item que deseja remover: 55
Nova lista: Lista: [ 1 2 3 5 12 22 23 25 43]
```

2. Atualiza:

```
Lista:
Lista: [ 1 2 3 5 12 22 23 25 43 55]
Digite o a chave do item que deseja atualizar: 55
Digite o novo valor para o dado: 123
Nova lista: Lista: [ 1 2 3 5 12 22 23 25 43 123]

sh: 1: pause: not found
```

```
Lista:
Lista: [ 1 2 3 5 12 22 23 25 43 55]
Digite o a chave do item que deseja atualizar: 22
Digite o novo valor para o dado: 546
Nova lista: Lista: [ 1 2 3 5 12 546 23 25 43 55]
```

```
Lista:
Lista: [ 1 2 3 5 12 22 23 25 43 55]
Digite o a chave do item que deseja atualizar: 153

Item não encontrado
```

3. **Intercala:**

```
Lista 1:
Lista: [ 1 2 22 23 25]

Lista 2:
Lista: [ 3 5 12 43 55]

Lista intercalada:
Lista: [ 1 2 3 5 12 22 23 25 43 55]
```

```
Lista 1:
Lista: [ 1 2 22 23 25]

Lista 2:
Lista: [ 3 5 12 43 55]

Lista intercalada:
Lista: [ 1 2 3 5 12 22 23 25 43 55]

Lista intercalada após as operações:
Lista: [ 3 5 12 22 23 43 123]

Lista intercalada final em ordem decrescente:
Lista: [ 123 43 23 22 12 5 3]

Quantidade de itens na lista: 7
```

Código Fonte:

1. **Busca:**

```
No* LDL::busca(int chave){
  if(!isEmpty()){
    No *pAnda = this->cabeca;
    do{
      if(pAnda->getChave() == chave){
        return pAnda;
      }
      pAnda = pAnda->getProx();
    }while(pAnda != this->cabeca);
    return nullptr;
  }else{
    return nullptr;
  }
}
```

2. **Remove:**

```
bool LDL::remove(int chave){
  if(!isEmpty()){
    No *aux = this->busca(chave);
    if(aux != nullptr){
      if(this->qtde > 1){
        if(aux == this->cabeca){
          this->cabeca = aux->getProx();
        }
        aux->getAnt()->setProx(aux->getProx());
        aux->getProx()->setAnt(aux->getAnt());
        this->qtde--;
      }else{
        this->cabeca = nullptr;
        this->qtde = 0;
      }
        delete aux;
        aux = nullptr;
        return true;
    }else{
      return false;
    }
  }else{
    return false;
  }
}
```

3. **Atualiza:**

```
bool LDL::atualiza(int chave, Elemento dado){

  if(!isEmpty()){

    No *aux = busca(chave);
    if(aux != nullptr){
      aux->setDado(dado);
      return true;
    }else{
      return false;
    }

  }else{
    return false;
  }

}
```

4. **Intercala:**

```
LDL *intercala(LDL *Lista1, LDL *Lista2){
  No* pAnda = Lista2->getCabeca();
  do{
    Lista1->insereOrdemCrescente(pAnda->getChave(), pAnda->getDado());
    pAnda = pAnda->getProx();
  }while(pAnda != Lista2->getCabeca());
  return Lista1;
}
```